



INTRODUCTION À LA PROGRAMMATION D'EXERCICES INTERACTIFS

Sophie Lemaire et Bernadette Perrin-Riou

27 août 2010

bernadette.perrin-riou@math.u-psud.frsophie.lemaire@math.u-psud.fr

Ce document est une introduction à la programmation d'exercices interactifs à l'aide de l'interface proposée par WIMS appelée **Createxo**. Les exercices ainsi créés seront au format OEF (Online Format Exercise). Les copies d'écran sont faites sur un serveur wims sous le thème standard (version > 3.65d). Certaines fonctionnalités ne sont que dans les version supérieures à la version 4.00

Copyright ©2007 S. Lemaire, B. Perrin-Riou
GNU FDL Copyleft 2009

Table des matières

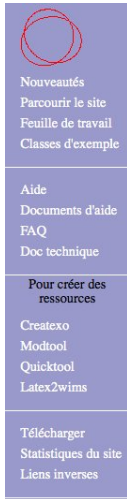
Chapitre 1. Les bases	3
1.1. L’interface Createxo	3
1.1.1. La documentation sur Createxo	4
1.2. Structure d’un exercice OEF	4
1.2.1. Premiers exemples de sources d’exercices	5
1.2.2. La déclaration de variables	7
1.3. Les types de réponses, premiers exemples	11
1.3.1. L’analyse de réponses numériques	12
1.3.2. Demander d’associer des objets	15
1.3.3. Répondre par un mot	16
1.3.4. Réponse à choix multiples	17
1.3.5. Répondre par une formule mathématique	20
1.4. Analyser une réponse par des conditions	21
Chapitre 2. Des outils pour développer des exercices OEF	23
2.1. Utilisation de commandes WIMS	23
2.2. Utilisation de macros de la slib	24
2.3. La présentation de l’énoncé	26
2.3.1. Quelques balises html	26
2.3.2. Disposition personnalisée des champs de réponses dans le cas d’un type à choix	28
2.4. Les boucles et les branchements	28
2.4.1. Conditions de test	28
2.4.2. La commande si ... alors ... (sinon) ...	29
2.4.3. La boucle for	30
2.4.4. La boucle tant que	31
2.5. L’utilisation de logiciels extérieurs	31
Chapitre 3. Exercices à base d’images ou de dessins	33
3.1. Mettre une image dans un exercice	33
3.2. Images et dessins	36
3.2.1. Faire un dessin	36
3.2.2. Dessiner sur une image	37
3.2.3. Insérer des champs de réponses sur une photo ou dans une figure	39
3.3. Les types de réponse « graphiques »	40
3.3.1. Le type coord (clic sur image)	40
3.3.2. Le type javacurve	40
Chapitre 4. Aides contextuelles et exercices à étapes	43
4.1. Aides dans le texte	43
4.1.1. Une explication sur un mot de l’énoncé qui apparaît uniquement en cliquant sur ce mot	43

4.1.2.	Une aide qui apparaît quand on clique sur un mot ou sur l'aide générale	44
4.1.3.	Une aide contenant elle-même une aide contextuelle	44
4.1.4.	Une aide qui dépend des données tirées pour réaliser l'exercice	44
4.1.5.	Aide à la souris : <code>utilities/tooltip</code>	45
4.2.	Les exercices à étapes	46
4.2.1.	Le principe	46
4.2.2.	Avoir un nombre de questions dépendant des données aléatoires de l'exercice	47
4.2.3.	Poser les questions les unes après les autres	48
4.2.4.	Permettre de refaire un essai	49
4.2.5.	Utiliser les réponses pour décider des étapes suivantes	50
4.2.6.	Complément	51
4.2.7.	Un exercice pour tout récapituler ! Le compte est bon	51
Annexe A.	Tableaux	53
1.1.	Conditions de test	53
1.2.	Commandes WIMS de base	54
1.3.	L'analyse des réponses	54
1.4.	Les commandes de dessin FLYDRAW	68
1.5.	Divers	71
Annexe B.	Les solutions	73

CHAPITRE 1

Les bases

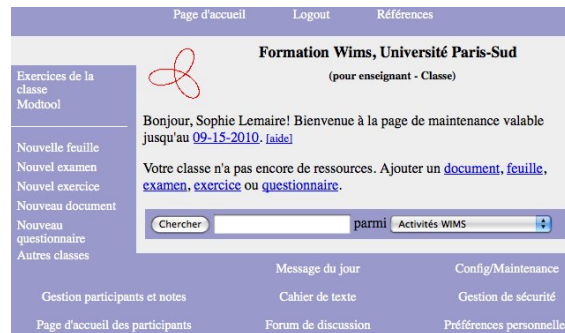
1.1. L'interface Createxo



(a) menu de gauche de la page principale de Wims

L'interface **Createxo** s'ouvre automatiquement lorsqu'on clique sur le lien « **Createxo** » dans le menu de gauche de la page principale de WIMS ou si vous êtes dans une classe en cliquant sur l'un des deux liens « Ajouter un exercice » ou « Nouvel exercice » de la page d'accueil de la classe.

NB : Si vous n'êtes pas dans une classe, l'exercice ne pourra être sauvegardé que sur votre ordinateur.



(b) la page d'accueil d'une classe Wims

FIG. 1.1.

Les indications qui seront données dans la suite sont faites pour des exercices créés dans une classe en utilisant le mode brut.

L'interface de **Createxo** permet :

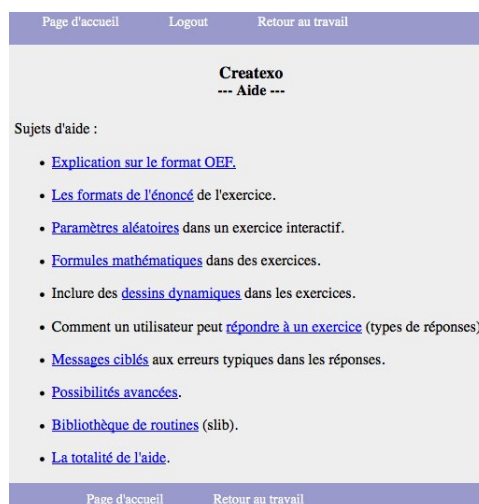
- de créer un nouvel exercice,
- de modifier un exercice existant dans sa classe,
- d'« importer » un exercice OEF présent sur le serveur WIMS dans sa classe, c'est-à-dire d'afficher le source d'un exercice OEF présent sur le serveur WIMS et de mettre cet exercice dans sa classe afin de s'en inspirer pour faire un nouvel exercice.

L'utilisation de **Createxo** pour créer ou modifier un exercice dans sa classe sera détaillée après la présentation d'exemples de sources d'exercices (figures 1.3, page 9 et 1.4, page 10). La figure 1.2, page 8 décrit les différentes étapes pour importer un exercice (l'exercice « Pourcentage » qui se trouve dans le module « OEF pourcentage ») dans sa classe.

NB : Pour mettre un exercice dans une feuille d'exercices, il n'est pas nécessaire de l'importer dans sa classe, il suffit de lancer dans sa classe l'exercice et de cliquer sur le lien « Insérer dans une feuille de travail » qui apparaît dans le menu à gauche de l'énoncé de l'exercice (voir figure 1.2.d, page 8).

1.1.1. La documentation sur Createxo

Lorsque vous êtes dans **Createxo**, le menu « Aide » donne accès à une documentation sur la programmation d'exercices OEF qui contient l'ensemble des commandes disponibles ; ce menu se trouve dans le bandeau en haut des fenêtres de **Createxo** (images des étapes (d), (e) et (f) de la figure 1.2, page 8). Si vous travaillez avec l'éditeur de **Createxo**, pensez à enregistrer ce que vous avez fait avant de cliquer sur ce menu. Une fois dans l'aide, le menu « Retour au travail » (figure ci-contre) permet de revenir à son fichier dans **Createxo**.



*La fenêtre obtenue en cliquant sur le menu « Aide » en haut d'une fenêtre de **Createxo**.*

Vous disposez sur la page d'entrée des sites WIMS d'un document d'aide à la programmation d'exercices OEF (DocAide Exercices interactifs) comprenant de nombreux exemples de sources d'exercices commentés. On l'obtient en cliquant sur le lien « Documents d'aide » (figure 1.1(a), page 3), puis sur le lien « Documentation par Bernadette Perrin-Riou ». Mentionnons enfin la documentation technique qui décrit les commandes wims et les macros développés par les utilisateurs : on l'obtient en cliquant sur le lien « Doc technique » apparaissant sur le menu de gauche de gauche de la page d'accueil du site (figure 1.1 (a)).

1.2. Structure d'un exercice OEF

Le fichier d'un exercice OEF peut être décomposé schématiquement en trois parties :

- (1) Une première partie s'exécute pendant le chargement de l'exercice (on appellera cette partie **Avant**)
 Dans cette partie du programme on trouve en particulier :
 - certaines instructions générales concernant l'exercice et son auteur, comme son titre, le format d'affichage (`tex` ou `html`, les coordonnées de l'auteur) ;

<code>\title{...}</code>	définit le titre de l'exercice
<code>\language{fr}</code>	dit que la langue de l'exercice est le français
<code>\author{...}</code>	définit l'auteur de l'exercice
<code>\email{...}</code>	définit l'adresse électronique de l'auteur
<code>\computeanswer{a}</code>	selon la valeur yes ou no donnée à a , l'utilisateur pourra entrer une réponse numérique en donnant une formule sans l'évaluer ou devra entrer un nombre
<code>\precision{n}</code>	n est un entier positif qui précise que la comparaison entre la réponse de l'utilisateur et la solution sera effectuée avec une tolérance de $1/n$ pour les réponses de type numérique.
<code>\range{n1..n2}</code>	la comparaison entre des fonctions se fera en évaluant les fonctions sur l'intervalle $[n1, n2]$

- la définition des variables nécessaires pour écrire l'énoncé et faire les calculs des réponses aux questions posées ;
 - tous les calculs nécessaires pour écrire l'énoncé et déterminer les réponses aux questions posées (si ces réponses ne dépendent pas de ce que va entrer l'utilisateur).
- (2) Une deuxième partie permet de gérer les affichages sur l'écran depuis le début de l'énoncé jusqu'à ce que l'utilisateur ait entré toutes ses réponses (on appellera cette partie **Pendant**).

L'affichage comporte l'énoncé et éventuellement des indications.

<code>\statement{...}</code>	pour écrire l'énoncé (cette commande doit apparaître une et une seule fois)
<code>\hint{...}</code>	pour écrire une indication (ce qui apparaît lorsqu'on clique sur le lien « Indications »)
<code>\help{...}</code>	pour un texte qui apparaîtra lorsque l'utilisateur clique sur « Aide » de la barre de menu en haut de l'exercice.

- (3) Une troisième partie permet d'analyser les réponses de l'utilisateur (on appellera cette partie **Après**). On peut de nouveau, dans cette partie, déclarer des variables et faire des calculs.

<code>\answer{...}{ ... }{type= ...}</code>	pour analyser une réponse
<code>\condition{...}{...}</code>	
<code>\feedback{...}{...}</code>	pour faire afficher un message en fonction de la réponse de l'utilisateur
<code>\solution{...}</code>	pour afficher un message lorsque l'exercice est terminé.

1.2.1. Premiers exemples de sources d'exercices

EXEMPLE 1.1. On veut faire un exercice donnant un entier, demandant à l'utilisateur de déterminer le carré de cet entier et renvoyant un message si l'utilisateur a entré un nombre négatif.

Avant: On déclare deux variables, l'entier n et la réponse $N = n^2$: ici n sera choisi comme un entier pris au hasard entre -50 et 50 :

```
\title{ Le carré d'un entier }
\computeanswer{ no }
\integer{ n = randint(-50..50) }
\integer{ N = (\n)^2 }
```

Pendant: On écrit l'énoncé « Calculer le carré de \n. »

```
\statement{ Calculer le carré de \n. }
```

Après: On analyse la réponse de l'utilisateur. Si l'utilisateur entre un nombre négatif, on fait afficher le message suivant « Le carré d'un entier est toujours positif. »

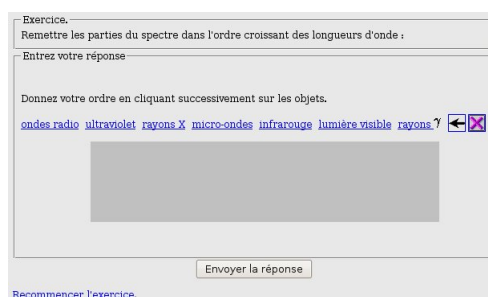
```
\answer{ Carré de \n }{ \N }{ type=numeric }
\feedback{ \reply1 < 0 }{ Le carré d'un entier est toujours positif. }
```

NB : Remarquer que

- la syntaxe des commandes est de la forme `\nom_commande{...}`
- si l'on appelle une variable que l'on a déclarée, on ajoute `\` devant son nom.

NB : La commande `\computeanswer{ no }` précise que l'utilisateur doit lui-même faire les calculs et entrer la valeur finale. Si par contre on met `\computeanswer{ yes }`, l'utilisateur peut entrer une formule comme $5*5$ laissant à l'ordinateur le soin de faire les calculs.

EXEMPLE 1.2. On veut écrire un exercice demandant de remettre les parties du spectre électromagnétique dans l'ordre croissant des longueurs d'onde.



Avant: On définit une variable contenant la réponse, c'est-à-dire la liste des noms des parties du spectre dans le bon ordre.

```
\title{Spectre électromagnétique}
\text{liste = rayons \(\gamma\),rayons X, ultraviolet,lumière visible,
infrarouge, micro-ondes, ondes radio}
```

Pendant: On écrit l'énoncé :

```
\statement{Remettre les parties du spectre dans l'ordre croissant des
longueurs d'onde :}
```

Après: On analyse la réponse en faisant appel au type de réponse : `reorder` qui fait le travail de comparaison entre `\liste` et la liste des items définie par l'utilisateur.

```
\answer{}{\liste}{type=reorder}
```

NB : La figure 1.3 décrit les différentes étapes de la création de cet exercice dans une classe. La figure 1.4 montre les étapes pour modifier un exercice déjà sauvegardé de la classe.

1.2.2. La déclaration de variables

On peut schématiquement distinguer quatre types de déclaration :

- les variables numériques (`integer`, `rational`, `real`, `complex`);
- les chaînes de caractères (`text`);
- les tableaux (`matrix`);
- les fonctions (`function`).

La déclaration d'une variable `A` se fait à l'aide d'une commande de la forme : `\nom_commande{A=...}`. Pour appeler une variable, on fait précéder son nom d'un `\`.

Par exemple,

- la commande `\integer{A = 3 + 2}` prend la chaîne de caractères `3 + 2`, la manipule comme entier (enlève les espaces, calcule le résultat en entier), puis renvoie le résultat comme chaîne de caractères `5`;
- la commande `\rational{B = 3 + 4/6}` prend la chaîne de caractères `3 + 4/6`, la manipule comme rationnel (enlève les espaces, calcule le résultat comme fraction, simplifie la fraction) puis renvoie le résultat comme chaîne de caractères `11/3`;
- la commande `\text{C = \B + 1}` prend la chaîne de caractères `\B + 1`, remplace `\B` par la chaîne de caractères correspondante, puis renvoie le résultat comme chaîne de caractères `11/3 + 1`.
- la commande

```
\matrix{D = 1,2
3,4}
```

prend la chaîne de caractères

```
1,2
3,4
```

remplace les retours à la ligne par un point-virgule puis renvoie le résultat comme chaîne de caractères `1,2;3,4`.

- la commande `\function{f = +x-1+0}` prend la chaîne de caractères `+x-1+0`, enlève les espaces et le `+` inutile puis renvoie la chaîne de caractères `x-1+0`.

Autre exemple :

```
\function{t = x+4}
\integer{a = -2}
\function{f = +1-\a*\t/(1-\a)}
```

La variable `f` contient la chaîne de caractères `1+2*x+4/(1+2)` ; `-2` a été simplifié et `\t` a été remplacé par la chaîne de caractères `x+4` telle quelle, ce qui donne un résultat différent de `1+2*(x+4)/(1+2)`.

NB : La déclaration des variables ne peut se faire que dans les parties **Avant** et **Après**.

NB : Les réponses de l'utilisateur sont conservées dans des variables appelées `reply1`, `reply2`, ...

(a) Taper un mot-clé par exemple « pourcentage » dans la zone de saisie de la page d'accueil de la classe et cliquer sur « Chercher » (figure 1.1.a), page 3.

Formation Wims, Université Paris-Sud
(pour enseignant - Classe)

Bonjour, Sophie Lemaire! Bienvenue à la page de maintenance valable jusqu'au 09-15-2010. [aide]

Votre classe n'a pas encore de ressources. Ajoutez un [document](#), [feuille](#), [examen](#), [exercice](#) ou [questionnaire](#).

Chercher pourcentage parmi Exercices interactifs [vider]

Nous avons trouvé 9 Exercices interactifs pour votre recherche.

[OEF pourcentage](#), collection d'exercices sur les pourcentages (Bernadette PERRIN-RIOU)

(b) Cliquer sur le module « OEF pourcentage ».

OEF pourcentage
--- Introduction ---

Ce module regroupe pour l'instant 2 exercices sur les pourcentages.

Choisissez les exercices : Pourcentage Pourcentage II

Une série aura 1-3 exercices.

Niveau de sévérité 3-4

Chronomètre secondes.

Nombre de réponses affichées dans les QCM: 4-5

Donner une solution (si disponible) ? si la note est non maximale

☒ Toujours afficher une bonne réponse dans les QCM.

☒ Pénalité pour mauvaise réponse dans les QCM.

☒ Afficher la bonne réponse.

☒ Permettre les indications (si disponibles).

[Au travail] [Version imprimable]

(c) Sélectionner l'exercice « Pourcentage » et cliquer sur « Au travail ».

Insérer dans une feuille de travail Importer dans la classe

Exercice.

On augmente un volume v de 85 %. Par combien doit-on multiplier v ?

Entrez votre réponse

On doit multiplier v par =

[Envoyer la réponse]

[Recommencer l'exercice.](#)

Page d'accueil Intro/Config Aide A propos de cette ressource

(d) L'énoncé de l'exercice apparaît. Cliquer sur le lien « Importer dans la classe ».

Page d'accueil Logout Aide

Createxo

Voici le source d'un exercice public que vous voulez importer. Vous pouvez le modifier dans la fenêtre ci-dessous avant de le mettre dans la classe. Vous pouvez tester certains programmes externes avec le [testeur de script](#).

```
\title{Pourcentage}
\language{fr}
\range{-5..5}
\author{Bernadette PERRIN-RIOU}
\email{bpe@math.u-psud.fr}
\computeanswer{no}
\format{html}
\precision{10000}

\integer{n=randint(10..90)}
\text{u=randint(1..5)}
\text{longueur=item(u,une longueur, un volume,une hauteur,une masse,un
\text{l=item(u,l,v,h,m,p)}
\text{choix=randint(1..1)}
\text{augdim=augmente,diminue}
\text{augmente=choix=1 ? item(1,augdim):item(2,augdim)}

\real{rep=(100*(choix)*n)/100}

\statemnt{On augmente \longueur \l de \n %. Par combien doit-on m
\answer{On doit multiplier \l par \rep{type=numeric}}
```

[Vider] [Abandonner les modifications].

(e) Le source de l'exercice est visible dans l'éditeur de Createxo et on peut le modifier. Cliquer ensuite sur « Envoyer le source ».

Page d'accueil Logout Aide

Createxo

Le logiciel a reconnu votre exercice avec succès, avec les informations suivantes. (Vérifiez avec ce que vous avez défini.)

Titre Pourcentage

Nombre de paramètres 9

Nombre de réponses libres 1

Nombre de choix multiples 0

Maintenant vous pouvez

- [tester votre exercice.](#)
- [modifier le source de l'exercice.](#)
- [déposer un fichier \(image, pdf, ...\) dans l'exercice.](#)
- [mettre cet exercice dans votre classe.](#)
- [sauvegarder le source](#) de l'exercice sur votre ordinateur !

Page d'accueil Aide A propos de cette ressource

(f) Cliquer sur le lien « mettre cet exercice dans votre classe ».

Page d'accueil Logout Aide

Createxo

Votre exercice est mis en place avec succès. Cet exercice sera accessible uniquement aux participants de votre classe et uniquement si vous le mettez dans une feuille de travail.

Vous pouvez maintenant :

- [consulter les exercices de la classe.](#) (et éventuellement en mettre dans vos feuilles de travail.)
- [retourner à l'étape précédente.](#)

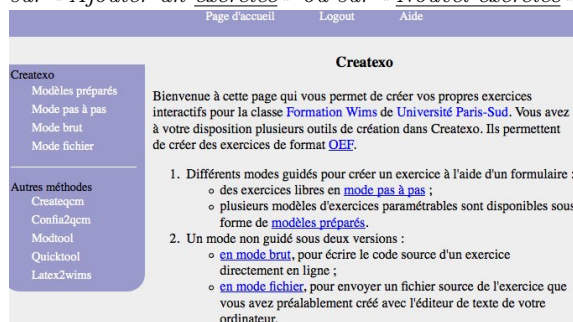
Page d'accueil Aide A propos de cette ressource

(g) Cliquer sur « consulter les exercices de la classe », l'exercice « Pourcentage » a été ajouté dans la liste des exercices de la classe.

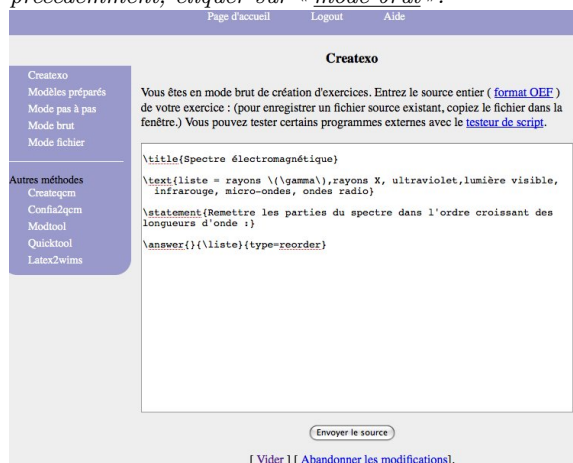
FIG. 1.2. Les étapes pour importer l'exercice « Pourcentage » qui se trouve dans le module « OEF pourcentage »



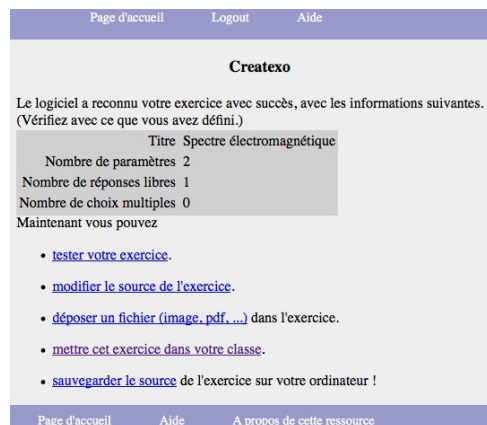
(a) Pour créer l'exercice « Spectre électromagnétique » cliquer sur « Ajouter un exercice » ou sur « Nouvel exercice ».



(b) Pour entrer le source de l'exercice tel qu'il a été présenté précédemment, cliquer sur « mode brut ».



(c) L'éditeur de Createxo en mode brut : taper le source de l'exercice dans la fenêtre comme cela est visible sur l'image et cliquer sur le bouton « Envoyer le source ».



(d) L'exercice est maintenant visible en cliquant sur « tester votre exercice » mais n'est pas encore sauvegardé. Pour le sauvegarder dans la classe, cliquer sur mettre cet exercice dans votre classe (le lien « sauvegarder le source » permet d'avoir une copie du source de l'exercice sur son ordinateur)



(e) Cliquer sur « consulter les exercices de la classe ». On sort de l'interface de Createxo.

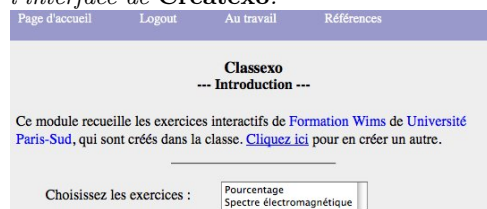


FIG. 1.3. Les images (a) à (e) décrivent les différentes étapes pour créer l'exercice « Spectre électromagnétique » en étant dans une classe. Remarquer le menu « Aide » sur le bandeau en haut de chaque fenêtre de **Createxo** (images (b) à (e)) : en cliquant dessus, on obtient la documentation de Createxo décrivant les commandes utilisables dans les exercices OEF (il vaut mieux enregistrer l'exercice avant de cliquer sur « Aide »)

(a) Cliquer sur le lien « *Exercices de la classe* » dans la page d'accueil de sa classe (figure 1.3.a), page 9

Classeexo
--- Introduction ---

Ce module recueille les exercices interactifs de Formation Wims de Université Paris-Sud, qui sont créés dans la classe. [Cliquez ici](#) pour en créer un autre.

Choisissez les exercices : Le carré d'un entier **Spectre électromagnétique**

Une série aura 1 exercices.

Niveau de sévérité 3

Chronomètre secondes.

Nombre de réponses affichées dans les QCM : 4

Donner une solution (si disponible) ? si la note est non maximale

☒ Toujours afficher une bonne réponse dans les QCM.
☒ Pénalité pour mauvaise réponse dans les QCM.
☒ Afficher la bonne réponse.
☒ Permettre les indications (si disponibles).

Au travail Version imprimable

(b) Sélectionner l'exercice à modifier (ici « *Spectre électromagnétique* »). Cliquer sur le bouton « *Au travail* ».

Exercice.
Remettre les parties du spectre dans l'ordre croissant des longueurs d'onde.

Entrez votre réponse

Donnez votre ordre en cliquant successivement sur les objets.

rayons X rayons gamma infrarouge ondes radio ultraviolet micro-ondes lumière visible ↔ ✖

Envoyer la réponse

[Recommencer l'exercice.](#)

(c) Cliquer sur le lien « *Modifier le source* » dans le menu à gauche de l'énoncé de l'exercice ; le source de l'exercice s'affiche dans l'éditeur du mode brut (figure 1.3.c, page 9).

Page d'accueil Logout Aide

Createxo

Voici le source d'un exercice existant de votre classe. Vous pouvez le modifier dans la fenêtre ci-dessous. Vous pouvez tester certains programmes externes avec le [testeur de script](#).

```
\title{spectre électromagnétique}
\author{S. Lemaire}

\text{liste = rayons X, ultraviolet, lumière visible,
infrarouge, micro-ondes, ondes radio}

\statement{Remettre les parties du spectre dans l'ordre croissant des
longueurs d'onde :}

\answer{\{liste\}(type=reorder)}
```

Envoyer le source

[\[Voir \]](#) [\[Abandonner les modifications \]](#)

(d) Faire les modifications voulues et cliquer sur « *Envoyer le source* »

Page d'accueil Logout Aide

Createxo

Le logiciel a reconnu votre exercice avec succès, avec les informations suivantes. (Vérifiez avec ce que vous avez défini.)

Titre Spectre électromagnétique
 Nombre de paramètres 2
 Nombre de réponses libres 1
 Nombre de choix multiples 0
 Maintenant vous pouvez

- [tester votre exercice.](#)
- [modifier le source de l'exercice.](#)
- [déposer un fichier \(image, pdf, ...\)](#) dans l'exercice.
- mettre cet exercice dans votre classe [pour remplacer l'ancien](#) ou [pour en faire un nouveau](#).
- [sauvegarder le source](#) de l'exercice sur votre ordinateur !

(e) Vous pouvez tester les modifications effectuées sur le source en cliquant sur « *tester votre exercice* » mais attention les modifications effectuées ne sont pas encore sauvegardées. Pour sauvegarder les modifications dans votre classe, vous avez deux possibilités : vous pouvez cliquer sur le lien « *pour remplacer l'ancien* ». Dans ce cas, la version précédente sera écrasée. Vous pouvez cliquer sur « *pour en faire un nouveau* » pour conserver les deux versions ; pensez dans ce cas à modifier le titre pour pouvoir distinguer les deux versions.

FIG. 1.4. Les images décrivent les différentes étapes pour pouvoir modifier le source d'un exercice (ici « *Spectre électromagnétique* ») qui se trouve dans la classe.

NB : Les variables déclarées comme `text` ou `matrix` peuvent contenir plusieurs éléments, les virgules et/ou les points-virgules servent à séparer les différents éléments : par exemple, la commande

```
\matrix{D = 1,2
3,4}
```

prend la chaîne de caractères

```
1,2
3,4
```

remplace les retours à la ligne par un point-virgule puis renvoie le résultat comme chaîne de caractères `1,2 ; 3,4`. la variable `D` en tant que liste contient 3 éléments séparés les uns des autres par des virgules : `\D[1]` est égal à 1, `\D[2]` est égal à 2 ; 3 et `D[3]` est égal à 4.

En tant que tableau `D` contient 4 éléments, 2 sur la première ligne et deux sur la deuxième ligne, `\D[i ; j]` est l'élément de la i -ème ligne et de la j -ème colonne, ici `\D[2 ; 1]` contient 3. On peut extraire plusieurs éléments en même temps : `\D[2 ;]` est la chaîne de caractères `3,4` qui a deux éléments 3 et 4.

Autre exemple : la commande

```
\matrix{E = tu as,1
tu vas,1,0}
```

définit une variable en tant que tableau à 5 éléments : par exemple, `\E[1 ; 1]` est la chaîne de caractères `tu as`, `\E[2 ; 2..3]` est la chaîne de caractères `1,0`.

1.3. Les types de réponses, premiers exemples

La commande `\answer{texte}{\A}{type=un_type}` exécute en général les actions suivantes :

- formatage de la présentation `html` sous la forme sous laquelle la question va être posée : zone de texte à écrire, étiquettes à déplacer, zone à cliquer ;
- récupération des données transmises par l'utilisateur ;
- analyse de la réponse en la comparant à la réponse donnée par le développeur de manière différente selon qu'il s'agit de nombres, de matrices, ou de tout autre format ;
- renvoi d'une note ;
- renvoi des variables `reply1` et `sc_reply1`¹ contenant des éléments de réponses qu'on peut réutiliser dans un feedback.

Dans la première accolade, on peut mettre du texte qui apparaît devant le champ réservé à la réponse. La deuxième accolade doit contenir les éléments permettant de décider si la réponse est juste ou non. Ces éléments vont dépendre du type précisé dans le troisième accolade.

On peut poser plusieurs questions dans un même exercice. La source de l'exercice contiendra autant de commandes `\answer` qu'il y a de questions posées.

EXEMPLE 1.3.

¹De façon générale, `\reply1` renvoie toute l'information nécessaire pour reconstituer la réponse de l'utilisateur alors que `\sc_reply1` vaut 1 si la réponse donnée est juste, 0.5 si la réponse donnée est partiellement juste et 0 si la réponse donnée est fausse pour les types de réponse où elle n'a pas été définie explicitement.

Exercice.

Calculer le carré et le cube de -13.

Entrez votre réponse

Carré de -13 =

Cube de -13 =

Envoyer la réponse

Un exercice comme ci-dessus demandant de calculer le carré et le cube d'un entier peut se programmer de la façon suivante :

```
\title{ Le carré et le cube d'un entier }
\computeanswer{ no }
\integer{ n = randint(-50..50) }
\integer{ N = (\n)^2 }
\integer{ M = \n*\N }

\statement{ Calculer le carré et le cube de \n. }

\answer{ Carré de \n }{ \N }{ type = numeric }
\answer{ Cube de \n }{ \M }{ type = numeric }
\feedback{ \reply1 < 0 }{ Le carré d'un entier est toujours positif. }
\feedback{ \M < 0 and \reply2 > 0 }{ Le cube d'un entier négatif est négatif. }
\feedback{ \M > 0 and \reply2 < 0 }{ Le cube d'un entier positif est positif. }
```

On a utilisé pour faire afficher des commentaires que la seconde commande `\answer` renvoie des variables `reply2` et `sc_reply2`.

Après ce résumé général, nous allons regarder quelques types simples.

NB : Il est conseillé au départ de finir tous les exercices par `\feedback{1 = 1}{\reply1}` pour voir afficher le contenu de la variable `reply1`.

NB : En mode développement, lorsqu'on teste un exercice les réponses calculées apparaissent dans les champs de réponse afin de faciliter la vérification de l'exercice.

1.3.1. L'analyse de réponses numériques

1.3.1.1. Le type *numeric*

La réponse est comparée numériquement à la solution avec une tolérance qui est précisée par la commande `\precision{...}`² dans l'en-tête de l'exercice.

EXEMPLE 1.4. Dans l'exemple 1.1, on a utilisé la commande

```
\answer{ Carré de \n }{ \N }{type=numeric}.
```

Si `\n` contient l'entier 5, cela donne l'affichage :

Entrez votre réponse :

Carré de 5 =

Avec le type `numeric`, une fois que l'utilisateur a envoyé sa réponse, la valeur de `\reply1` sera celle entrée par l'utilisateur et la valeur de la variable `sc_reply1` sera 1 si la réponse est bonne, 0 si la réponse est fautive et 0.5 si elle est bonne à précision près.

²La réponse numérique r est acceptée pour la solution demandée s si et seulement si : $\frac{|s-r|}{\max(|s+r|, \frac{1}{M})} < \frac{1}{M}$ où M est l'entier déclaré par la commande `\precision{M}`

EXERCICE 1.1. L'objectif est de programmer un exercice demandant la longueur de la bordure d'un pré rectangulaire.

- (1) Cliquez³ sur **Ajouter un exercice**, puis sur le lien **mode brut** pour pouvoir écrire directement le source de l'exercice (n'utilisez donc ni un mode guidé, ni le mode fichier).
- (2) Donnez un titre à votre exercice, déclarez deux variables : longueur et largeur que l'on choisira parmi les nombres entiers, puis écrivez l'énoncé de l'exercice (vous pouvez vous aider du source de l'exemple 1.1, page 5).
- (3) Votre exercice n'est pas fini, mais vous pouvez déjà sauvegarder ce que vous avez fait : cliquez sur « envoyer le source », puis sur « mettre cet exercice dans votre classe. », puis sur « consulter les exercices de la classe ». Regardez ce que donne votre exercice. Puis cliquez sur « Modifier le source ».

NB : Le source doit au moins contenir un titre et la commande `\statement` pour pouvoir être enregistré sans ambiguïté.

- (4) Complétez le source de l'exercice
 - en déclarant une autre variable par exemple `\per` donnant la valeur du périmètre,
 - en rajoutant une commande de la forme `\answer{Le périmètre est}{\per}{type=numeric}` permettant l'analyse automatique de la réponse de l'utilisateur.
- (5) Cliquez de nouveau sur « envoyer le source », puis sauvegardez-le en cliquant sur le lien « pour remplacer l'ancien ». Vérifiez votre exercice en cliquant sur « tester votre exercice ».

1.3.1.2. Le type `numexp` pour les fractions rationnelles

Dans l'exercice 1.1, la bonne réponse est nécessairement un entier, or toute réponse suffisamment proche de la solution sera considérée comme bonne. Par exemple, si la solution est 40000, l'utilisateur ne sera pas pénalisé s'il entre 40000.1.

Pour que seule la réponse numériquement identique à la solution soit considérée comme bonne, on peut utiliser le type `numexp` :

```
\answer{}{\rep}{type=numexp}
```

Ce type permet d'analyser des réponses qui s'écrivent comme des fractions rationnelles : si `\rep = 1/4`, les réponses 1/4 et 0.25 seront acceptées. Par contre, la réponse 2/8 sera refusée, un message d'erreur apparaîtra :

« Votre réponse 2/8 n'est pas une écriture irréductible. Veuillez réduire la fraction. »

Si `\rep = 1/3`, toute réponse de la forme 0.333333 sera considérée comme mauvaise.

NB : On peut ajouter l'option `noreduction` au type `numexp` pour que les réponses sous forme de fractions non réduites soient acceptées :

```
\answer{}{\rep}{type=numexp}{option=noreduction}
```

EXERCICE 1.2. Faire une variante de l'exercice 1.1 en demandant aussi de calculer la superficie du pré et utiliser le type `numexp` au lieu du type `numeric`.

NB : La première réponse de l'utilisateur (pour la valeur du périmètre) est conservée dans une variable qui s'appelle `reply1`. La deuxième réponse (pour la valeur de la superficie) sera conservée dans la variable `reply2`.

³Vous pouvez vous aider des copies d'écran données figure 1.3, page 9 pour créer votre exercice

1.3.1.3. Le type *units*

Dans l'exercice 1.2, il peut être intéressant de demander des réponses avec une unité. On peut utiliser le type **units** pour cela.

Par exemple, avec `\answer{réponse}{4 m^2}{type=units}`, la réponse sera juste si l'on entre $4m^2$, mais aussi $400dm^2$. La réponse 4 par contre sera refusée.

EXERCICE 1.3. Modifier le source de l'exercice 1.2 afin d'utiliser le type **units**.

EXERCICE 1.4. Faire une variante de l'exercice 1.3 pour que les champs des réponses soient intégrés dans l'énoncé ; par exemple, l'énoncé de l'exercice pourrait être :

Un pré rectangulaire de longueur 40 m et de largeur 10 m a une bordure de et une superficie de .

Indication : pour inclure un champ de réponse dans l'énoncé, on ajoute la commande `\embed` à l'endroit de l'énoncé où on veut que le champ de réponse apparaisse. Par exemple, pour avoir un énoncé de la forme

Le carré de -12 est .

dans l'exemple 1.1, page 5, il suffit de remplacer le **statement** dans le source de l'exercice par

`\statement{ Le carré de \n est : \embed{reply1,5} }`

Le deuxième paramètre⁴ de `\embed{}`⁵, ici 5, précise la longueur du champ de réponse.

1.3.1.4. Le type *range*

Avec la commande

`\answer{texte}{val_min,val_max,val_affi}{type=range}`

toutes les réponses incluses dans l'intervalle $[val_min, val_max]$ sont acceptées. La valeur **val_affi** précise la valeur affichée par WIMS en cas de mauvaise réponse. Si **val_affi** n'est pas donnée, c'est le milieu de l'intervalle qui est affiché.

EXERCICE 1.5. Faire un exercice demandant la circonférence d'un disque. Prendre le rayon avec deux décimales et utiliser le type **range** pour la réponse.

Indications : La constante π s'écrit *pi*.

1.3.1.5. Les types *vector* et *matrix*

Les types **vector** et **matrix** permettent d'analyser les vecteurs et les matrices dont les coefficients sont des nombres. La comparaison de la réponse avec la solution se fait coefficient par coefficient avec une tolérance précisée par la commande `\precision`. Un seul champ de réponse est affiché ; l'utilisateur doit entrer les coefficients d'un vecteur ou d'une ligne d'une matrice en les séparant par une virgule et passer à la ligne pour écrire les coefficients de la ligne suivante dans le cas d'une matrice.

EXEMPLE 1.5. Le fichier source d'un exercice demandant de calculer le produit d'une matrice 2x2 à coefficients entiers par un vecteur colonne à coefficients entiers.

⁴la signification de ce deuxième paramètre varie en fonction du type choisi pour analyser la réponse.

⁵on peut aussi écrire `\embed{ r1 }` ou `\embed{ r 1 }`


```

\title{Mutiplication par une matrice 2x2}
\precision{10000}
\integer{a = randint(-5..5)}
\integer{b = randint(-5..5)}
\integer{c = randint(-5..5)}
\integer{d = randint(-5..5)}
\integer{x = randint(-5..5)}
\integer{y = randint(-5..5)}
\matrix{M = \a,\b
\c,\d}
\matrix{v=\x
\y}
\integer{ux = \a*\x+\b*\y}
\integer{uy = \c*\x+\d*\y}
\matrix{rep = \ux
\uy}

\statement{Calculer le produit  $(AB)$  avec  $(A) = ([M])$  et
 $(B) = ([v])$ }
\answer{ $(AB)$ }{\rep}{type = matrix}

```

1.3.2. Demander d'associer des objets

Le type **correspond** permet de faire un exercice demandant d'associer correctement les objets de deux listes, par exemple une liste de mots et leur traduction. Les deux listes peuvent être déclarées comme un tableau avec deux colonnes, le premier élément d'une ligne devant être associé au deuxième élément de la ligne. La structure du source d'un tel exercice est :

```

\matrix{data = a1,b1
a2,b2
...
an,bn}

\statement{Associer les éléments :
<center>\embed{reply1}</center>}

\answer{}{\data[;1];\data[;2]}{type=correspond}

```

NB : `\data[;1]` extrait la première colonne du tableau (pour plus de détails sur la syntaxe des tableaux voir la section 1.2.2, page 7)

Tel quel, tous les éléments de la liste seront présentés. Si on veut par exemple que l'exercice propose d'associer 4 éléments pris au hasard dans la liste, on rajoutera dans la première partie de l'exercice :

```

\text{mix = shuffle(rows(\data))}
\text{data = \data[\mix[1..4]; ]}

```

La première commande permet de mélanger aléatoirement les numéros des lignes du tableau de données. La 2ème commande extrait les quatre lignes correspondant aux quatre premiers numéros de la liste `\mix`.

EXERCICE 1.6. Ecrire un exercice de correspondance sur les unités du système SI en utilisant les données suivantes⁶ :

```
longueur,le mètre
masse,le kilogramme
temps,la seconde
intensité de courant électrique,l'ampère
température,le kelvin
intensité lumineuse,le candela
quantité de matière,la mole
angle plan,le radian
angle solide,le stéradian
```

Exécuter votre exercice en faisant une faute, puis deux fautes. Comparer les notes obtenues.

NB : On peut rajouter une option à la commande `\answer` si l'on n'est pas satisfait de cette notation :

```
\answer{{...}}{type=correspond}{option=split}
```

1.3.3. Répondre par un mot

Les types **case**, **nocase** et **atext** permettent de faire une analyse automatique d'une réponse composée de mots. La syntaxe est la même pour ces trois types. Par exemple,

```
\answer{...}{liste_sol}{type=case}
```

où `liste_sol` est la bonne réponse éventuellement suivie de variantes qui seront considérées aussi comme des bonnes réponses, chacune étant séparée par une barre verticale |.

Différences entre ces trois types.

- Avec le type **case** (sensible à la casse), chaque mot de la réponse doit être exactement le même que le mot correspondant d'une des bonnes réponses.
- Avec le type **nocase**, la comparaison se fera sans tenir compte des différences entre lettres majuscules et lettres minuscules.
- Dans le cas du type **atext**, la comparaison n'est faite que sur les éléments essentiels des textes : les différences majuscule/minuscule, certaines différences singulier/pluriel (s en fin de mot), les accents sur les lettres, les mots très communs (de, le, un, ...) sont ignorés.

EXEMPLE 1.6. Pour analyser la réponse à la question « quel est le nom de la monnaie américaine ? », si l'on utilise la commande `\answer{{dollar}}{type=atext}`, les réponses « le dollar » « dollar » mais aussi « dollars » ou « le dollars » sont acceptées.

On peut utiliser la commande `\answer{{dollar | le dollar }}{type=case}` pour n'autoriser que les réponses « dollar » et « le dollar ».

EXERCICE 1.7. Faire une variante de l'exercice 1.6 en demandant d'écrire l'unité d'une grandeur physique choisie au hasard parmi la liste précédente.

Indications : La commande `randomrow(\liste)` permet de sélectionner au hasard une ligne. La commande `position(\a,\b)` permet de trouver la position de `\a` dans la liste `\b`.

⁶Vous pouvez faire un copier-coller de ces données en allant dans la classe d'exemple « Développement de Ressources » du serveur `wims.auto.u-psud.fr` (cliquez sur le lien « classes d'exemple », puis sur le lien « Autre »)

NB : Lorsqu'on déclare une variable **A** comme une chaîne de caractères par la commande

```
\text{A = texte }
```

on ne peut pas utiliser certains caractères de ponctuations comme « , », « ; », « : » et « ? » ni « \ » : ce sont des caractères réservés. de ces caractères à la place. Dans le cas de « : », « ? » et « \ » on peut aussi utiliser la commande **asis()** qui empêche WIMS d'interpréter ce qui est mis entre les parenthèses :

```
\text{A = asis( texte ) }
```

Le type de réponse **raw** joue le même rôle pour empêcher toute analyse de la réponse : la chaîne de caractères entrée par l'utilisateur sera recopiée telle quelle dans la variable **\reply1**.

1.3.4. Réponse à choix multiples

1.3.4.1. Les types *menu*, *radio*, *click*, *checkbox*, *flashcard* et *mark*

Lorsqu'on veut faire afficher une liste de choix possibles et demander à l'utilisateur de sélectionner le ou les bons choix, on dispose :

- des types **menu**, **radio** et **click** s'il suffit de sélectionner un des bons choix pour que la réponse soit juste,
- des types **checkbox**, **flashcard** et **mark** s'il faut sélectionner tous les bons choix pour que la réponse soit juste,

La présentation des différents choix varie suivant le type choisi. Dans tous les cas, la syntaxe est :

```
\answer{text}{num_bon;liste_totale}{type=...}{option=...}
```

où

- **liste_totale** désigne la liste des énoncés proposés
- **num_bon** désigne la liste des numéros des bons choix (i.e. les places des bons choix dans **liste_totale** séparés par des virgules).

On peut ajouter des options pour les types **checkbox**, **menu** et **radio** afin préciser l'ordre dans lesquels les choix apparaîtront (**shuffle** pour un ordre aléatoire, **sort** pour l'ordre alphabétique). Dans le cas de **checkbox** et **mark**, l'option **split** permet de pondérer la note en fonction du nombre de bons choix et de mauvais choix (deux bons choix compensent un mauvais choix).

NB : Le type **click** retourne la position de la réponse choisie dans la liste des choix possibles : **\reply1** est alors de la forme **~k**. Par contre, les autres types de réponses à choix multiples retournent le texte de la réponse choisie.

EXEMPLE 1.7. Le type **mark** permet par exemple de faire des exercices de langues où on demande de cliquer sur des éléments d'une phrase. Comme exemple nous donnons le source d'un exercice qui demande de cliquer sur les articles de la phrase « le chien et le chat sont noirs ».

```
\title{Marquer les articles}
\text{phrase=le,chien,et,le,chat,sont,noirs}
\text{reponse=1,4}
\statement{Cliquer sur les articles de la phrase suivante :
  <p align="center">
    \embed{r1,1} \embed{r1,2} \embed{r1,3} \embed{r1,4}
    \embed{r1,5} \embed{r1,6} \embed{r1,7}
  </p>
}
```

```
\answer{}{\reponse;\phrase}{type=mark}{option=noanalyzeprint}
\feedback{\sc_reply1<1>}{Les articles précédent les noms communs
<font color=green>chien</font> et <font color=green>chat</font>}
```

NB :

- `r1` est une abbréviation de `reply1`.
- L'option `noanalyzeprint` supprime l'affichage automatique de la bonne réponse effectuée normalement dans l'analyse de la réponse. Elle peut être utilisée avec n'importe quel type d'analyse de réponse.
- On a utilisé la variable `sc_reply1` pour afficher un commentaire seulement lorsque l'utilisateur a donné une réponse incorrecte.

EXEMPLE 1.8. Les lignes de codes suivantes proposent un schéma pour un QCM utilisant le type **radio** : l'exercice pose une question choisie au hasard parmi trois questions, l'utilisateur a le choix entre plusieurs réponses (ici trois ou quatre choix suivant la question posée) : la réponse à `quest1` est `a1`, à `quest2` est `b2` et à `quest3` est `c3`.

```
\title{Schéma de QCM avec une bonne réponse}
Tableau des données : sur chaque ligne : énoncé suivi de la liste des différents choix et de la
position du bon choix
\matrix{ data = quest1, a1, a2, a3, 1
quest2, b1, b2, b3, b4, 2
quest3, c1, c2, c3, 3}
Choix au hasard d'une ligne
\integer{ L = randint(1..rows(\data)) }
La question posée
\text{ question = \data[\L ; 1]}
la liste des textes affichés (on extrait de la ligne \L, les éléments 2,3,..., jusqu'à l'avant dernier
désigné par l'indice -2)
\text{ list = \data[\L ; 2..-2] }
Le numéro du bon choix : le dernier élément de la ligne \L est désigné par l'indice -1
\integer{num=\data[\L ; -1]}

\statement{ \question : \embed{reply1} }

\answer{ }{ \num; \list }{ type = radio }{ option = shuffle }
L'ajout de l'option shuffle fait que les différents choix possibles apparaissent dans un ordre
aléatoire.
```

- EXERCICE 1.8. (1) Faire une variante de l'exercice 1.7 en utilisant successivement les types de réponses **click**, **menu**, **radio**. Utiliser la commande `\feedback{1 = 1}{\reply1}` pour voir comment la réponse de l'utilisateur est mémorisée.
- (2) Dans l'exercice que vous venez de faire avec le type **radio**, faites afficher un commentaire si l'utilisateur s'est trompé d'unité, pour indiquer quelle est la grandeur physique qui a pour unité celle sélectionnée par l'utilisateur.

Indications : On pourra utiliser `\feedback{a != b}{commentaire}` si `a` et `b` désigne les positions de la solution et de la réponse de l'utilisateur dans la liste des choix; le commentaire est affiché si ces deux positions sont différentes.

La réponse peut aussi être construite à partir de plusieurs éléments donnés dans une liste en les sélectionnant successivement. Si chaque élément de la liste ne peut être utilisé qu'une fois, on utilisera le type **dragfill**, sinon on utilisera le type **clickfill**.

NB : Si un élément intervient plusieurs fois dans la réponse, il apparaît plusieurs fois dans la liste proposée avec **dragfill** et une seule fois avec **clickfill**.

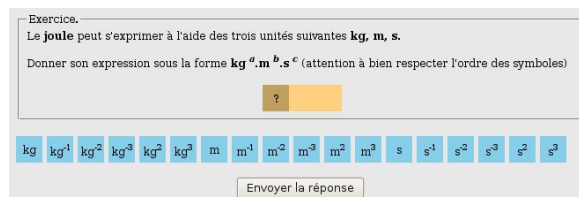


FIG. 1.5. Exemple d'énoncé utilisant **dragfill**

1.3.4.2. Les types *dragfill* et *clickfill*

La syntaxe pour l'analyse de la réponse est la même pour les deux types :

```
\answer{}{\sol;\liste_obj}{type=dragfill}
```

où `\liste_obj` est la liste des objets que l'utilisateur peut sélectionner et `\sol` désigne la solution : elle est constituée d'une liste d'items apparaissant dans `\liste_obj`.

EXERCICE 1.9. Reprendre l'exercice 1.8 et remplacer simplement le type de la réponse par le type **dragfill**. Tester votre exercice.

EXERCICE 1.10. Le code suivant permet de construire l'exercice dont l'énoncé apparaît figure 1.5. Analyser ce code et faire l'exercice en rajoutant à la fin la ligne `\feedback{1 = 1}{\reply1}` pour voir comment la réponse de l'utilisateur est conservée.

```
\title{Unités de grandeurs physiques I}
\matrix{liste=
fréquence,hertz,Hz,s\^({-1})\
force,newton,N,kg,m, s\^{(-2)}\
pression,pascal,Pa,kg,m\^{(-1)}\,s\^{(-2)}\
travail,joule,J,kg,m\^{(2)}\,s\^{(-2)}\
puissance,watt,W,kg,m\^{(2)}\,s\^{(-3)}\}
\text{quest = randomrow(\liste)}
\text{enonce = \quest[2]}
\text{symbol = kg,m,s,kg\^{(2)}\,m\^{(2)}\,s\^{(2)}\,kg\^{(3)}\,m\^{(3)}\,
s\^{(3)}\,kg\^{(-1)}\,m\^{(-1)}\,s\^{(-1)}\,kg\^{(-2)}\,m\^{(-2)}\,s\^{(-2)}\,
kg\^{(-3)}\,m\^{(-3)}\,s\^{(-3)}}
\statement{Le <b>\enonce</b> peut s'exprimer à l'aide
des trois unités suivantes <b>kg, m, s</b>.<p>
Donner son expression sous la forme
<b>kg\^{a}\.m\^{b}\.s\^{c}</b>
(attention à bien respecter l'ordre des symboles) </p>
<center>\embed{reply1,30 x 30 x 3}</center>
\answer{}{\quest[4..-1];\symbol}{type=dragfill}{option=sort}
```

1.3.5. Répondre par une formule mathématique

1.3.5.1. Fonction numérique : le type *function*

La réponse est évaluée en tant que fonction et la comparaison avec la solution est effectuée dans un intervalle $[a, b]$ défini dans l'entête de l'exercice par la commande `\range(a,b)` avec une tolérance définie par la commande `\precision`. Par défaut, la comparaison se fait sur l'intervalle $[-5, 5]$.

EXEMPLE 1.9. Supposons que la variable `f` soit définie par `\function{f = 5*x}` et qu'on analyse la réponse à l'aide de :

```
\answer{y =}{\f}{type = function}
```

Les réponses `5x`, `5*x`, `5*x + 0.000001` seront considérées comme justes. Si l'utilisateur entre la réponse `5*t`, un message d'erreur s'affichera « Votre réponse `5*t` n'est pas compréhensible, veuillez resoumettre la réponse ».

Dans la commande `\answer`, on peut ajouter après la solution, une liste de symboles qui peuvent être utilisés dans l'écriture de la réponse.

EXEMPLE 1.10. Supposons que la variable `g` soit définie par :

```
\function{g = randint(0..2)*x + randint(-1..1)*t}
```

On analysera la réponse à l'aide de :

```
\answer{y =}{\g,x,t}{type = function}
```

Même si `\g` est la fonction `2*x`, les réponses utilisant les inconnues `x` et `t` seront acceptées et comparées à la solution.

NB : Dans le cas d'une expression mathématique dont les coefficients sont des entiers ou des fractions, on peut aussi utiliser le type **formal** ; avec le type **formal**, seules les réponses numériquement identiques à la solution sont considérées comme bonnes.

1.3.5.2. Expression mathématique (type *algexp* et *litexp*)

Les types **algexp** et **litexp** permettent de demander à l'utilisateur de faire des opérations élémentaires sur les expressions algébriques comme factoriser un polynôme ou réduire une fraction rationnelle.

La réponse sera comparée avec la solution selon différents critères d'identification. Dans la commande `\answer`, on peut mettre plusieurs bonnes réponses, en les séparant par des virgules :

```
\answer{{liste_sol}}{type = algexp}
```

- Avec le type **litexp** (expression littérale), la comparaison sera littérale sans aucune simplification algébrique. Par exemple, $x + y$ ne sera pas identifié à $y + x$, ni $3/2$ avec $6/4$. Mais, $2x$ et $2*x$ seront considérés comme identiques, et les espaces seront écrasés avant la comparaison. Si la solution est $x^2 + 3$, la réponse $x*x + 3$ ne sera pas acceptée : un message d'erreur invitera l'utilisateur à écrire sa réponse différemment. A utiliser avec beaucoup de précautions.
- Avec le type **algexp** (expression algébrique), en plus des expressions qui sont considérées comme identiques avec le type **litexp**, les réponses où les coefficients numériques ne sont pas simplifiés sont acceptées : par exemple, $(24 + 4)*x - 53$ sera accepté si la solution est $28*x - 53$. De plus, $x - y*y$ et $-y^2 + x$ seront considérés comme identiques. Par contre, $(x + 1)(x - 1)$ ne sera pas accepté quand la bonne réponse est $x^2 - 1$, ni $\sin(x)^2 + \cos(x)^2$ à la place de 1.

EXERCICE 1.11. Faire un exercice demandant de donner l'équation de la tangente à une courbe en un point donné. La courbe pourra être définie par un polynôme de degré deux dont les coefficients sont des nombres décimaux avec un chiffre après la virgule.

Indications : Pour afficher une formule mathématique, il suffit de mettre cette formule entre `\(\)`. Par exemple, `\(3*x^2\)` donnera l'affichage : $3x^2$. Il existe des commandes (voir tableau 3) faisant les opérations classiques sur les fonctions comme évaluer en un point, dériver, intégrer, simplifier son expression.

1.4. Analyser une réponse par des conditions

Si on ne veut pas que la réponse de l'utilisateur soit analysée automatiquement par WIMS en fonction du type choisi, on met à la place de la solution le nom d'une variable qui n'a pas été déclarée auparavant, ici `\var`. On utilise alors la commande

```
\answer{}{ \var }{ type =... }
\condition{commentaire}{conditions portant sur \var}
```

Dans la première accolade, on peut mettre un texte qui sera affiché lors de l'analyse de la réponse. Dans la seconde accolade, on met la liste des conditions que la réponse de l'utilisateur contenue dans `\var` doit satisfaire pour être considérée comme bonne. Le tableau à l'annexe 1.1, page 53 décrit les différentes possibilités pour comparer deux chaînes de caractères.

EXEMPLE 1.11. Dans un exercice qui demande un entier plus grand que 10 et multiple de 3, on pourra analyser la réponse par :

```
\answer{}{\rep}{ type = numeric }
\integer{ v1 = pari(\rep == round(\rep)) }
\if{ \v1 = 1 }{ \integer{ v2 = pari(\rep%3==0) }}
{ \integer{ v2 = 0 } }

\condition{ Vous avez respecté les consignes }{ \v2 = 1 and \rep > 10 }
```

La variable `rep` ne doit pas avoir été déclarée auparavant. Elle contiendra la réponse de l'utilisateur.

NB : On a utilisé le logiciel PARI/GP⁷ pour calculer la partie fractionnaire de `\rep`, puis pour vérifier si `\rep` est divisible par 3.

On peut aussi tester les conditions séparément, par exemple en remplaçant la dernière ligne par

```
\condition{ Vous avez donné un entier supérieur à 10 }{ \v1 = 1 and \rep > 10 }
\condition{ Vous avez donné un entier multiple de 3 }{ \v2 = 1 }
```

Dans ce dernier cas, si l'utilisateur a entré un entier supérieur à 10 mais qui n'est pas multiple de 3, sa note ira de 5 à 0 suivant le niveau de sévérité choisi dans le menu de configuration de l'exercice.

NB :

- En ajoutant l'accolade `{option=hide}` à la fin de la commande `\condition`, le résultat du test effectué par cette commande ne sera pas affiché. On peut aussi ajouter une accolade `{weight=un_nombre}` qui permet de pondérer les conditions pour la notation finale.
- Lorsqu'on utilise l'option `noanalyzeprint` avec la commande `\answer`, l'analyse des conditions n'est pas non plus affichée au moment de l'analyse de la réponse.

⁷Le paragraphe 2.5 explique comment faire appel à d'autres logiciels.

- Le contenu de la variable `sc_reply1` est vide dans le cas où la réponse est uniquement analysée à l'aide de conditions.

EXERCICE 1.12.

Faire une variante de l'exercice 1.5 pour que seules les réponses données avec une précision relative de 0.001 soient considérées comme bonnes et faire afficher un message différent suivant que la valeur fournie est trop petite ou trop grande.

CHAPITRE 2

Des outils pour développer des exercices OEF

2.1. Utilisation de commandes WIMS

Certaines des commandes que l'on trouve dans la documentation technique « WIMS technical documentation » (ou « Doc technique ») sont utilisables à travers la commande

```
wims(nom paramètres)
```

et donnent de grandes facilités de manipulation de texte. On peut accéder à cette documentation technique en cliquant sur le lien « Doc technique » apparaissant dans le menu de gauche de la page d'entrée du site WIMS puis en cliquant sur « List of commands » ; attention toutes les commandes présentées dans cette liste ne sont pas utilisables dans la version des exercices OEF.

EXEMPLE 2.1.

```
\text{ A = 3,6,8,9,2 }  
\text{ T = wims(replace internal , by in \A) }
```

\T contient le mot 36892

EXEMPLE 2.2.

```
\text{ L = wims(makelist reply x for x = 1 to 5) }
```

\L contient la liste reply 1,reply 2,reply 3,reply 4,reply 5

NB : Dans la documentation technique, la description de la syntaxe de la commande `makelist` est :

```
Syntax: !makelist templ for v = v1 to v2 [step st ], or  
!makelist templ for v in v1,v2,v3,...
```

Dans les exercices oef, la syntaxe de cette commande doit être transposée ainsi :

```
Syntax: wims(makelist templ for v = v1 to v2 [step st ]), or  
wims(makelist templ for v in v1,v2,v3,...)
```

Les crochets servent à distinguer les paramètres optionnels ; les crochets ne sont pas à mettre si on utilise un paramètre optionnel. Par exemple pour faire une liste des entiers impairs de 1 à 10, on peut utiliser :

```
wims(makelist x for x = 1 to 10 step 2)
```

EXERCICE 2.1. Reprendre le code source donné dans l'exercice 1.10 et changer l'analyse de la réponse afin que l'utilisateur puisse entrer les symboles dans l'ordre qu'il veut.

Indications : On analysera la solution en utilisant `\condition` en ordonnant les éléments constituant la réponse de l'utilisateur par ordre alphabétique à l'aide de la commande `wims sort` (voir tableau 1). Le test `issametext b` permet de comparer les chaînes de caractères `a` et `b` (voir tableau 1.1).

EXERCICE 2.2. Faire un exercice donnant un nombre divisible par 9 et 11 présenté avec un chiffre manquant et demander le chiffre manquant. Un exemple d'énoncé pourrait être :

« Déterminer le chiffre manquant x pour que $4x085$ soit un multiple de 99. »

Nous donnons dans le tableau 1 les commandes les plus utiles au démarrage.

TAB. 1. Quelques commandes `wims()`

<code>wims(values x^2 + x + 1 for x = 1 to 20)</code>	permet de calculer des valeurs (plus efficace qu'une boucle)
<code>wims(makelist r x for x = 1 to 20)</code>	permet de faire des listes sans aucune évaluation (plus efficace qu'une boucle)
<code>wims(sort numeric \liste)</code>	range dans l'ordre croissant la liste de nombres <code>\liste</code>
<code>wims(replace item number 2 by x + 1 in \liste)</code>	remplace le deuxième élément de la liste par <code>x + 1</code> ; <code>item</code> peut être remplacé par <code>char</code> (caractères), <code>word</code> (mot), ou <code>line</code> (ligne).
<code>wims(append item xxx to \liste)</code>	rajoute à la liste <code>\liste</code> l'élément <code>xxx</code>
<code>wims(nonempty items \liste)</code>	enlève les éléments vides de la liste, par exemple <code>a, , b</code> devient <code>a,b</code> ;
<code>wims(nospace \variable)</code>	enlève tous les espaces dans <code>\variable</code>
<code>wims(listuniq \liste)</code>	enlève les répétitions en réordonnant les éléments de la liste d'une certaine manière.
<code>wims(text remove ^,; in \variable)</code>	enlève les caractères <code>^,;</code> de la chaîne de caractères <code>\variable</code>
<code>wims(text select aeiouy in \variable)</code>	garde uniquement les caractères <code>aeiouy</code> de la chaîne de caractères <code>\variable</code>
<code>wims(embraced randitem \le {chien,chat} marche sur {la route,le chemin}.)</code>	prend au hasard un des items de chaque accolade

2.2. Utilisation de macros de la `slib`

Les **slib** sont des macros que certains développeurs ont décidé de mettre à la disposition de tous. Ils correspondent la plupart du temps à un problème concret qu'ils ont eu. Ces **slib** sont classés par thèmes et sont documentés. Nous en donnons un exemple, en expliquant comment lire la documentation. On trouve le lien soit en passant par la documentation de **Createxo**, soit en passant par la documentation technique.

1	Name	list/selshuf
2	Effect	Selective shuffle
3	Call from module	<code>!readproc slib/list/selshuf [parameters]</code>
4	Call from OEF / DOC	<code>slib(list/selshuf [parameters])</code>
5	Parameters	Up to 3, comma-separated
6	Parameter 1	n
7	Parameter 2	m (default : n)
8	Parameter 3	k (default : n)
9	Output	the shuffled list (empty if error)
10	Comment	Selective shuffle : let $1 \leq m \leq n$ and $1 \leq k \leq n$. Output a shuffled list of m non-repeating random integers within $1, 2, \dots, n$, that always contains k . n is limited to 100.
11	Examples OEF	<code>\text{A = slib(list/selshuf 10, 7, 2)}</code>
12	Output	3,2,9,4,6,5,1

La ligne 1 donne le nom de la **slib**, sous forme d'adresse. La ligne 3 donne la commande utilisable dans le langage WIMS et la ligne 4 la syntaxe d'utilisation pour les exercices OEF ou les documents (remarquer que comme le montre l'exemple, `[parameters]` doit être remplacé par les paramètres et donc que l'on ne met pas les crochets) La ligne 5 indique le nombre de paramètres et les lignes suivantes, ici de 6 à 8, les paramètres possibles. Ici le paramètre 2 est par défaut le même que le paramètre 1. Ainsi,

```
\text{A = slib(list/selshuf 10)}
```

est équivalent à

```
\text{A = slib(list/selshuf 10,10,10)}
```

Les lignes 9 et 10 essaient d'expliquer ce que fait la commande. Enfin, la ligne 11 donne un exemple et la ligne 12 le résultat. Ici, il s'agit de tirer au hasard m nombres différents compris entre 1 et n et contenant toujours l'entier k . Vous êtes d'autre part prévenu que si vous entrez n'importe quoi comme paramètres, par exemple si $n < m$, il n'y aura rien comme sortie !

EXEMPLE 2.3. Le type d'analyse de réponse **sigunits** permet de demander une réponse avec son unité et un nombre de chiffres significatifs fixé.

```
\answer{}\rep\type=sigunits}
```

La variable **rep** correspond à la bonne réponse avec son unité et le bon nombre de chiffres significatifs. Par exemple, si le contenu de **rep** est 1.00 m, la réponse 0.00100 km est considérée comme une bonne réponse mais pas la réponse 0.001 km. La **slib text/sigunits** permet de transformer le résultat d'un calcul pour qu'il ait le bon nombre de chiffres significatifs et permet aussi de faire une conversion dans une autre unité.

Par exemple, la commande `\text{rep=slib(text/sigunits 3.33333 km,2,m)}` affecte le texte 3.3e3 m à la variable **rep**.

EXERCICE 2.3. Faire un exercice qui donne la vitesse d'un véhicule en km par heure et une durée en secondes, puis demande la distance que parcourt ce véhicule pendant cet durée (on fera en sorte que la réponse affichée soit donnée en mètres).

EXERCICE 2.4. Chercher la documentation de **slib** et regarder l'aide des **slib** suivantes : **lang/frapostrophe**, **matrix/non0**, **stat/binomial**, **utilities/nopaste**.

2.3. La présentation de l'énoncé

2.3.1. Quelques balises html

Faire un paragraphe	<code><p>texte</p></code>
Passer à la ligne	<code>
</code>
Insérer une ligne blanche	<code><hr></code>
Centrer	<code><center>texte</center></code>
Mettre en gras	<code>texte</code>
Mettre en italique	<code><i>texte</i></code>
Mettre en rouge avec une font plus petite	<code>texte</code>
Mettre en exposant ou en indice	<code><sup>texte</sup></code> ou <code><sub>texte</sub></code>
Une liste d'items	<pre> item item </pre>
Une liste numérotée	<pre> item item </pre>
Un exemple de tableau à fond jaune de la forme :	<pre> <table align="center" border=1 bgcolor="yellow" width="50%"> <tr align="center"> <td>1</td> <td>2</td> </tr> <tr align="center"> <td>3</td> <td>4</td> </tr> </table> </pre>

EXERCICE 2.5. Construire l'exercice dont l'énoncé serait par exemple :

On a fait un test sur 77 échantillons de sang dont 16 contenaient une substance interdite X. Le test s'est révélé positif pour 6 échantillon(s) contenant la substance X et le test s'est révélé négatif pour 26 échantillon(s) ne contenant pas la substance X. Compléter le tableau des effectifs :

	Avec X	Sans X
Test positif		
Test négatif		

Il est bien sûr possible de définir un style css pour personnaliser la présentation de son exercice :

EXEMPLE 2.4.

Dans l'exercice suivant, la consigne est écrite en bleu foncé sur un fond jaune, les deux équations à résoudre sont encadrées en rouge et l'utilisateur est invité à entrer sa solution à droite de chaque équation. La solution est affichée en gras et en vert.

```
\title{Ex de styles css}
\text{style_consigne = background-color: #FFFFCC;
      color: navy;
      margin: 2% 5%;
      padding: 1%; }
\text{style_cadre = border-style:none;
      border-width:10px;width:70% }
\text{style_cellule = border-style:solid;
      width:30%;
      border-width:1px;
      border-color:red; }
\text{style_sol = color:green;
      font-size:100%;
      font-weight: bold; }
\text{style = <style type="text/css">
      .consigne {\style_consigne}
      .cadre {\style_cadre}
      .red {\style_cellule}
      .sol {\style_sol}
      </style> }

\statement{ \style
  <div class="consigne"> Résoudre les deux équations suivantes :</div>
  <center><table class="cadre"><tr align="center">
    <td class="red"> 2x - 1 = 0 </td>
    <td> a pour solution x = \embed{reply1,10}</td></tr>
    <tr align="center"><td class="red"> 4y + 3 = y </td>
    <td> a pour solution y = \embed{reply2,10}</td></tr>
  </table></center> }

  \answer{x=}{0.5}{type = numexp}
  \answer{y=}{-1}{type = numexp}
  \solution{<p class="sol"> 2x - 1 = 0 a pour solution x = 0.5 et
    4y + 3 = y a pour solution y = -1</p>}
```

Ici, `\style` rajoute la commande de style css dans la page html créée.

2.3.2. Disposition personnalisée des champs de réponses dans le cas d'un type à choix

Si on veut que l'utilisateur sélectionne la réponse parmi une liste de n choix, la commande `\embed{reply1}` met les choix les uns après les autres. D'autre part, une fois la réponse donnée, seul le choix sélectionné apparaît. Pour une meilleure présentation ou une présentation adaptée au contexte de l'énoncé, on utilise la possibilité suivante : le i -ième choix peut être extrait comme

```
\embed{reply1, i}
```

On peut alors présenter les choix comme une énumération

```
<ul>
  <li> \embed{reply1,1}</li>
  <li> \embed{reply1,2}</li>
  <li> \embed{reply1,3}</li>
  <li> \embed{reply1,4}</li>
</ul>
```

ou dans un tableau

```
<table border=1>
  <tr>
    <th> A</th>
    <th> B</th>
    <th> C</th>
    <th> D</th>
  </tr>
  <tr>
    <td> \embed{reply1,1}</td>
    <td> \embed{reply1,2}</td>
    <td> \embed{reply1,3}</td>
    <td> \embed{reply1,4}</td>
  </tr>
</table>
```

Dans le cas d'un nombre variable de choix, il est recommandé d'utiliser une boucle (voir section 2.4), par exemple

```
<ul>
  \for{h = 1 to \n}{<li> \embed{reply1,\h}</li>}
</ul>
```

2.4. Les boucles et les branchements

L'utilisation des boucles et des branchements permet de construire des exercices plus évolués.

2.4.1. Conditions de test

Les opérateurs logiques sont **and** et **or**. La liste des conditions est donnée dans l'annexe 1.1.

EXERCICE 2.6. Pour chacune des lignes suivantes, chercher un test qui considérera les deux chaînes de caractères de la ligne comme identiques

vrai	Vrai
C'est faux	C' est faux
C'est faux	c' est faux
faux	c'est faux
purée,pommes de terre	purée
5,6	5
5 + 6	11
x	exp(x)
y	exp(y)

Faites afficher le résultat de ces tests.

2.4.2. La commande si ... alors ... (sinon) ...

```
\if{ conditions }
{ instructions } réalisées si conditions vraies
{ instructions } réalisées si conditions fausses
```

La troisième accolade est facultative.

Dans les parties **Avant** et **Après** (définition de variables), les **instructions** sont de la forme usuelle, il doit donc y avoir les commandes de déclaration : `\integer{n = 0 }`.

Dans la partie **Pendant** et en général dans toute zone **html** (`\feedback`, `\hint`, `\help`), on écrit directement le texte.

Lorsqu'une variable peut être définie de deux façons différentes suivant qu'une condition est satisfaite ou non, on peut aussi utiliser la syntaxe suivante :

```
\type{variable = condition ? valeur1 : valeur2}
\type{variable = condition ? valeur1}
```

Dans la deuxième ligne, `\variable` n'est pas modifiée si `condition` n'est pas vérifiée.

EXEMPLE 2.5. Pour définir deux variables `nom1` et `nom2` contenant aléatoirement soit Jean et Paul respectivement, soit Paul et Jean :

```
\text{nom1 = random(Jean,Paul)}
\if{\nom1 issametext Jean}{\text{nom2 = Paul}}{\text{nom2 = Jean}}
```

On peut remplacer la ligne d'instructions avec `\if` par :

```
\text{nom2 = \nom1 issametext Jean ? Paul : Jean}
```

EXEMPLE 2.6. En utilisant `\if`, on peut contrôler dans l'énoncé le singulier/pluriel d'un mot selon une variable aléatoire `n`.

```
\integer{n = randint(1..2)}
\statement{
  \n \if{ \n = 1 }{essai}{essais}
}
```

EXERCICE 2.7. Modifier l'énoncé de l'exercice construit en 2.5 pour que les pluriels ne soient mis que si c'est nécessaire.

2.4.3. La boucle for

```
\for{ j = 1 to \n } { instructions }
```

Les instructions écrites dans l'accolade sont exécutées $\backslash n$ fois avec une valeur de j qui augmente de un après chaque exécution.

EXEMPLE 2.7. Calcul de la moyenne de $\backslash k$ chiffres choisis aléatoirement et avec remise entre -10 et 10 :

```
\integer{k = randint(10...20)}
\text{U = slib(stat/random \k,-10,10,Z)}
\rational{s = 0}
\for{i = 1 to \k}
  {\rational{s = \s + \U[\i]}}
\rational{moy = \s/\k}
```

On aurait pu aussi utiliser la slib `stat/arithmean` pour calculer la moyenne.

NB : On peut spécifier aussi le pas d'incrément : par exemple,

```
\for{ j = 1 to 10 step 2 } {\j}
```

affiche 1 3 5 7 9.

EXERCICE 2.8. En utilisant les données suivantes,

Maladie infectieuse	vecteur	Maladie infectieuse	vecteur
Tuberculose	bactéries	Tétanos	bactéries
Typhoïde	bactéries	Lèpre	bactéries
Rage	virus	Poliomyélite	virus
Rougeole	virus	Hépatite	virus
Grippe	virus	Bronchiolite	virus
Paludisme	parasites	Toxoplasmose	parasites

faire un exercice de ce type :

Exercice.

Les maladies infectieuses peuvent être dues à des virus, des bactéries ou des parasites qui se multiplient dans l'organisme.

Parmi les maladies suivantes, sélectionner toutes les maladies qui sont dues à des virus :

- ☐ Toxoplasmose
- ☐ Rougeole
- ☐ Tuberculose
- ☐ Tétanos
- ☐ Rage
- ☐ Grippe

☐ ces maladies ne sont pas dues à des virus

Envoyer la réponse

Indications : On pourra utiliser le type de réponse *checkbox*.

2.4.4. La boucle tant que

```
\while{ condition }{ instructions } réalisées tant que condition vraie
```

Pour être sûr que la boucle s'arrête et ne dure pas trop longtemps, mettez un test d'arrêt.

2.5. L'utilisation de logiciels extérieurs

WIMS peut appeler un certain nombre de logiciels extérieurs dont PARI/GP, MAXIMA, OCTAVE, GAP, POVRAY et moins connu float_calc (qui est en fait le logiciel bc). En général, cet appel se fait de la manière suivante

```
\text{ a = wims(exec nom_logiciel lignes_instructions) }
```

Il y a deux cas particuliers : PARI/GP et MAXIMA.

```
\text{ a = pari( lignes_instructions ) }
\text{ a = maxima( lignes_instructions ) }
```

Si vous ne disposez pas de ces logiciels, vous pouvez les utiliser et faire vos tests à travers l'outil « Direct exec »¹ de WIMS.

EXEMPLE 2.8.

```
\integer{ dspace = pari(P = Mat([\P]);matrank(P)) }
```

La variable dspace contient le rang de la matrice \P.

EXEMPLE 2.9.

```
\function{ f = 5 * y * (2 * x - 3) + 1 }
\text{ f = maxima(expand(\f)) }
```

La variable f contient alors la chaîne de caractères $10 * x * y - 15 * y + 1$.

Le logiciel reste ouvert entre deux exécutions et l'on peut donc garder des routines ou des variables "ouvertes".

EXEMPLE 2.10. On utilise OCTAVE ci-dessous pour résoudre numériquement l'équation $\frac{dx(t)}{dt} = 2tx(t)$ à des points qui discrétisent l'intervalle $[-10,10]$ et avec la condition initiale $x(0) = 1$, puis la condition initiale $x(0) = 2$.

```
\text{ a = wims(makelist x/10 for x=-10 to 10) }
\text{ fonct = 2*t*x }
\text{ reponse1 = wims(exec octave
function
  xdot = f(x,t);
  xdot = \fonct;
endfunction;
lsode("f",1,[\a])) }
\text{ reponse2 = wims(exec octave
lsode("f",2,[\a])) }
\matrix{ reponse1 = slib(text/octavematrix \reponse1) }
\matrix{ reponse2 = slib(text/octavematrix \reponse2) }
```

NB : On utilise la slib text/octavematrix pour récupérer la liste des valeurs comme une matrice.

¹Pour obtenir « Direct exec » à partir de Createxo, cliquer sur « Testeur de script » au-dessus de l'éditeur du mode brut

CHAPITRE 3

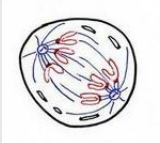
Exercices à base d'images ou de dessins

3.1. Mettre une image dans un exercice

Pour utiliser dans un exercice une image qui se trouve sur votre ordinateur, vous devez d'abord mettre cette image sur le serveur WIMS. La figure 3.1 décrit les étapes pour créer un exercice faisant appel à une image. Les commandes à utiliser dans le source de l'exercice sont présentées dans les exemples qui suivent.

EXEMPLE 3.1. L'exercice suivant porte sur l'identification d'une image choisie au hasard dans une liste de sept images :

Exercice.
Le schéma suivant



décrit une cellule animale dans une des phases du cycle cellulaire. Laquelle ?

☐ l'anaphase, ☐ l'interphase, ☐ la cytotdiérèse, ☐ la métaphase, ☐ la prométaphase, ☐ la prophase, ☐ la télophase

NB : les choix sont présentés dans l'ordre alphabétique

```
\title{Schémas du cycle cellulaire}
\text{phases = l'interphase,la prophase,la prométaphase, la métaphase,
l'anaphase, la télophase,la cytotdiérèse}
\integer{n = randint(1..7)}
\text{liste = interphase.jpg,prophase.jpg, prometaphase.jpg, metaphase.jpg,
anaphase.jpg,telophase.jpg,cytodierese.jpg}
\text{choix = \liste[\n]}

\statement{Le schéma suivant
<center> <img src=\imagedir/\choix> </center>
décrit une cellule animale dans une des phases du cycle cellulaire. Laquelle ?
<p><center>\embed{reply1}</center></p>
<i> NB : les choix sont présentés dans l'ordre alphabétique</i>}

\answer{{\n;\phases}{type = radio}{option = sort}}
```

NB : La variable `imagedir` est une variable créée par WIMS et dans laquelle se trouve l'adresse html du dossier contenant l'image que vous avez chargée.

(a) Commencez par créer l'exercice qui utilisera une image : étapes (a) à (c) de la création d'un exercice (figure 1.3); ne mettez éventuellement que le titre et le `statement{ }`,

Vous êtes en mode brut de création d'exercices. Entrez le source entier (format OEF) de votre exercice : (pour enregistrer un fichier source existant, copiez le fichier dans la fenêtre.) Vous pouvez tester certains programmes externes avec le [testeur de script](#).

```

\title{Schémas du cycle cellulaire}
\text{phases = {interphase, la prophase, la métaphase, la télophase, la cytokinèse}}
\integer{n = randint(1..7)}
\text{liste = {interphase.jpg, prophase.jpg, metaphase.jpg, telophase.jpg, cytokinèse.jpg}}
\text{choix = \listel(n)}

\statement{Le schéma suivant
<center></center>
décrir une cellule animale dans une des phases du cycle cellulaire.
Laquelle ?
<p><center>\embed{reply}</center></p>
<i>NB : les choix sont présentés dans l'ordre alphabétique</i>}

\answer{H\n, \phases[type = radio]option = sort}
  
```

[Valider] [Abandonner les modifications].

(b) Cliquez sur le bouton « Envoyer le source »

Le logiciel a reconnu votre exercice avec succès, avec les informations suivantes. (Vérifiez avec ce que vous avez défini.)

Titre Schémas du cycle cellulaire
 Nombre de paramètres 5
 Nombre de réponses libres 1
 Nombre de choix multiples 0
 Maintenant vous pouvez :

- [tester votre exercice.](#)
- [modifier le source de l'exercice.](#)
- [déposer un fichier \(image, pdf, ...\) dans l'exercice.](#)
- [mettre cet exercice dans votre classe.](#)
- [sauvegarder le source](#) de l'exercice sur votre ordinateur !

(c) Cliquez maintenant sur « déposer un fichier (image, pdf, ...) ».

Vous pouvez charger différents types de fichiers chargés (image, mp3, pdf, ...).

Fichier à déposer dans l'exercice : Parcourir... OK

Cet exercice n'a pas encore de fichiers binaires.

(d) Cliquez sur le bouton « Parcourir » et sélectionnez l'image à envoyer. Puis cliquez sur le bouton « OK ». L'image sera accessible pour cet exercice-là, mais pas pour un autre.

Vous pouvez charger différents types de fichiers chargés (image, mp3, pdf, ...).

Fichier à déposer dans l'exercice : Parcourir... OK

Voici la liste des fichiers associés à l'exercice.

Nom du fichier	Aperçu	
anaphase.jpg		Effacer

anaphase.jpg On peut accéder à ces fichiers dans l'énoncé de l'exercice via le paramètre interne `\imagedir`. Par exemple, vous pouvez écrire

```

```

Pour des fichiers d'images, vous pouvez écrire ``, mais aussi `<img(\imagedir/anaphase.jpg) ou <img(\imagedir/anaphase.jpg|html options)>`. L'avantage de cette dernière méthode est que les étudiants ne verront pas le nom du fichier. À cause de son coût en performance, n'utilisez cette méthode que lorsque c'est nécessaire.

Pour d'autres types de fichiers, les commandes html doivent être adaptées : `file.pdf ...`

Si vous avez besoin d'autres images pour cet exercices, recommencez cette étape autant de fois que nécessaire.

(e) A ce stade, les images sont visibles sur le serveur mais ne sont pas sauvegardées dans la classe. Cliquez sur le lien « Retourner à la page de création d'exercices ».

Le logiciel a reconnu votre exercice avec succès, avec les informations suivantes. (Vérifiez avec ce que vous avez défini.)

Titre Schémas du cycle cellulaire
 Nombre de paramètres 5
 Nombre de réponses libres 1
 Nombre de choix multiples 0
 Maintenant vous pouvez :

- [tester votre exercice.](#)
- [modifier le source de l'exercice.](#)
- [déposer un fichier \(image, pdf, ...\) dans l'exercice.](#)
- [mettre cet exercice dans votre classe.](#)
- [sauvegarder le source](#) de l'exercice sur votre ordinateur !

(f) Cliquez sur le lien « mettre cet exercice dans votre classe ». Les images sont maintenant sauvegardées comme des images attachées à l'exercice que vous venez de créer.

Votre exercice est mis en place avec succès. Cet exercice sera accessible uniquement aux participants de votre classe et uniquement si vous le mettez dans une feuille de travail.

Vous pouvez maintenant :

- [consulter les exercices de la classe.](#) (et éventuellement en mettre dans vos feuilles de travail.)
- [retourner à l'étape précédente.](#)

(g) Cliquez enfin sur « Consulter les exercices de la classe ».

FIG. 3.1. Les étapes pour créer l'exercice de l'exemple 3.1

NB : Si vous avez la curiosité de regarder le code source de la page html (accessible dans tous les navigateurs), vous verrez que le nom de l'image y est écrit explicitement. Cela peut donner des indications pour la réponse. Pour éviter cela, on peut remplacer la commande `` par `\img{\imagedir/\choix}` qui renomme l'image en lui donnant un nom choisi aléatoirement et donc différent à chaque renouvellement de l'exercice¹.

¹Commande utilisable uniquement dans l'énoncé, on la remplace sinon par la commande `imgrename` (voir exemple 3.2). Pour l'instant, même avec cette commande, les noms des images apparaissent dans le code source de la page html pour les exercices créés dans la classe

EXERCICE 3.1. Modifier le source de l'exercice de l'exemple 3.1 afin d'afficher le schéma correspondant à la réponse de l'utilisateur lorsque celui-ci s'est trompé.

EXEMPLE 3.2. Vous pouvez facilement transformer le source de l'exemple précédent en un exercice de correspondance sur trois images.

```
\title{Schémas du cycle cellulaire 2}
\text{phases = interphase,prophase,prométaphase,métaphase,anaphase,télophase,
cytodierèse}
\text{liste = interphase.jpg,prophase.jpg,prometaphase.jpg,metaphase.jpg,
anaphase.jpg,telophase.jpg,cytodierese.jpg}
\text{choix = shuffle(7)}
\text{noms = \phases[\choix[1..3]]}
\text{liste = \liste[\choix[1..3]]}
\text{images = wims(makelist <img src=\imagedir/x> for x in \liste)}

\statement{
  Les schémas décrivent une cellule animale dans
  trois phases différentes du cycle cellulaire.
  Associer les noms des phases aux schémas :
  <p><center>\embed{reply1,150x150x100}</center></p>
}

\answer{}{\images;\noms}{type = correspond}
```

La commande `` doit être remplacée ici par la commande `imgrename(\imagedir/x)` si on veut que les noms des images n'apparaissent pas dans le source de la page html.

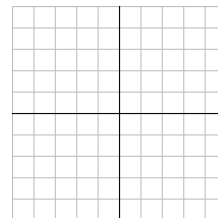
NB : L'espace avant la parenthèse dans `imgrename(\imagedir/x)` est important. Vous pouvez aussi ajouter des options de présentations des images (options acceptées par ``) en tapant par exemple `imgrename(\imagedir/ttt width=150 align=middle)`.

On remarque que la première ligne contient l'intervalle mathématique de l'axe des x , la seconde ligne l'intervalle mathématique de l'axe des y . Les lignes suivantes tracent une série de parallèles ; les deux dernières lignes une ligne horizontale et une ligne verticale passant par le point $(0,0)$. La commande dans le `\statement`

```

```

donne le dessin ci-contre :



NB : Il existe une slib `draw/repere` qui retourne les lignes de commande nécessaires pour dessiner un repère. Par exemple, un dessin analogue au précédent est obtenu par :

```
\text{dessin1 = slib(draw/repere 200,200,0,-5,5,-5,5)}
\text{\url=draw(200,200
\dessin1)}
```

EXERCICE 3.2. Faire un exercice proposant trois carrés de couleur différente à mettre en correspondance avec le nom de ces couleurs.

EXERCICE 3.3. Dans l'énoncé de l'exercice 1.11, rajouter un dessin de la courbe avec la position du point. Faire réapparaître le dessin de la courbe en y ajoutant la tangente dans la solution.

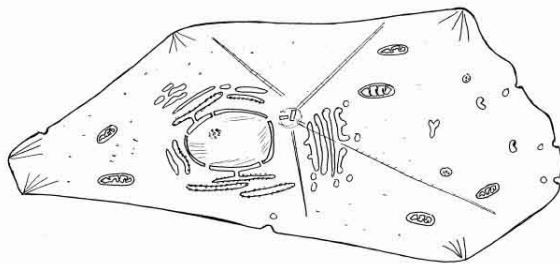
Indications : On pourra utiliser la slib `function/bounds` pour déterminer les bornes approximatives de la courbe sur un intervalle.

3.2.2. Dessiner sur une image

On peut ajouter des éléments à une image extérieure, comme une flèche pour désigner une zone de l'image.

On va donc maintenant rajouter des éléments en écrivant dans le source de l'exercice. On commence par créer une image vide en utilisant la commande `draw()` ; on crée ensuite une autre image (toujours avec la commande `draw()` dans laquelle on commence par intégrer l'image par exemple à l'aide de la commande : `copy 0,H,-1,-1,-1,-1,nom.jpg` où H est la hauteur de l'image en pixels. On rajoute à la suite de cette ligne les commandes de dessin que l'on désire. Si l'url de la figure est conservée dans la variable `fig`, alors la commande `` permet d'afficher le contenu de cette figure.

EXEMPLE 3.5. Voici un schéma d'une cellule :



Les commandes suivantes permettent d'ajouter une flèche bleue sur l'image de la cellule pour désigner certains organites de la cellule

```

\text{Size = 625,320}
\matrix{coord = 245,102,1,-1,un reticulum endoplasmique
50,107,1,1,un filament d'actine
120,107,-1,-1,une mitochondrie
70,180,-1,1,la membrane plasmique
238,159,1,-1,des nucléoles
358,190,1,1,l'appareil de Golgi
354,225,-1,1,une microtubule
542,207,-1,1,une vésicule
160,100,0,-1,le cytoplasme
}
\integer{k = rows(\coord)}
\integer{k = randint(1..\k)}
\text{co = \coord[\k;1]+20*(\coord[\k;3]),\coord[\k;2]+20*\coord[\k;4],
\coord[\k;1],\coord[\k;2]}
\text{dessinprelim=
xrange 0,\Size[1]
yrange 0,\Size[2]
copy 0,\Size[2],-1,-1,-1,-1,cellule.jpg
arrow \co,10,blue}
\text{figure = draw(\Size
\dessinprelim)
}

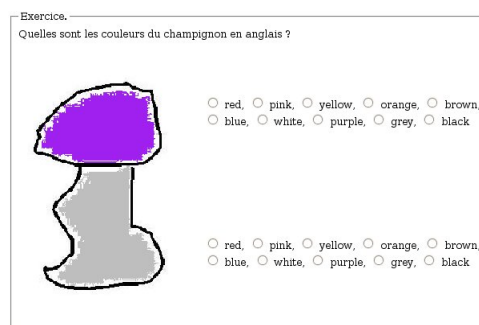
```

Ici, l'intervalle mathématique a été choisi comme la taille en pixels de l'image.

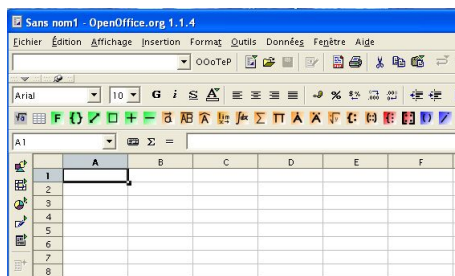
EXERCICE 3.4. Construire un exercice utilisant le dessin précédent pour poser une question. Compléter le source afin de rajouter l'enveloppe nucléaire aux organites sur lesquels peuvent porter les questions.

EXERCICE 3.5. Faire un exercice coloriant aléatoirement deux zones avec deux couleurs parmi **red**, **pink**, **yellow**, **orange**, **brown**, **blue**, **white**, **purple**, **grey**, **black** et demander quelles sont les couleurs. Une présentation de l'énoncé pourra être la suivante :

Indications : Vous devez d'abord trouver les coordonnées en pixels de l'intérieur des deux zones que vous avez à colorier. Pour cela, un outil commode est *Gimp*.



EXERCICE 3.6. En utilisant l'image suivante d'un tableur, remplir aléatoirement les cases du tableur et demander ce que contient une des cases.



On commence par préparer l'image. Si vous ne prenez pas celle qui vous est proposée, ouvrez votre tableur préféré et faites une photo d'écran.

Trouvez les dimensions de l'image (ne pas la prendre trop grande). Si vous êtes sur linux ou si ImageMagick est installé, vous pouvez taper la commande suivante

```
identify image.jpg
```

pour redimensionner

```
convert -resize 200 image.jpg image2.jpg
```

Vous devez aussi trouver la position et les dimensions de la première case (coordonnées en pixel du coin en bas à gauche et du coin en haut à droite) (Pour définir la liste des coordonnées des autres cellules, vous pouvez utiliser la commande `WIMSvalue`).

3.2.3. Insérer des champs de réponses sur une photo ou dans une figure

On peut placer des champs de réponses de type **dragfill** ou **clickfill** directement sur une photo ou dans un dessin à l'aide de la commande `\special{imagefill }`. Cette commande est à mettre dans la commande `\statement{ }`.

EXEMPLE 3.6. On désire un énoncé contenant une image `nom_photo.jpg` de taille 625x320 en pixels sur laquelle on positionne deux champs de réponses de taille 40x20 en pixels, la première à la position 100x200 en pixels et la deuxième à la position 200x50. L'utilisateur remplit les champs en choisissant parmi les éléments d'une liste appelée ici `listechoix`. Les variables `bonchoix1` et `bonchoix2` désignent les deux éléments de `\listechoix` qui constituent la réponse correcte pour chacun des deux champs.

```
\statement{Compléter l'image : <center>
\special{imagefill \imagedir/nom_photo.jpg,625x320,40x20
  reply1, 100x200
  reply2, 200x50
</center>
}
\answer{}{\bonchoix1; \listechoix}{type=clickfill}
\answer{}{\bonchoix2}{type=clickfill}
```

Pour faire la même chose sur une figure construite avec la commande

```
\text{dessin = draw(625,320
  ligne d'instruction
  ...
  ligne d'instruction)}
```

il suffit de remplacer `\imagedir/nom_photo.jpg` par `\dessin` dans les lignes de code précédentes.

EXERCICE 3.7. Dans l'exercice 3.5, faire mettre les couleurs en étiquette dans la zone de couleur.

NB : On peut avoir un champ de réponse de longueur plus grande que 1 (nombre de réponses possibles) : dans l'exemple suivant, 3 : `reply1,100x100x 3`.

3.3. Les types de réponse « graphiques »

3.3.1. Le type coord (clic sur image)

Le type **coord** permet de cliquer sur une image. Plusieurs analyses de la réponse sont possibles selon la zone acceptée comme juste (**point**, **rectangle**, **circle**, **polygon**, **ellipse**). Vous trouverez dans la documentation de **Createxo** (voir aussi le tableau 8) les différentes possibilités utilisables. La syntaxe générale est

```
\statement{\embed{reply1}
\answer{}{\url;type_zone,liste_par}{type=coord}
```

où

- **liste_par** est une liste de coordonnées en pixels caractérisant la zone.
- **\url** est l'url donné par la commande **draw(\SIZEx,\SIZEy)**.

Si vous mettez en fond une image déjà préparée et que vous avez téléchargée, mettez le nom de l'image **\imagedir/nom.jpg** par exemple.

EXEMPLE 3.7. Si l'utilisateur doit cliquer dans le disque de diamètre 10 pixels centré au point de coordonnées (100,200) en pixels (l'origine étant le sommet en haut à droite de l'image), la ligne de réponse doit être la suivante.

```
\answer{}{\imagedir/nom.jpg;circle,100,200,10}{type=coord}
```

NB : **\imagedir** est une variable prédéfinie qui donne la racine url de l'image.

EXERCICE 3.8. En reprenant l'image du tableur utilisée à l'exercice 3.6, faites un exercice demandant de cliquer dans une case.

3.3.2. Le type javacurve

En attendant d'utiliser GEOGEBRA, on peut déjà faire tracer quelques objets.

Le type **javacurve** utilise JAVA et permet de faire construire un objet de manière très simple sur un dessin construit par FLYDRAW ou sur une image chargée sur le serveur. On peut ainsi demander de sélectionner plusieurs points (contrairement au type **coord**), une droite, une demi-droite, un vecteur, un segment, un cercle, un polygone. Vous trouverez dans la documentation de **Createxo** (voir aussi le tableau 9) les différentes possibilités utilisables. La syntaxe est

```
\statement{ \embed{reply1,\SIZEx x \SIZEy} }
\answer{}{\url;sline,x1,y1,x2,y2}{type=javacurve}
\answer{}{\url;points,x1,y2,x2,y2,x3,y3}{type=javacurve}
```

L'url **\url** est l'adresse url donnée par la commande **draw(\SIZEx,\SIZEy)**. Si vous mettez en fond une image déjà préparée et que vous avez téléchargée, mettez simplement le nom de l'image **nom.jpg** par exemple. Vous devez réécrire la taille dans le deuxième paramètre de **\embed**. Les coordonnées dans la réponse sont en pixels. Si vous avez fait vos calculs en coordonnées mathématiques (**\rangex**, **\rangey**), vous devez donc faire la conversion, par exemple en utilisant la **slib draw/convpixel** :

```
\text{coord =
slib(draw/convpixel \xpoint,\ypoint,\SIZEx,\SIZEy,\rangex,\rangey,0,pixels)}
```

EXERCICE 3.9. Faire une variante de l'exercice 3.3 en demandant de tracer la tangente de la courbe au point donné.

EXERCICE 3.10. En optique, on peut tracer le rayon réfracté en utilisant la règle des sinus (construction de Descartes). Faire un exercice demandant de tracer le rayon réfracté en ayant affiché les cercles de rayon les indices.

Indications : *Cette construction repose sur le tracé des « cercles des indices ». On trace les deux cercles de rayon n_1 et n_2 centrés sur le point d'incidence (I). Le rayon incident (provenant du milieu 1) est prolongé dans le milieu 2 et coupe le cercle 1 en un point A dont la projection H est telle que, par construction, $IH = n_1 \sin i$. Pour satisfaire la relation de Snell-Descartes, le rayon réfracté doit couper le cercle 2 en un point B ayant même projection. Il suffit donc de prolonger la droite (AH) jusqu'à son intersection avec le cercle 2.*

CHAPITRE 4

Aides contextuelles et exercices à étapes

4.1. Aides dans le texte

Avant de nous lancer dans les exercices à étapes, nous allons voir les aides contextuelles : il s'agit d'aides pouvant être mises dans le texte de l'énoncé. Habituellement si vous utilisez le champ `\help{}`, un lien **Aide** apparaît en bas de la page ainsi que dans le menu d'en haut. L'aide contextuelle permet de mettre un lien sur un mot de l'énoncé. Le texte peut alors apparaître ou non dans les lieux habituels d'aide. Vous devez

- (1) indiquer le mot qui aura un lien permettant d'obtenir l'aide en utilisant la commande :

```
\special{help mot_court, mot(s)_sur_lequel_on_clique}
```

qui se met dans le `\statement{}` ou dans toute zone destinée à être sur une page html; cette commande donne naissance à une variable `help_subject` dont la valeur sera `mot_court` lorsque l'utilisateur cliquera sur `mot(s)_sur_lequel_on_clique`.

- (2) indiquer dans le champ `\help{...}` quand cette aide doit apparaître en mettant des conditions sur la variable `help_subject`.

NB : Par défaut, `\help_subject` est vide et le texte mis dans le champ `\help{...}` apparaît dans l'aide générale de l'exercice.

Donnons quelques exemples.

4.1.1. Une explication sur un mot de l'énoncé qui apparaît uniquement en cliquant sur ce mot

EXEMPLE 4.1. Dans l'exemple ci-dessous : une fenêtre contenant la phrase « Carabistouille est ma sorcière préférée » s'ouvre en cliquant sur le lien « Carabistouille ». De même, une fenêtre contenant la phrase « Mélusine est ma fée préférée » s'ouvre en cliquant sur le mot « Mélusine ».

```
\statement{
  Un jour, \special{help cara,Carabistouille}
  et \special{help melu,Mélusine} ...
}
\help{
  \if{ \help_subject issametext cara}{ Carabistouille est ma sorcière préférée.}
  \if{ \help_subject issametext melu}{ Mélusine est ma fée préférée.}
}
```

Par contre, le menu d'aide général de l'exercice ne contient rien.

4.1.2. Une aide qui apparaît quand on clique sur un mot ou sur l'aide générale

EXEMPLE 4.2. Par rapport au premier exemple, la phrase « Carabistouille est ma sorcière préférée » apparaîtra en plus dans l'aide générale de l'exercice :

```
\statement{
  Un jour, \special{help cara,Carabistouille}
  et \special{help melu,Mélusine} ...
}
\help{
  \if{ \help_subject issametext cara or \help_subject issametext }{
    Carabistouille est ma sorcière préférée.}
  \if{ \help_subject issametext melu}{ Mélusine est ma fée préférée.}
}
```

4.1.3. Une aide contenant elle-même une aide contextuelle

Par rapport, au premier exemple, on va ajouter un lien sur le mot « sorcière » de la phrase « Carabistouille est ma sorcière préférée » qui apparaît lorsqu'on clique sur le mot « Carabistouille » ; en cliquant sur le mot « sorcière », la phrase « Et la vôtre, comment s'appelle-t-elle ? » apparaît à la place de la phrase sur Carabistouille.

EXEMPLE 4.3.

```
\statement{
  Un jour, \special{help cara,Carabistouille}
  et \special{help melu,Mélusine} ...
}
\help{
  \if{ \help_subject issametext cara}{
    Carabistouille est ma \special{help sorc, sorcière} préférée.
  }
  \if{ \help_subject issametext melu}{Mélusine est ma fée préférée.}
  \if{ \help_subject issametext sorc}{Et la vôtre, comment s'appelle-t-elle ?}
}
```

NB : Si on veut que la même phrase d'aide apparaisse en cliquant à différents endroits du texte, il suffit de mettre un lien avec la commande `\special` en utilisant toujours le même mot court, à plusieurs endroits dans le texte.

4.1.4. Une aide qui dépend des données tirées pour réaliser l'exercice

Dans l'exemple suivant, un des deux noms Mélusine ou Carabouille apparaît aléatoirement dans l'énoncé, une aide contextuelle adaptée au nom qui apparaît dans l'énoncé est proposée :

EXEMPLE 4.4.

```
\title{Fée ou sorcière}
\integer{n = randint(1,2)}
\text{nom = Carabistouille,Mélusine}
```

```

\text{nom = \nom[\n]}
\text{Nom = cara, melu}
\text{Nom = \Nom[\n]}
\statement{ Connaissez-vous \special{help \Nom, \nom} }
\help{
  \if{ \help_subject issametext cara}{
    Carabistouille est ma \special{help sorc, sorcière} préférée.}
  \if{ \help_subject issametext melu}{
    Mélusine est ma \special{help fee, fée} préférée.}
  \if{ \help_subject iswordof fee sorc}{Et la vôtre, comment s'appelle-t-elle ?}
}

```

EXERCICE 4.1. Reprendre un exercice déjà fait et ajouter des aides sur certains mots.

4.1.5. Aide à la souris : utilities/tooltip

Plutôt que de faire apparaître les aides dans une nouvelle page, il est possible d'utiliser un outil : `tooltip`. Les indications apparaissent alors quand on passe la souris sur un mot.

Il est possible d'utiliser `tooltip` de deux manières différentes : soit en insérant directement à l'intérieur du champ `statement` en passant par la commande `special`, soit en utilisant un `slib`

```

\statement{
  \special{tooltip un pentagone,,polygone à 5 côtés}
}

```

EXEMPLE 4.5.

```

\title{Polygones}
\text{n=randint(3..7)}
\text{polygone=triangle,rectangle,pentagone, hexagone, septagone}
\text{aide_contextuelle=}
slib(utilities/tooltip \polygone[\n-2],[FONTSIZE, '8pt', ABOVE, 'false'],
  polygone à \n côtés)}
\statement{ \aide_contextuelle}

```

Comme on le voit dans cet exemple, il est possible de rajouter en deuxième position des options. Ce sont celles du javascript utilisé : <http://www.walterzorn.com> (voir la documentation de **Createxo**). Par exemple

```

\statement{
  \special{tooltip un pentagone,[FONTSIZE, '20pt',TITLE,'Aide',PADDING,9,BGCOLOR,'red'],
    polygone à 5 côtés}
}

```

augmente la taille des fontes, met un titre, laisse des marges autour du texte et met le fond en rouge.

NB : Dans un document, on peut utiliser la commande `\tooltip` dont les trois champs sont le texte sur lequel cliquer, les options et le texte à mettre dans la bulle d'aide.

NB : A chaque appel, le javascript utilisé est chargé. Or il n'est utile qu'une seule fois en haut de la page. Vous pouvez le faire charger en ne donnant pas d'argument. Ensuite, si le mot `nojs` est rajouté aux options, il ne sera pas rechargé. Par exemple,

```

\statement{
\special{tooltip un pentagone,nojs, polygone à 5 côtés}
\special{tooltip un triangle,nojs [FONTSIZE, '20pt',TITLE,'Aide',PADDING,9,BGCOLOR,'red'],
polygone à 3 côtés}
}

```

4.2. Les exercices à étapes

4.2.1. Le principe

Une étape est une partie de l'exercice qui a lieu entre deux requêtes de l'utilisateur. La première requête se situe au moment du clic **Renouveler l'exercice**, les autres à chaque clic sur **Envoyer votre réponse**. Tant que les étapes de l'exercice ne sont pas épuisées, l'utilisateur n'a pas de note visible.

Les commandes ou variables supplémentaires utilisables pour un exercice à étapes sont

<code>\steps{...}</code>	sert à définir les questions apparaissant à chaque étape ; doit être mis dans la partie Avant .
<code>\nextstep{...}</code>	sert à définir de manière dynamique les questions qui devront être posées ; dans la partie Avant .
<code>\step</code>	variable indiquant le numéro de l'étape ; utilisable Pendant et Après
<code>\conditions{...}</code>	commande permettant d'indiquer les numéros des conditions utiles pour l'exercice servant à contrôler les réponses de l'utilisateur.

Une seule de ces deux commandes `\steps` ou `\nextstep` peut être utilisée dans un exercice. On utilise `\steps` lorsque les questions qui vont être posées pendant tout l'exercice sont fixées avant l'apparition de l'énoncé. On utilise `\nextstep` au lieu de `\steps` si on veut pouvoir poser des questions différentes en fonction de la réponse de l'utilisateur à une étape précédente.

La variable `ETAPE` est créée automatiquement avec la valeur 1 dès qu'une des commandes `\steps` ou `\nextstep` est exécutée. Sa valeur augmente de 1 à chaque fois que l'utilisateur clique sur « Envoyer votre réponse ».

En général, on définit une variable de nom libre que l'on applique à `\steps` ou `\nextstep`. On l'appellera `ETAPE` dans les exemples.

Commençons en donnant deux exemples schématiques montrant comment utiliser `\steps` puis `\nextstep`.

EXEMPLE 4.6. Structure schématique d'un exercice à deux étapes où l'on pose deux questions à la première étape et une question à la seconde étape :

```

\text{ETAPE = r1, r2
r3}
\steps{\ETAPE}
\statement{
  \if{\step = 1 }{question 1 : \embed{r1}, question 2 : \embed{r2}}
  \if{\step = 2}{question 3 : \embed{r3}}
}
\answer{}{\a}{type=...}
\answer{}{\b}{type=...}
\answer{}{\c}{type=...}

```


NB :

- `r1` est un mot réservé équivalent à `reply1` ou `reply 1` ou `r 1`. De même pour `r2` et `r3`.
- Dans cet exercice à deux étapes, l'utilisateur voit l'énoncé de la question 3 seulement s'il a répondu correctement aux questions 1 et 2 posées à la première étape.
- Si on veut que l'utilisateur accède à la seconde étape même si il a fait une erreur à la seconde étape, il faut ajouter l'option `nonstop` aux commandes `\answer` de la première étape.

EXEMPLE 4.7. Structure schématique d'un exercice où l'on pose une seconde question si la réponse à la première question est fausse :

```
\text{etap = r1}
\nextstep{etap}

\statement{Enoncé de la question 1.
\if{\step = 1}{ \embed{r1} }
\if{\step >= 2 and \sc_reply1 = 1}{ Votre réponse \reply1 est correcte. }
\if{\step >= 2 and \sc_reply1 = 0}{ Votre réponse \reply1 n'est pas correcte.
Enoncé de la question 2. \embed{r2} } }

\answer{}{\rep1}{type = numeric}{option = nonstop noanalyzeprint}
\answer{}{\rep2}{type = numeric}{option = noanalyzeprint}

\text{etap = }
\if{\step = 2 and \sc_reply1 = 0}{ \text{etap = r2} }
```

Dans cet exemple, l'option `nonstop` est nécessaire pour que la seconde question soit posée lorsque la réponse à la première question est fausse. La variable `sc_reply1` sert dans cet exemple pour définir la seconde étape et pour afficher un message différent à la seconde étape dans l'énoncé de l'exercice selon que la réponse à la première question est juste ou fausse.

Nous allons donner d'autres exemples illustrant différentes utilisations typiques et `\steps` et `\nextstep`.

4.2.2. Avoir un nombre de questions dépendant des données aléatoires de l'exercice

On donne ici un exemple d'exercice où il n'y a en fait qu'une étape mais où on posera deux ou trois questions suivant la valeur d'une variable `n` déclarée avant. Les lignes clés sont les suivantes si `\n` est égal à 1 ou 2 :

```
\text{ETAPE = \n = 2 ? r1,r2 : r1,r2,r3}
\steps{\ETAPE}
\statement{ ---- }
\answer{}{\rep1}{type=xxx}
\answer{}{\rep2}{type=xxx}
\answer{}{\rep3}{type=xxx}
```

Si $n = 2$, la dernière commande `\answer` n'est pas utilisée. On peut remplacer la ligne

```
\text{ ETAPE = \n = 2 ? r1,r2 : r1,r2,r3}
```

par

```
\text{ ETAPE = wims(makelist r x for x = 1 to \n)}
```

EXERCICE 4.2. Ecrire un exercice demandant de remplir un tableau de quatre, cinq ou six cases selon un paramètre aléatoire. Par exemple, demander de calculer un nombre variable de carrés à partir de l'entier `\m`.

Indications : *Pensez à utiliser les commandes `WIMSwims(makelist)` et `wims(values)` qui évitent de faire des boucles.*

4.2.3. Poser les questions les unes après les autres

On désire ici poser une question à chaque étape. La i -ième ligne de notre variable **ETAPE** contient les questions qui doivent être posées à la i -ème étape.

EXEMPLE 4.8. Dans l'exemple schématique suivant, il y a trois étapes et une question est posée à chaque étape :

```
\matrix{ ETAPE = r1
r2
r3}
\steps{\ETAPE}

\statement{ ... }

\answer{}{\rep1}{type=xxx}
\answer{}{\rep2}{type=xxx}
\answer{}{\rep3}{type=xxx}
```

EXEMPLE 4.9. Dans l'exemple schématique suivant, deux questions sont posées à la première étape et une question est posée à la seconde étape :

```
\matrix{ ETAPE = r1,r2
r3}
\steps{\ETAPE}

\statement{ ... }

\answer{}{\rep1}{type=xxx}
\answer{}{\rep2}{type=xxx}
\answer{}{\rep3}{type=xxx}
```

EXEMPLE 4.10. L'exercice suivant est un exercice à deux étapes, à chaque étape une question est posée. On passe à l'étape 2 même si la réponse à l'étape 1 est fautive grâce à l'option **nonstop**. Dans le cas d'une mauvaise réponse à l'étape 1, un message d'avertissement est affiché à l'étape 2.

```
\title{Doubles}
\precision{100000}
\integer{n=randint(50..100)}
\integer{N1=2*\n}
\integer{N2=4*\n}
\text{N=\n,\N1,\N2}
\steps{r1
r2}
\statement{\if{\step=2 and \sc_reply1<1}{
Vous vous êtes trompé : le double de \n est \N1.}
<p> Calculer le double de \N[\step].
<center>\embed{r \step}</center>}
```

```

}
\answer{double de \N[1]}\N[2]}{type=numeric}{option=nonstop}
\answer{double de \N[2]}\N[3]}{type=numeric}{option=nonstop}

```

EXERCICE 4.3. Modifier l'exercice précédent pour que les carrés soient demandés les uns après les autres.

NB : On peut remarquer que si on ne remplit pas le premier champ de `\answer`, l'analyse de la réponse est difficilement lisible dans le cas d'un exercice à étapes.

4.2.4. Permettre de refaire un essai

On désire ici reposer la question si la réponse est fausse. Il faut analyser soi-même la réponse à l'aide de `\condition` et éventuellement de `\conditions`.

EXEMPLE 4.11. Dans l'exemple ci-dessous, on demande le carré d'un entier ; l'utilisateur a le droit à deux essais.

```

\title{ Le carré d'un entier avec 2 essais }
\computeanswer{ no }
\integer{ n = randint(-5..5) }
\integer{ N = (\n)^2 }
\text{COND = 1}
\text{ETAPE = r1}
\nextstep{\ETAPE}
\conditions{1}

\statement{ Calculer le carré de \n.<br>
}
\answer{ Carré de \n }{ \var1 }{ type=numeric }
\answer{ Carré de \n (2ème essai) }{ \N }{ type=numeric }

\condition{Votre première réponse est-elle correcte ?}{\var1 = \N}
\text{ETAPE=}
\if{\step = 2 and \var1!=\N}{\text{ETAPE = r2}}
\feedback{ \reply1 < 0 }{ Le carré d'un entier est toujours positif. }
\feedback{\var1!=\N and \reply2 = \N}{
  Vous avez répondu correctement au 2ème essai.
}
\feedback{\var1!=\N and \reply2!=\N}{Vos deux réponses sont incorrectes.}

```

EXERCICE 4.4. (1) Ecrire un exercice demandant la lettre de l'alphabet suivant une lettre donnée et permettant au plus trois essais.

- (2) Modifier l'exercice pour qu'il affiche le numéro de l'essai.
- (3) Rajouter un `feedback` indiquant le nombre d'essais qui a été nécessaire.
- (4) Ecrire un exercice demandant deux lettres de l'alphabet à partir d'une lettre donnée et permettant au plus trois essais. Les réponses justes ne doivent pas être redemandées.
- (5) Modifier l'exercice pour qu'il affiche le numéro de l'essai dans le texte.

4.2.5. Utiliser les réponses pour décider des étapes suivantes

L'exemple 4.7 montre déjà un premier exemple d'utilisation de `\nextstep` dans ce but. En voici un second qui montre comment utiliser la commande `\conditions` pour une analyse des réponses à l'aide de conditions dans un exercice utilisant `\nextstep`.

EXEMPLE 4.12. Structure schématique d'un exercice où la première étape est formée d'une question et la deuxième étape, d'une question différente suivant la réponse de l'utilisateur à la question 1, la réponse à l'étape 2 sera conservée dans la variable `reply2` ou `reply3` suivant la question posée. On analyse chaque réponse à l'aide de commandes `\condition`; dans un cas, l'analyse de la réponse tiendra compte des deux premières commandes `\condition{}{}` et dans l'autre cas, l'analyse de la réponse tiendra compte de la première et de la troisième commande `\condition{}{}`.

```
\text{ETAPE = r1}
\text{COND = 1}
\nextstep{\ETAPE}
\conditions{\COND}
\statement{
  \if{\step = 1}{énoncé pour la 1ère question}
  \if{\step = 2 and \ETAPE = r2}{version 1 de l'énoncé de la 2ème question}
  \if{\step = 2 and \ETAPE = r3}{version 2 de l'énoncé de la 2ème question}
}
\answer{}{\var1}{type= ...}
\answer{}{\var2}{type= ...}
\answer{}{\var3}{type= ...}
\condition{}{conditions sur \var1}
\condition{}{conditions sur \var2}
\condition{}{conditions sur \var3}

\text{ETAPE=}
\if{ \step = 2 and \var1 = " 1ère valeur possible" }{
  \text{ETAPE = r2}
  \text{COND = 1,2}
}
\if{ \step = 2 and \var1 = "2ème valeur possible" }{
  \text{ETAPE = r3}
  \text{COND = 1,3}
}
```

L'exercice s'arrête lorsque `\ETAPE` est vide : ici, il s'arrête après l'étape 2 si l'un des deux choix possibles pour la réponse à la première réponse est donné par l'utilisateur. Dans le cas contraire, il s'arrête après l'étape 1. Ici `var1`, `var2` et `var3` sont des variables non utilisées avant.

NB : On écrit `\steps{r1,r2,r3}` mais `\conditions{1,2,3}`.

EXERCICE 4.5. Ecrire un exercice où l'on demande le carré d'un entier. Si la réponse est juste, on demande le carré suivant. Si la réponse est fausse, on demande les deux carrés suivants.

4.2.6. Complément

Une des retombées de l'utilisation de `\nextstep` est de pouvoir changer la valeur d'une variable après une requête de l'utilisateur et de la faire apparaître dans l'énoncé avec sa nouvelle valeur. Il est obligatoire que cette variable soit définie une fois avant le statement, quitte à ne rien mettre dedans.

EXEMPLE 4.13.

```
\title{Compter de 3 en 3}
\precision{100000}
\integer{n=randint(50..100)}
\text{N=wims(values \n + 3*k for k = 0 to 5)}
\text{ETAPE=wims(makelist r x for x = 1 to 5)}
\text{W=}
\text{W1=}
\text{nstep=r1}
\nextstep{\nstep}
\statement{Comptez de 3 en 3 à partir de \n.
  <p>\if{\step > 1 and \W[\step-1] < 1}{
    <font color=red> Vous vous êtes trompé, vous avez répondu \W1[\step-1]</font>,
    or c'était \N[\step] }
  <p align="center"><font color=green>\N[1..\step] </font></p>
  \if{\step <= 5}{<p>Donner le suivant de \N[\step] :
    <center>\embed{\nstep}</center>}
  }
\answer{\N[2]}{type=numeric}{option=nonstop}
\answer{\N[3]}{type=numeric}{option=nonstop}
\answer{\N[4]}{type=numeric}{option=nonstop}
\answer{\N[5]}{type=numeric}{option=nonstop}
\answer{\N[6]}{type=numeric}{option=nonstop}

\text{nstep=\step <= 5 ? \ETAPE[\step]:}
\text{W1=\reply1,\reply2,\reply3,\reply4,\reply5}
\text{W=\sc_reply1,\sc_reply2,\sc_reply3,\sc_reply4,\sc_reply5}
```

EXERCICE 4.6. Améliorer l'exercice 3.4 en ajoutant un commentaire lorsque l'utilisateur s'est trompé : on fera apparaître une flèche rouge montrant un organite correspondant à la réponse de l'utilisateur sur le dessin de la cellule qui se trouve dans l'énoncé (l'utilisation de `\nextstep` évite qu'un second dessin s'affiche en dessous de l'analyse de la réponse).

4.2.7. Un exercice pour tout récapituler ! Le compte est bon

EXERCICE 4.7. Ecrire un exercice donnant 3 entiers positifs et un nombre qui peut s'obtenir par addition ou multiplication de ces nombres, chaque chiffre ne pouvant être utilisé qu'une fois :

- Dans une première étape, poser la question :

De combien d'opérations aurez-vous besoin ?

- Si la réponse est différente de 1 et de 2, l'exercice s'arrête et on met un message pour expliquer l'erreur.

- Si la réponse est 1 ou 2, on passe à la deuxième étape en demandant d'entrer la formule avec un nombre de champs de réponse qui sera différent suivant que le nombre d'opérations est 1 ou 2.
- Mettre des conditions pour s'assurer que l'utilisateur a bien respecté les consignes.

ANNEXE A

Tableaux

1.1. Conditions de test

relation	signification : vrai si
<code>=</code> or <code>==</code>	<code>string1</code> et <code>string2</code> sont identiques.
<code>!=</code> or <code><></code>	<code>string1</code> et <code>string2</code> ne sont pas identiques.
<code><</code>	l'évaluation numérique de <code>string1</code> est strictement inférieure à celle de <code>string2</code>
<code><=</code>	l'évaluation numérique de <code>string1</code> est inférieure ou égale à celle de <code>string2</code>
<code>></code>	l'évaluation numérique de <code>string1</code> est strictement supérieure à celle de <code>string2</code>
<code>>=</code>	l'évaluation numérique de <code>string1</code> est supérieure à celle <code>string2</code>
<code>isin</code>	<code>string1</code> est une sous-chaîne de caractères de <code>string2</code>
<code>notin</code>	<code>string1</code> n'est pas une sous-chaîne de caractères de <code>string2</code>
<code>iswordof</code>	<code>string1</code> est un mot de <code>string2</code>
<code>notwordof</code>	<code>string1</code> n'est pas un mot de <code>string2</code>
<code>isvarof</code>	<code>string1</code> est une variable mathématique de l'expression <code>string2</code>
<code>notvarof</code>	<code>string1</code> n'est pas une variable mathématique de l'expression <code>string2</code>
<code>isvariableof</code>	<code>string1</code> est une variable mathématique de l'expression <code>string2</code>
<code>notvariableof</code>	<code>string1</code> n'est pas une variable mathématique de l'expression <code>string2</code>
<code>isitemof</code>	<code>string1</code> est un item de la liste <code>string2</code>
<code>notitemof</code>	<code>string1</code> n'est pas un item de la liste <code>string2</code>
<code>islineof</code>	<code>string1</code> est une ligne de <code>string2</code>
<code>notlineof</code>	<code>string1</code> n'est pas une ligne de <code>string2</code>
<code>issamecase</code>	<code>string1</code> et <code>string2</code> sont les mêmes textes à des espaces multiples près, mais tenant compte de la casse des lettres
<code>notsamecase</code>	vrai si <code>string1</code> et <code>string2</code> ne vérifient pas le critère ci-dessus
<code>issametext</code>	vrai si <code>string1</code> et <code>string2</code> sont les mêmes textes à des espaces multiples près, à la casse près et aux lettres accentuées près
<code>notsametext</code>	vrai si <code>string1</code> et <code>string2</code> ne vérifient pas le critère précédent

NB : Un *mot* est une chaîne de caractères sans espaces, une liste est une suite d'items séparés par des virgules. Un *item* d'une liste est une sous-chaîne de caractères entre deux virgules (ou au début ou la fin).

1.2. Commandes WIMS de base

TAB. 1. Commandes de base pour rendre aléatoire les données d'un exercice

<code>randint(a..b)</code>	retourne un entier choisi au hasard entre a et b
<code>random(a..b)</code>	retourne un réel choisi au hasard entre a et b
<code>random(liste)</code>	retourne un élément de la liste choisi au hasard
<code>shuffle(n)</code>	retourne la liste des entiers de 1 à n dans un ordre choisi au hasard
<code>shuffle(liste)</code>	retourne les éléments de liste dans un ordre choisi au hasard
<code>randomitem(\list)</code>	retourne un des éléments de <code>\list</code> choisi au hasard
<code>randitem(\list)</code>	
<code>randomrow(\mat)</code>	retourne une des lignes du tableau <code>\mat</code> choisie au hasard

TAB. 2. Commandes pour les manipulations de base des listes et tableaux

<code>items(\liste)</code>	nombre d'éléments de la liste <code>\liste</code> (chaque élément est séparé par une virgule)
<code>rows(\M)</code>	nombre de lignes du tableau <code>\M</code>
<code>position(item,\liste)</code>	liste des positions de « item » dans <code>\liste</code>
<code>asis(texte)</code>	pour déclarer une variable contenant une chaîne de caractères qui ne sera pas interprétée par WIMS

TAB. 3. Les commandes de bases pour manipuler les fonctions

<code>evaluate(x^2 + sin(y),x = 3,y = 4)</code>	valeur de la fonction $x^2 + \sin(y)$ au point $(x, y) = (3, 4)$.
<code>solve(x^3 - 3*x + 1,x = 0..1)</code>	la solution de $x^3 - 3x + 1 = 0$ dans $[0, 1]$
<code>simplify(x^5*y^3*x^2/y)</code>	simplifie l'expression et rend x^3*y^2
<code>diff(sin(x)+cos(y),x)</code>	la dérivée de $\sin(x) + \cos(y)$ par rapport à x
<code>int(x^2 + 3*x + 1,x)</code>	une primitive de $x^2 + 3x + 1$
<code>int(t^2 + 3*t, t = 0..1)</code>	valeur de $\int_0^1 (t^2 + 3t)dt$

1.3. L'analyse des réponses

TAB. 4. Liste des différents types pour l'analyse automatique des réponses

Le tableau décrit les types existants dans la version 3.64 pour l'analyse automatique des réponses. Rappelons la syntaxe :

```
\answer{un commentaire}{réponse}{type=nom}{option= }{weight= }
```

Par exemple,

```
\answer{un commentaire}{chat,chien;veste}{type=clickfill}{option=align=left shuffle} {weight=2}
```

La première colonne donne le nom d'un type (qui doit être mis dans le troisième champ de `\answer`), la troisième colonne décrit la syntaxe pour remplir le deuxième champ de `\answer` et donne un exemple. Certains types de réponses ont des options que l'on peut utiliser en ajoutant des champs après le troisième champ décrivant le type. Les syntaxes de ces options sont données dans la quatrième colonne.

Caractéristiques des types

Nom	Description	Réponse	Option
alexp	Expression mathématique, comparaison formelle. <i>Décrit page 20, voir aussi litexp. la liste des variables est facultative (elle permet de supprimer un message d'erreur dans le cas où une variable n'apparaît pas dans l'expression.)</i>	$\text{expr_math, liste_variable}$ $x^4 + y, x, y$	<code>default=xxx</code>
aset	Ensemble fini avec une analyse tolérante dans l'écriture des expressions. <i>Tolère une écriture approximative des formules mathématiques.</i>	liste_expression $3, 4, 2x$	<code>default=xxx</code>
atext	Texte, la comparaison tolère les différences du type singulier/pluriel, ... <i>Décrit page 16, voir aussi case et nocase.</i>	texte du carbone	<code>default=xxx</code>

Caractéristiques des types

Nom	Description	Réponse	Option
case	Texte (mots ou phrases), la comparaison tient compte de la casse. <i>noreaccent=yes Comparaison en tant que texte sans tolérance ; décrit page 16 ; voir aussi atext et nocase.</i>	texte N'oubliez pas les majuscules	default=xxx
checkbox	Une sorte de réponses à choix multiples. <i>Réponse de type choix multiples où tous les bons choix doivent être sélectionnés (décrit page 17) ; voir aussi mark, click, menu et radio.</i>	liste de numéros ; liste de choix 2,3 ; oui, non, peut-être	split sort shuffle eqweight
chembrut	Formule brute en chimie	la formule brute demandée C7H4Cl2O2	
chemclick	Dessin d'une molécule à l'aide d'un applet <i>La comparaison est faite avec un fichier MDMol. Des options dans le champ embed permettent de configurer l'applet. Voir la documentation sur WIMS pour des précisions.</i>	adresse ... à compléter	
chemdraw	Cliquer sur certains atomes ou certaines liaisons d'une molécule affichée <i>Voir la documentation sur WIMS pour des précisions</i>	atomes : 1,2 bonds : 4,1 ; imagedir/fichier.mol atomes : 1,2 ; imagedir/fichier.mol bonds : 4,1 ; data/fichier.mol	

Caractéristiques des types

Nom	Description	Réponse	Option
chemeq	Equation chimique avec analyse d'équilibre. <i>La seconde « ligne » est l'équation proposée à modifier.</i>	<code>equ_solution ; equ_fournie</code> <code>2H2 + O2 -> 2H2O ; H2 + O2 -> H2O</code>	
chset	Ensemble de caractères.		<code>norepeat</code> <code>default=xxx</code>
click	Une sorte de réponses à choix multiples. <i>Ne pas utiliser en même temps qu'un autre type de réponses ; décrit page 17 ; voir aussi mark, menu et radio.</i>	<code>numéro ; liste de choix</code> <code>1 ; Paris,Londres,Amsterdam</code>	<code>split</code> <code>sort</code> <code>eqweight</code>
clickfill	Objets à glisser-déposer qui peuvent être du texte, des images. <i>Réponse construite avec des éléments à sélectionner (décrit page 19), voir aussi dragfill.</i>	<code>une_rep autre_rep ;</code> <code>liste_de_distructeurs</code> <code>pomme poire ;aubergine</code>	<code>split</code> <code>sort</code> <code>keeporder</code> <code>shuffle</code> <code>align=left</code> <code>align=right</code> <code>align=center</code> <code>noorder</code>
clicktile	Mettre des pavés rectangulaires d'une certaine couleur <i>Chaque carré est repéré par son coin en haut à gauche sous la forme entier :entier. Nécessite l'utilisation de <code>\embed</code>.</i>	<code>color1,x1 :y1,x2 :y3;color2,x3 :y3</code> <code>blue,1 :0,1 :-1,1 :-2 ; green,2 :-2,2 :-3</code>	

Caractéristiques des types

Nom	Description	Réponse	Option
code	sert pour analyser des programmes <i>Non utilisable sur la plupart des serveurs WIMS.</i>	Voir l'aide de Createxo .	
compose	Réponse à construire en utilisant des éléments fournis. <i>Les listes peuvent être formées de petites images ou de formules $\backslash (\)$. Ne pas utiliser d'apostrophes dans une formule mathématique (utiliser $\backslash \sim \{prime\}$)</i>	<pre>xxx,yyy,zzz aaa,bbb,ccc ; liste_de_distracteurs</pre> <hr/> <pre>le,chat,est,noir;noire,et</pre>	default=xxx
coord	Clic sur une image dans une zone prédéfinie.	Voir le tableau spécial.	
correspond	Correspondance entre deux listes d'objets.	<pre>liste_objets_à_droite ; liste_objets_à_gauche</pre> <hr/> <pre>blanc,noir,bleu ; white,black,blue</pre>	split
dragfill	Objets à glisser-déposer qui peuvent être du texte, des images. <i>Réponse construite avec des éléments à sélectionner au plus une fois ; décrit page 19 ; voir aussi clickfill.</i>	<pre>réponse autre_réponse;distracteurs</pre> <hr/> <pre>\(x)\ (2*x)\ ; \((x^2)\, \((x^3)</pre>	<pre>align=left align=right align=center shuffle sort noorder</pre>

Caractéristiques des types

Nom	Description	Réponse	Option
equation	Equation numérique, la comparaison se fait numériquement. <i>Evaluation numérique, la liste de noms de variable acceptées est facultative.</i>	<code>expression1 = expression2,</code> <code>liste_nom_de_variable</code> <code>x + 3y = 1, x, y, z</code>	<code>default=xxx</code> <code>eqsign=yes</code>
flashcard	Une sorte de réponses à choix multiples. <i>Réponse de type choix multiples où tous les bons choix doivent être sélectionnés (décrit page 17); voir aussi mark, click, menu et radio. L'utilisation avec <code>\embed</code> permet d'améliorer la présentation.</i>	<code>liste de numéros; liste de choix; prompt</code> <code>1,2;a,b,c;face cachée</code>	<code>split</code> <code>sort</code> <code>shuffle</code> <code>eqweight</code> <code>show</code>
formal	Expression mathématique, comparaison formelle.	<code>expression_algébrique</code>	<code>default=xxx</code>
fset	Ensemble fini avec une analyse formelle. <i>Expressions formelles avec quelques simplifications (rationnelles ou trigonométriques).</i>	<code>liste_fonction</code> <code>x,x^2,sin(x)</code>	<code>default=xxx</code>
function	Fonction numérique, la comparaison se fait numériquement. <i>La liste de variables acceptées est facultative; décrit page 20.</i>	<code>fonction, liste_nom_de_variable</code> <code>sin(x) + ln(x), x,y</code>	<code>default=xxx</code>

Caractéristiques des types

Nom	Description	Réponse	Option
javacurve	Dessin utilisant Java permettant des tracés avec des formes prédéfinies.	Voir le tableau spécial	
jmolclick	Demande de sélectionner des atomes d'une molécule en 3D <i>Doit être utiliser avec la commande \embed</i>	atomes à sélectionner ; adresse du fichier descriptif de la molécule 1,2 ;data/xxx_N ;imagedir/xxx	
keyboard	Texte tapé à partir du clavier proposé <i>L'option keyboard permet de choisir son clavier, par défaut, clavier phonétique (ou les lettres accentuées des langages parmi de en es fr el it pl ru si en_ipa)</i>	texte écrit avec des symboles html pour les caractères qui ne sont pas isolatin1	
litexp	Expression mathématique, comparaison formelle.	2 + 3	
mark	Une sorte de réponses à choix multiples. <i>Décrit page 17, voir aussi radio, click, menu.</i>	liste de numéros ; liste de mots marqués 2,3 ;choix1,choix2,choix3,choix4	split eqweight
matrix	Matrice numérique, la comparaison se fait numériquement coefficient par coefficient. <i>Matrice numérique (au sens mathématique) ; décrit page 14.</i>	a,b,c ; d,e,f 0,1,0 ;pi,4,5.2 ;0,1,0	default=xxx

Caractéristiques des types

Nom	Description	Réponse	Option
menu	Une sorte de réponses à choix multiples. <i>L'option multiple=n permet un choix multiple avec une fenêtre de n lignes ; décrit page 17 ; voir aussi mark, click et radio.</i>	liste de numéros ; liste 2 ; Paris, Londres, Amsterdam	split shuffle multiple sort eqweight
nocase	Texte (mots ou phrases), la comparaison ne tient pas compte de la casse. <i>Décrit page 16 ; voir aussi atext et case.</i>	texte	default=xxx
numeric	Nombre, la comparaison se fait numériquement avec une précision fixée.	nombre 1.55	default=xxx comma
numexp	Nombre, la comparaison se fait de façon formelle. <i>Traite les nombres rationnels, décrit page 13</i>	nombre 4/8	default=xxx noreduction
radio	Une sorte de réponses à choix multiples.	liste de numéros ; liste de choix	sort shuffle split
range	Nombre dans une zone : tout ce qui est dans la zone est accepté. <i>Décrit page 14.</i>	rep_min, rep_max, rep_affichée 3.67, 3.72, 3.7	default=xxx

Caractéristiques des types

Nom	Description	Réponse	Option
raw	Texte brut, la comparaison se fait par des options. <i>Décrit page 17.</i>	texte brut Cl^-	Voir le tableau spécial
reorder	Mise en ordre d'une liste d'objets. <i>Décrit page 6 ; voir aussi compose. Ne pas utiliser d'apostrophes dans une formule mathématique (utiliser \^f{prime}).</i>	liste ; prompt qui se mettra entre les mots Mercure, Vénus, Terre, Mars, Jupiter, Saturne, Uranus, Neptune ; --	
set	Ensemble fini avec une analyse textuelle. <i>Evaluation de chaînes de caractères, pas de score partiel possible.</i>	liste_non_ordonnée	default=xxx repeat
sigunits	Nombre avec possibilité de demander une unité et un nombre de chiffres significatifs. <i>Ce qui est entre crochets est facultatif, voir aussi units. Il est recommandé de préparer la bonne réponse avec le slib text/sigunits</i>	nombre [unité physique] [#nombre_chiffres_significatifs] [,unité_physique_affiché] 10 m#2,cm	
syntext	Outil avancé d'identification de textes.	Voir la documentation de l'aide de Createxo	
units	Nombre avec unité pour la physique. <i>Décrit page 14, voir aussi sigunits.</i>	nombre unité_SI 10 m	

Caractéristiques des types

Nom	Description	Réponse	Option
vector	Comparaison numérique des coordonnées. <i>Comparaison formelle (ordonnée), décrit page 14.</i>	liste_de_nombres 1, 4, 25, 36	default=xxx
wlist	Liste de mots.		default=xxx

NB : L'option `default=xxx` permet qu'une réponse vide soit acceptée. A la place de `xxx`, on met une valeur qui sera prise par défaut.

EXEMPLE A.1. Avec le code suivant, la bonne réponse est 1. Si l'utilisateur clique sur « Envoyer la réponse » sans avoir entré de réponse, sa réponse sera assimilée à 0 et donc il aura une note de 0/10.

```
\statement{\embed{r1}}
\answer{}{1}{option=default=0}
\feedback{1 = 1}{\reply1}
```

NB : `noanalyzeprint` Cette option permet de ne pas afficher le texte automatique d'analyse de réponses. C'est donc à l'auteur de l'exercice de fournir un feedback convenable. Cela est utile dans certains types comme le type **mark** où le texte automatique est souvent inadéquat. Attention, l'analyse des conditions n'est pas non plus affichée. Seul le score subsiste.

L'exemple précédent utilise la variable `sc_reply1` que l'on a déjà utilisé avec l'option `nonstop`.

Pour certains types de réponses, des lignes supplémentaires au deuxième argument peuvent être rajoutées. Par exemple, dans le cas de réponses de type texte libre, il est possible ainsi de personnaliser le style du champ de réponses. Ainsi, pour `alexp`, `aset`, `case`, `chset`, `default`, `equation`, `fset`, `keyboard`, `nocase`, `numeric`, `numexp`, `range`, `raw`, `set`, `vector`, `wlist`

```
\embed{reply 1,12
style="font-size:18px;background-color:#e8ffff;border:1px solid #3333CC;"
autocomplete=off }
```

Si la première ligne supplémentaire est le mot `default`, le champ aura la classe de style css `wims_oef_input` : le gestionnaire du site ou l'enseignant peuvent alors le configurer de manière générale.

TAB. 6. Le deuxième argument de `\embed`

Type			par défaut	explication
<code>numeric default</code> <code>algexp set aset</code> <code>fset chemeq</code> <code>atext</code>	n	$1 \leq n \leq 100$	20, 20, 40, 30, 30, 30, 50, 40	insérer la première réponse libre avec la taille du champ de réponse égale à n .
<code>checkbox</code> , <code>click</code> , <code>radio</code>	n		tous les choix	insérer seulement le choix numéro n
<code>matrix</code>	$L \times C$	$1 \leq L \leq 15$ $1 \leq C \leq 100$	5 x 25	hauteur et largeur du champ de réponse
<code>clickfill</code> <code>dragfill</code>	$H \times V \times n$	$5 \leq V \leq 500$ $5 \leq H \leq 1000$ $1 \leq n \leq 100$	20 x 80	H : taille horizontale (en pixels) d'une case; V : taille verticale (en pixels) d'une case; n : nombre de cases que contient le champ de réponses. Un quatrième paramètre est possible pour augmenter l'espace vertical en dessous du champ de réponses.
<code>mark</code> <code>flashcard</code>	n n <code>style="..."</code> <code>style="..."</code>	$1 \leq n \leq 1000$	tous les choix et un style par défaut	nombre de mots proposés au marquage insérer seulement le choix numéro n la deuxième ligne est le style de la carte face cachée, la troisième ligne le style css de la carte face visible.
<code>reorder</code> <code>correspond</code>	$10 * H$ $V \times HG \times HD$	$1 \leq H \leq 40$ $10 \leq V \leq 300$ $10 \leq HG \leq 500$ $10 \leq HD \leq 500$	10 40 x 200 x HG	hauteur du champ de réponses V : taille verticale des items, HG et HD : tailles horizontales des colonnes gauche et droite.
<code>clicktile</code>			<code>xrange</code> 0,10 <code>yrange</code> 0,1 <code>colors</code> <code>red,blue</code> <code>back-</code> <code>ground_color</code> <code>green</code>	La première ligne de la commande <code>\embed</code> est formée de la taille en pixels de l'applet X x Y . La seconde ligne entre [et de] est formée des commandes précisant les pavés colorés (ils ne pourront pas être modifiés). Par exemple [<code>xrange -5,5</code> <code>yrange -5,5</code> <code>background_color yellow</code> <code>square blue,1:1,1:2,1:3</code> <code>square green,2:3,2:4</code> <code>point red,0:0</code>]

TAB. 8. Les possibilités du type **coord**

<code>point,x,y</code>	Point en (x,y) - C'est un point « épais », de largeur fixe.
<code>rectangle,x1,y1,x2,y2</code>	l'intérieur d'un rectangle de diagonale (x1,y1) —(x2,y2).
<code>circle,x,y,d</code>	l'intérieur d'un cercle de centre (x,y) et diamètre d.
<code>ellipse,x,y,w,h</code>	l'intérieur d'une ellipse de centre (x,y), largeur w, et hauteur h.
<code>polygon,x1,y1,x2,y2,x3,y3,...</code>	l'intérieur d'un polygone engendré par les points (x1,y1), (x2,y2), (x3,y3), ...
<code>bound,NOMFIC,x,y</code>	zone définie dans le fichier image NOMFIC, qui doit être de la même taille que l'image cliquée (mais peut être une image différente). La condition est remplie si (x,y) est dans la même zone de remplissage que le clic de l'utilisateur.
<code>bound,NOMFIC</code>	comme ci-dessus, mais la condition est remplie si le clic de l'utilisateur a une couleur différente du pixel du coin en haut à gauche de NOMFIC.

TAB. 9. Les possibilités du type **javacurve**

<code>points,x1,y1,x2,y2,...</code>	Points en (x1,y1), (x2,y2), C'est un point « épais », de largeur fixe.
<code>line,x1,y1,x2,y2</code>	Droite passant par (x1,y1), (x2,y2).
<code>sline,x1,y1,x2,y2</code>	Demi-droite passant par (x1,y1), (x2,y2), d'origine (x1,y1).
<code>polygon,x1,y1,x2,y2,...</code>	Polygone de sommets (x1,y1), (x2,y2),
<code>segment,x1,y1,x2,y2</code>	Segment d'extrémités (x1,y1), (x2,y2).
<code>vector,x1,y1,x2,y2</code>	Vecteur de (x1,y1) vers (x2,y2).
<code>rectangle,x1,y1,x2,y2</code>	Rectangle de diagonale (x1,y1), (x2,y2) .
<code>rectangle,x1,y1,x2,y2,x3,y3,x4,y4</code>	Rectangle contenu dans la différence du rectangle de diagonale (x1,y1), (x2,y2) et du rectangle de diagonale (x3,y3), (x4,y4).
<code>circle,x1,y1,r</code>	Cercle de centre (x1,y1) et de rayon r.

TAB. 10. Les options du type **raw**

noaccent	enlève les accents sur les lettres.
nocase	transforme les lettres en minuscule.
nodigit	remplace les chiffres par des espaces.
nomathop	remplace les opérateurs mathématiques par des espaces
noparenthesis	remplace les parenthèses par des espaces.
nopunct	remplace les ponctuations par des espaces.
noquote	remplace les apostrophes ou guillemets (simple et double) par des espaces.
nospace	enlève tous les caractères d'espace (y compris ceux provenant de remplacement d'autres caractères).
reaccent	permet les lettres accentuées précédées de \.
singlespace	traite toutes les chaînes de caractères d'espaces comme un seul espace.

1.4. Les commandes de dessin FLYDRAW

Les tableaux suivants décrivent une partie des commandes FLYDRAW pour l'élaboration d'une image (voir 3.2.1 pour des exemples d'utilisation). L'indication [color] doit être remplacé (sans les crochets) par le nom html d'une couleur (blue, red, green,...) ou par le code RGB (i.e. 3 entiers entre 0 et 255, séparés par des virgules, correspondant aux valeurs de rouge, vert ,bleu).

TAB. 11. Tracés de points, de droites et de flèches

pixels [color],x1,y1,x2,y2,...	Points de diamètre 1 aux coordonnées (x1 ; y1), (x2 ; y2), ...
setpixel x,y,[color]	Point de coordonnées (x ; y) et de diamètre 1.
point x,y,[color]	Point de coordonnées (x ; y) et de diamètre l'épaisseur de trait.
points [color],x1,y1,x2,y2,...	Points de coordonnées (x1 ; y1), (x2 ; y2), ... et de diamètre l'épaisseur de trait.
line x1,y1,x2,y2,[color]	Segment entre les points de coordonnées (x1 ; y1) et (x2 ; y2).
lines [color],x1,y1,x2,y2,x3,y3...	Ligne polygonale joignant les points (x1 ; y1), (x2 ; y2), (x3 ; y3) ...
rays [color],x0,y0,x1,y1,x2,y2...	Segments joignant (x0 ; y0) et (x1 ; y1), (x0 ; y0) et (x2 ; y2), ...
parallel x1,y1,x2,y2,xv,yv,n,[color]	n segments parallèles partant du segment d'extrémités (x1 ; y1) et (x2 ; y2) avec le déplacement de vecteur (xv ; yv).
hline x,y,[color]	Droite horizontale passant par le point (x ; y).
vline x,y,[color]	Droite verticale passant par le point (x ; y).
dline x1,y1,x2,y2,[color]	Segment en pointillés entre les points de coordonnées (x1 ; y1) et (x2 ; y2).
dlines [color],x1,y1,x2,y2,x3,y3...	Ligne polygonale en pointillés joignant les points (x1 ; y1), (x2 ; y2), (x3 ; y3) ...
dhline x,y,[color]	Droite horizontale en pointillés passant par le point (x ; y).
dvline x,y,[color]	Droite verticale en pointillés passant par le point (x ; y).
arrow x1,y1,x2,y2,l,[color]	Flèche allant du point (x1 ; y1) vers le point (x2 ; y2) et dont la tête est de longueur l pixels.
arrow2 x1,y1,x2,y2,l,[color]	Flèche entre les points (x1 ; y1) et (x2 ; y2) à deux têtes de longueur l pixels.
darrow x1,y1,x2,y2,l,[color]	Flèche en pointillés allant du point (x1 ; y1) vers le point (x2 ; y2) dont la tête est de longueur l pixels.
darrow2 x1,y1,x2,y2,l,[color]	Flèche en pointillés entre les points (x1 ; y1) et (x2 ; y2) et à deux têtes de longueur l pixels.

TAB. 12. Tracés d'arcs, ellipse, polygones

arc x,y,w,h,a1,a2,[color]	Arc de l'ellipse de largeur w et de hauteur h centrée en (x,y) entre l'angle a1 et l'angle a2 en degrés.
circle x,y,d,[color]	Cercle de centre (x ; y) et de diamètre d pixels.
ellipse x,y,w,h,[color]	Ellipse de largeur w et de hauteur h centrée en (x;y).
polygon [color],x1,y1,x2,y2,x3,y3...	Polygone de sommets (x1 ; y1), (x2 ; y2), (x3 ; y3)...
rect x1,y1,x2,y2,[color]	Rectangle de diagonale (x1 ; y1) et (x2 ; y2).
square x,y,s,[color]	Carré de coin supérieur gauche (x ; y) et de côté r.
triangle x1,y1,x2,y2,x3,y3,[color]	Triangle de sommet (x1 ; y1), (x2 ; y2), (x3 ; y3).

TAB. 13. Figures pleines et coloriage de régions

fcircle x,y,d,[color]	Disque de centre (x ; y) et de diamètre d pixels.
fellipse x,y,w,h,[color]	Ellipse de largeur w et de hauteur h centrée en (x,y) et remplie avec la couleur color.
fpoly [color],x1,y1,x2,y2,x3,y3...	Polygone de sommets (x1 ; y1), (x2 ; y2), (x3 ; y3) ... et rempli avec la couleur color.
frect x1,y1,x2,y2,[color]	Rectangle de diagonale (x1 ; y1) et (x2 ; y2) et rempli avec la couleur color.
fsquare x,y,s,[color]	Carré de coin supérieur gauche (x ; y) et de côté de longueur s et rempli avec la couleur color.
ftriangle x1,y1,x2,y2,x3,y3,[color]	Triangle de sommets (x1 ; y1), (x2 ; y2), (x3 ; y3) et rempli avec la couleur color.
diamondfill x,y,nx,ny,[color]	Remplit la région contenant le point (x ; y) avec des lignes de couleur color (quadrillage oblique). (nx ; ny) est la distance verticale et horizontale entre deux lignes.
dotfill x,y,nx,ny,[color]	Remplit la région contenant le point (x ; y) avec des gros points de couleur color. (nx ; ny) est la distance verticale et horizontale entre deux points.
fill x,y,[color]	Colorie la région contenant le point (x ; y).
filltoborder x,y,[color1],[color2]	Colorie avec la couleur color2 la région contenant (x ; y) et délimitée par la couleur color1.
gridfill x,y,nx,ny,[color]	Remplit la région contenant le point (x ; y) avec des lignes de couleur color (quadrillage droit) . (nx ; ny) est la distance verticale et horizontale entre deux lignes.
hatchfill x,y,nx,ny,[color]	Remplit la région contenant le point (x ; y) avec des lignes (simples) de couleur color. (nx ; ny) est la distance verticale et horizontale entre deux lignes.

TAB. 14. Paramètres d'une figure

range x1,x2,y1,y2	Détermine les coordonnées des bords de l'image.
xrange x1,x2	Détermine les coordonnées horizontales des bords de l'image.
yrange y1,y2	Détermine les coordonnées verticales des bords de l'image.
linewidth w	Epaisseur des traits à w pixels.
trange t1,t2	Intervalle du paramètre pour le tracé des courbes paramétriques (par défaut 0 et 1).
transparent [color]	Définit la couleur color comme transparente.

TAB. 15. Texte

text [color],x,y,[font],[string]	Ecrit string au point de coordonnées (x ; y) avec la police font=small,medium,large ou giant.
textup [color],x,y,[font],[string]	Ecrit string en majuscules au point de coordonnées (x ; y) avec la police font=small,medium,large ou giant.
comment	ligne de commentaire non interprétée par FLYDRAW.

TAB. 16. Tracés de fonctions, surfaces

plot [color],[formula]	Courbe représentative de la fonction formula.
plotjump j	Saut de la courbe tracée si deux points consécutifs ont une distance de plus de j pixels. Utile afin d'éviter de dessiner des fonctions discontinues comme des fonctions continues (Valeur par défaut : 200).
plotsteps n	Nombre de points calculés dans le tracé de courbes (Valeur par défaut : 100).
levelcurve [color],[expression],l1,l2,...	Dessine des courbes de niveau de la surface décrite par expression de niveaux l1, l2,...
levelstep n	Règle le nombre d'étapes en pixels utilisé pour le dessin des courbes de niveaux. Entre 1 and 16, défaut : 4.

TAB. 17. Insertion d'une image dans un dessin et transformations d'un dessin (voir l'aide en ligne de **Createxo** pour toutes les commandes)

copy x,y,x1,y1,x2,y2,[filename]	Insère le rectangle de diagonale (x1 ; y1) et (x2 ; y2) du fichier filename au point (x ; y). si x1 = y1 = x2 = y2 = -1, tout le fichier est copié.
size x,y	Redimensionne l'image pour qu'elle soit de taille x pixels horizontalement et y pixels verticalement.
affine a,b,c,d,tx,ty	Transformation affine $(x;y) \rightarrow [a,b;c,d](x;y)+(tx;ty)$ des objets définis par les lignes de commande qui suivent (jusqu'à la ligne de commande killaffine).
rotation d	Rotation de d degrés dans le sens inverse des aiguilles d'une montre, de centre (0;0) des objets définis par les lignes de commande qui suivent (jusqu'à la ligne de commande killrotation)

1.5. Divers

TAB. 18. Code html de quelques caractères

(;	!	! ;	"	" ;	#	# ;	\$	$;	%	% ;	&	& ;	,	' ;
)	(;)) ;	*	* ;	+	+ ;	,	, ;	-	- ;	.	. ;	/	/ ;
:	: ;	;	; ;	<	< ;	=	= ;	>	> ;	?	? ;	@	@ ;	[[;
\	\ ;]] ;	^	^ ;	_	_ ;	{	{ ;	}	} ;				
dollar $; euro € ; livre Sterling £ ; yen ¥ ;															
α	α ;	β	β ;	γ	γ ;	δ	δ ;	ϵ	ε ;	ζ	ζ ;	η	η ;	θ	θ ;
κ	κ ;	λ	λ ;	μ	μ ;	ν	ν ;	ξ	ξ ;	π	π ;	ρ	ρ ;	σ	σ ;
τ	τ ;	ϕ	φ ;	χ	χ ;	ψ	ψ ;	ω	ω ;						

ANNEXE B

Les solutions

Exercice 1.1

```
\title{Un pré}
\language{fr}
\author{Sophie Lemaire}
\email{sophie.lemaire@math.u-psud.fr}
\computeanswer{no}
\precision{10000}

\integer{L = 10*randint(1..10)}
\integer{l = 10*randint(1..10)}
\integer{per = 2*(\L+\l)}

\statement{Donner le périmètre d'un pré rectangulaire
de longueur \L m et de largeur \l m}

\answer{périmètre (en m)}{\per}{type=numeric}
```

Exercice 1.3

```
\title{Un pré (units)}
\computeanswer{no}
\precision{10000}

\integer{L = 10*randint(1..10)}
\integer{l = 10*randint(1..10)}
\integer{per = 2*(\L+\l)}
\integer{super = \L*\l}

\statement{Donner le périmètre et la superficie d'un pré rectangulaire
de longueur \L m et de largeur \l m}

\answer{périmètre }{\per m}{type = units}
\answer{superficie}{\super m^2}{type = units}
```

Exercice 1.4

```
\title{Un pré bis}
\language{fr}
\computeanswer{no}
```

```

\precision{10000}

\integer{L = 10*randint(1..10)}
\integer{l = 10*randint(1..10)}
\integer{per = 2*(\L+\l)}
\integer{super = \L*\l}

\statement{Un pré rectangulaire de longueur \L m et de largeur \l m
a une bordure de \embed{reply1,8} et une superficie de \embed{reply2,8}.
}

\answer{périmètre}{\per m}{type=units}
\answer{superficie}{\super m^2}{type=units}

```

Exercice 1.5

```

\title{Disque (range)}
\language{fr}
\computeanswer{no}
\format{html}
\precision{10000}

\real{r = randint(100..500)/100}
\real{c = 2*pi*\r}
\real{valc = round(100*\c)/100}

\statement{Déterminer la circonférence d'un disque de rayon \r.}

\answer{Circonférence}{\c - 0.01,\c + 0.01,\valc}{type=range}

```

Exercice 1.6

```

\title{Unités du système SI (correspond)}
\matrix{liste = Longueur,mètre
Masse,kilogramme
Temps,seconde
Intensité de courant électrique, ampère
Température, kelvin
Intensité lumineuse, candela
Quantité de matière, mole
Angle plan, radian
Angle solide, stéradian}
\text{mix = shuffle(rows(\liste))}
\matrix{question = \liste[\mix[1..4];]}

\statement{Mettre en correspondance les unités du système international :
<center>\embed{reply1}</center>
}

\answer{}{\question[;1];\question[;2]}{type=correspond}{option=split}

```

Exercice 1.7

```

\title{Unités du système SI (nocase)}
\matrix{liste = une longueur,le mètre
  une masse, kilogramme,le
  une quantité de matière,mole,la,une
  un temps, seconde,un
  une intensité de courant électrique,ampère,l',un
  une température,kelvin,le, un
  une intensité lumineuse,candela,le,un
  un angle plan,radian,le,un
  un angle solide,stéradian,le,un}
\matrix{question = randomrow(\liste)}
\integer{rep = position(\question[;2],\liste[;2])}
\text{\lsol = \liste[\rep;3] \liste[\rep;2] | \liste[\rep;2]
  | \liste[\rep;4] \liste[\rep;2]}

\statement{Quelle est l'unité pour \question[;1] ?
<center>\embed{reply1}</center></p>
}

\answer{}{\lsol}{type=nocase}

```

Exercice 1.8

```

\title{Unités du système SI (click)}
\matrix{liste = une longueur, le mètre
  une masse, le kilogramme
  une quantité de matière, la mole
  un temps, la seconde
  une intensité de courant électrique, l'ampère
  une température, le kelvin
  une intensité lumineuse, le candela
  un angle plan, le radian
  un angle solide, le stéradian}
\matrix{question = randomrow(\liste)}
\integer{rep = position(\question[;2],\liste[;2])}

\statement{
  L'unité pour \question[;1] est :
  <center>\embed{reply1}</center>
}
\answer{}{\rep;\liste[;2]}{type=click}

```

Exercice 1.8

```

\title{Unités du système SI (radio)}
\matrix{liste = une longueur,le mètre
  une masse,le kilogramme
  une quantité de matière, la mole

```

```

un temps, la seconde
une intensité de courant électrique, l'ampère
une température, le kelvin
une intensité lumineuse, le candela
un angle plan, le radian
un angle solide, le stéradian}
\matrix{question = randomrow(\liste)}
\integer{rep = position(\question[;2],\liste[;2])}

\statement{Sélectionner l'unité pour \question[;1] :
  <center>\embed{reply1}</center>
}
\answer{}{\rep;\liste[;2]}{type=radio}

\integer{pos = position(\reply1,\liste[;2])}
\feedback{\rep != \pos}{
  <font color=red>\reply1 est l'unité pour \liste[\pos;1].</font>
}

```

Exercice 1.11

```

\title{Tangente (function)}
\range{-5,5}
\real{a = random(-1,1)*randint(1..30)/10}
\real{b = randint(-20..20)/10}
\real{c = randint(-20..20)/10}
\function{f = \a*x^2+simplify(\b*x)+\c}
\real{x0 = randint(-40..40)/10}
\real{y0 = evaluate(\f,x = \x0)}
\function{df = diff(\f,x)}
\real{df0 = evaluate(\df,x = \x0)}
\function{D = \df0*x-simplify(\df0*\x0-\y0)}

\statement{ Donner l'équation de la tangente à la courbe d'équation
\ (y = \f\ ) au point (\x0, \y0).
  <p align="center"> \ ( y = \ ) \embed{reply1} </p>
}

\answer{y=}{\D,x}{type = function}

```

Exercice 1.12

```

\title{Disque 2}
\language{fr}
\computeanswer{no}
\precision{1000}
\real{r = randint(100..500)/100}
\real{c = 2*pi*\r}

\statement{Déterminer la circonférence d'un disque de rayon \r.}

```

```

\answer{Circonférence}{\var}{type=numeric}
\feedback{1 = 1}{\reply1}
\real{ec1 = (1 + 0.001)*\c}
\real{ec2 = (1 - 0.001)*\c}
\condition{Bonne précision}{\var <= \ec1 and \var >= \ec2 }{option=hide}
\feedback{\var < \ec2}{Vous avez donné la circonférence d'un disque dont
le rayon est plus petit que \r. }
\feedback{\var > \ec1}{Vous avez donné la circonférence d'un disque dont
le rayon est plus grand que \r. }

```

Exercice 2.1

```

\title{Unités de grandeurs physiques II}
\matrix{liste=
fréquence,hertz,Hz,s\^{(-1)}\}
force,newton,N, kg,m, s\^{(-2)}\}
pression,pascal,Pa,kg,m\^{(-1)}\},s\^{(-2)}\}
travail,joule,J,kg, m\^{(2)}\},s\^{(-2)}\}
puissance,watt,W, kg,m\^{(2)}\},s\^{(-3)}\}\}
\text{quest = randomrow(liste)}
\text{enonce = \quest[2]}
\text{symbol = kg,m,s,kg\^{(2)}\},m\^{(2)}\},s\^{(2)}\},kg\^{(3)}\},m\^{(3)}\},
s\^{(3)}\},kg\^{(-1)}\},m\^{(-1)}\},s\^{(-1)}\},kg\^{(-2)}\},m\^{(-2)}\},
s\^{(-2)}\},kg\^{(-3)}\},m\^{(-3)}\},s\^{(-3)}\}\}
\statement{Le <b>\enonce</b> peut s'exprimer à l'aide
des trois unités suivantes <b>kg, m, s</b>.<p>
Donner son expression sous la forme <b>kg\^{(a)}\}.m\^{(b)}\}.s\^{(c)}\}</b>
(attention à bien respecter l'ordre des symboles)
</p>
<center>\embed{reply1,50 x 50 x 3}</center>
\answer{}{\var;\symbol}{type=dragfill}{option=sort}
\text{rep = wims(sort list \var)}
\condition{Votre réponse est-elle juste ? }{\quest[4..-1] issametext \rep}

```

Exercice 2.2

```

\title{Divisibilité à trous}
\author{Bernadette Perrin-Riou}
\email{bpr@math.u-psud.fr}
\integer{m = 9*11}
\text{N = \m * randint(300..600)}
\text{cnt = wims(charcnt \N)}
\text{u = shuffle(cnt)}
\text{u = \u[1]}
\text{sol = wims(char \u of \N)}
\text{N = wims(replace char number \u[1] by x\ in \N)}

\statement{ Déterminer le chiffre \ (x\ ) tel que \N soit un multiple de \m.}

```

```
\answer{\(x)}{\sol}
```

Exercice ??

```
\title{Distance de réaction (sigunits)}
\precision{10000}
\integer{vitesse=randint(2..30)*5}
\real{temps=randint(1..10)/2}
\real{distance=(\vitesse/3600)*\temps}
\text{sign=randint(2..4)}
\text{rep=slib(text/sigunits \distance km,\sign,,m)}

\statement{Une voiture se déplace à la vitesse de \vitesse km par heure.
  Quelle distance
  parcourt-elle en \temps seconde(s)
  (donner la réponse avec \sign
  chiffres significatifs, sans oublier l'unité) ? \embed{reply1,10}
}
\answer{Distance}{\rep}{type=sigunits}
```

Exercice 2.5

```
\title{Propriétés d'un test sanguin}
\integer{n = randint(10..100)}
\integer{x = randint(5..(\n - 1))}
\integer{y = \n - x}
\integer{xp = randint(1..x)}
\integer{ym = randint(1..y)}
\integer{xm = x - xp}
\integer{yp = y - ym}

\statement{On a fait un test sur \n échantillons de sang dont \x contenaient
une substance interdite X. Le test s'est révélé positif pour \xp échantillon(s)
contenant la substance X et le test s'est révélé négatif pour \ym échantillon(s)
ne contenant pas la substance X. Compléter le tableau des effectifs :


|                          |                           |
|--------------------------|---------------------------|
| <td></td><th>Avec X</th> | <th>Sans X</th>           |
| <th>Test positif</th>    | <td>\embed{reply1,5}</td> |
| <th>Test négatif</th>    | <td>\embed{reply3,5}</td> |


```



```

</table>
}
\answer{}\{xp\}{type=numexp}
\answer{}\{yp\}{type=numexp}
\answer{}\{xm\}{type=numexp}
\answer{}\{ym\}{type=numexp}

```

Exercice 2.7

```

\title{Propriétés d'un test sanguin 2}
\integer{n = randint(10..100)}
\integer{x = randint(5..(\n-1))}
\integer{y = \n-\x}
\integer{xp = randint(1..\x)}
\integer{ym = randint(1..\y)}
\integer{xm = \x-\xp}
\integer{yp = \y-\ym}

\statement{On a fait un test sur \n échantillons de sang dont \x contenaient
une substance interdite X. Le test s'est révélé positif pour
\xp \if{\xp = 1}{échantillon}{échantillons}
contenant la substance X et le test s'est révélé négatif pour \ym
\if{\ym = 1}{échantillon}{échantillons}
ne contenant pas la substance X. Compléter le tableau des effectifs :<p>
<center> <table align="center" border=1 width="30%" bgcolor="antiquewhite">
  <tr align="center">
    <td></td><th>Avec X</th>
    <th>Sans X</th>
  </tr>
  <tr align="center">
    <th>Test positif</th>
    <td>\embed{reply1,5}</td>
    <td>\embed{reply2,5}</td>
  </tr>
  <tr align="center">
    <th>Test négatif</th>
    <td>\embed{reply3,5}</td>
    <td>\embed{reply4,5}</td>
  </tr>
</table></center></p>
\answer{}\{xp\}{type=numexp}
\answer{}\{yp\}{type=numexp}
\answer{}\{xm\}{type=numexp}
\answer{}\{ym\}{type=numexp}

```

Exercice 2.8

```

\title{Maladies infectieuses}
\language{fr}
\format{html}

```

```

\matrix{liste = Tuberculose,bactéries
Tétanos,bactéries
Typhoïde,bactéries
Lèpre,bactéries
Rage,virus
Poliomyélite,virus
Rougeole,virus
Hépatite,virus
Grippe,virus
Bronchiolite,virus
Paludisme,parasites
Toxoplasmose,parasites}
\text{nom = random(bactéries,virus,parasites)}
\text{mix = shuffle(rows(\liste))}
\text{listchoix = \liste[\mix[1..6];]}
\text{choix = \listchoix[;1],ces maladies ne sont pas dues à des \nom}
\text{rep = position(\nom,\listchoix[;2])}
\if{\rep=}{\text{rep = 7}}

\statement{Les maladies infectieuses peuvent être dues à des virus,
des bactéries ou des parasites qui se multiplient dans l'organisme.
<p> Parmi les maladies suivantes, sélectionner toutes les maladies
qui sont dues à des \nom :
<ul>
\for{h = 1 to 6}{
<li> \embed{reply1,\h}</li>
}
</ul>
\embed{reply1,7}
}

\answer{}{\rep;\choix}{type=checkbox}{option=split}

```

Exercice B

```

\title{Schémas du cycle cellulaire (feedback)}
\text{phases = l'interphase,la prophase,la prométaphase,
la métaphase,l'anaphase,la télophase,la cytodierèse}
\integer{n = randint(1..7)}
\text{liste = interphase.jpg,prophase.jpg,
prometaphase.jpg, metaphase.jpg,anaphase.jpg,
telophase.jpg,cytodierese.jpg}
\text{choix = \liste[\n]}

\statement{Le schéma suivant
<center> \img{\imagedir/\choix} </center>
décrit une cellule animale dans une des phases du

```

```

cycle cellulaire. Laquelle ?
<p><center>\embed{reply1}</center></p>
<i> NB : les choix sont présentés dans l'ordre
alphabétique</i>}

\answer{}{\n;\phases}{type = radio}{option = sort}
\integer{k=position(\reply1,\phases)}
\text{imgrep=\liste[\k]}
\feedback{\k<>\n}{<font color=red> Attention ! Le schéma correspondant à \phases[\k] est :
</font> <center> <img src=\imagedir/\imgrep></center>}}

```

Exercice 3.3

```

\title{Tangente 2}
\range{-5,5}
\real{a = random(-1,1)*randint(1..30)/10}
\real{b = randint(-20..20)/10}
\real{c = randint(-20..20)/10}
\function{f = \a*x^2+simplify(\b*x)+\c}
\real{x0 = randint(-40..40)/10}
\rational{y0 = evaluate(\f,x = \x0)}
\function{df = diff(\f,x)}
\rational{df0 = evaluate(\df,x = \x0)}
\function{D = \df0*x-simplify(\df0*\x0-\y0)}
\integer{xmin = min(-3,\x0-2)}
\integer{xmax = max(3,\x0+2)}
\text{A = slib(function/bounds \f, x,\xmin,\xmax)}
\integer{ymin = min(-3,A[1]-2)}
\integer{ymax = max(3,A[2]+2)}
\text{rangex = \xmin,\xmax}
\text{rangey = \ymin,\ymax}

\text{dessin = rangex \rangex
rangey \rangey
arrow \xmin,0,\xmax,0,10, black
arrow 0,\ymin,0,\ymax,10, black
plot navy,\f
circle \x0,\y0,5, red
text black,0,0,roman,0
}
\text{url = draw(200,200
\dessin)}
\text{dessinc = \dessin
plot green,\D
}
\text{urlc = draw(200,200
\dessinc)}

\statement{ Donner l'équation de la tangente à la courbe d'équation \ (y = \f\ )

```

```

    au point d'abscisse \x0.
    <p><center></center></p>
}
\answer{y={\D,x}{type=function}}
\text{dessinc = \dessin
plot green,\D
}
\solution{La droite tangente au point d'abscisse \x0 est dessinée en vert :
    <center></center> }

```

Exercice 3.4

```

\title{Cellule}
\text{Size = 625,320}
\matrix{coord = 245,102,1,-1,un reticulum endoplasmique
    50,107,1,1,un filament d'actine
    120,107,-1,-1,une mitochondrie
    70,180,-1,1,la membrane plasmique
    238,159,1,-1,des nucléoles
    298,139,1,-1,l'enveloppe nucléaire
    358,190,1,1,l'appareil de Golgi
    354,225,-1,1,une microtubule
    542,207,-1,1,une vésicule
    160,100,0,-1,le cytoplasme
}
\integer{k = rows(\coord)}
\integer{k = randint(1..\k)}
\text{co = pari([\coord[\k;1]+20*(\coord[\k;3]),\coord[\k;2]+20*\coord[\k;4],
    \coord[\k;1],\coord[\k;2]])}
\text{dessinprelim=
    xrange 0,\Size[1]
    yrange 0,\Size[2]
    copy 0,\Size[2],-1,-1,-1,-1,cellule.jpg
    arrow \co,10,blue}
\text{figure = draw(\Size
    \dessinprelim)
}
\statement{Le schéma ci-dessous représente une cellule eucaryote.
    <p><center>

    </center> </p>
Cliquer sur le nom de l'organite désigné par la flèche bleue :
<p><center>\embed{reply1,100x40x1}</center></p>
}
\answer{{\coord[\k;5];\coord[;5]}{type=dragfill}}

```

Exercice 3.5

```

\title{Champignon à couleurs variables}
\text{choix = red,pink,yellow,orange,brown,blue,white,purple,grey,black}

```

```

\text{a = shuffle(items(\choix))}
\text{couleur1 = \choix[\a[1]]}
\text{couleur2 = \choix[\a[2]]}
\text{size = 200,310}
\text{image = draw(\size[1],\size[2]
  xrange 0,\size[1]
  yrange 0,\size[2]
  setparallelogram 0,0,\size[1],0,0,\size[2]
  multicopy champignon.jpg
  fill 100,100,\couleur2
  fill 60,180,\couleur1 )}

\statement{Quelles sont les couleurs du champignon en anglais ?
  <table>
    <tr>
      <td rowspan=2></td>
      <td>\embed{reply1}</td>
    </tr>
    <tr>
      <td>\embed{reply2}</td>
    </tr>
  </table>
}

\answer{}{\a[1];\choix}{type=radio}
\answer{}{\a[2];\choix}{type=radio}

```

Exercice 3.6

```

\title{Quel est le nombre écrit dans la cellule ...}
\text{nb = 8,6}
\matrix{A = slib(matrix/random \nb,100)}
\matrix{A = pari(abs([\A]))}
\integer{x = randint(1..\nb[2])}
\integer{y = randint(1..\nb[1])}
\text{liste = A,B,C,D,E,F}
\text{a = \liste[\x]}
\integer{rep = \A[\y;\x]}
\text{Size = 563,341}
largeur et hauteur d'une case
\text{lx = 80}
\text{ly = 17}
abscisse de la première cellule
\text{premier = 111,206}
\text{coord_x = wims(values \premier[1] + x * \lx for x = 0 to \nb[2]-1)}
\text{coord_y = wims(values \Size[2] - \premier[2] - y*\ly
  for y = 0 to \nb[1]-1)}

\text{dessinprelim=
  xrange 0,\Size[1]

```

```

        yrange 0,\Size[2]
        copy 0,\Size[2],-1,-1,-1,-1,tableur.jpg}
\text{dessin=}
\for{j = 1 to \nb[2]}{
  \for{i = 1 to \nb[1]}{
    \text{dessin = \dessin
      text black,\coord_x[\j],\coord_y[\i],,A[\i;\j]}
  }
}
\text{tableau = draw(\Size
\dessinprelim
\dessin)}
\statement{Voici une image de page de <b>tableur</b>.
  <p align="center"></p>
  Quel est le nombre écrit dans la cellule \a\y ?
  <p align="center"> \embed{r1,5} </p>
}

\answer{Case \a\y}{\rep}

```

Exercice 3.7

```

\title{Champignon à couleurs variables (clickfill)}
\text{choix = red,pink,yellow,orange,brown,blue,white,purple,grey,black}
\text{a = shuffle(items(\choix))}
\text{couleur1 = \choix[\a[1]]}
\text{couleur2 = \choix[\a[2]]}
\text{size = 200,310}
\text{image = draw(\size[1],\size[2]
  xrange 0,\size[1]
  yrange 0,\size[2]
  setparallelogram 0,0,\size[1],0,0,\size[2]
  multicopy champignon.jpg
  fill 100,100,\couleur2
  fill 60,180,\couleur1 )}
\statement{Quelles sont les couleurs du champignon en anglais ?
  <p align="center">
    \special{imagefill \image, 200x310,60x20
      reply1,100x100
      reply2,60x180
    }
  </p>
}

\answer{}{\couleur1 ;\choix}{type=clickfill}
\answer{}{\couleur2 ;\choix}{type=clickfill}

```

Exercice 3.8

```

\title{Quelle est la case ?}

```

```

\text{nb = 8,6}
\text{case = randint(1..8),randint(1..6)}
\integer{x = randint(1..\nb[2])}
\integer{y = randint(1..\nb[1])}
\text{liste = A,B,C,D,E,F}
\text{a = \liste[\x]}
\text{Size = 563,341}
\text{lx = 80}
\text{ly = 17}
\text{premier = 70,219}
\integer{coord_x = \premier[1] + (\x - 1) * \lx}
\integer{coord_y = \premier[2] + (\y - 2) * \ly}
\integer{Coord_x = \coord_x + \lx}
\integer{Coord_y = \coord_y + \ly}
\text{rect = \coord_x,\coord_y,\Coord_x,\Coord_y}
\statement{
Voici une image de page de <b>tableur</b>. Cliquer sur la cellule \a\y.
<p align="center"> \embed{r1, \Size[1] x \Size[2]} </p>
}

\answer{}{\imagedir/tableur.jpg;rectangle, \rect}{type=coord}

```

Exercice 3.9

```

\title{Tangente 3}
\range{-5,5}
\real{a = random(-1,1)*randint(1..30)/10}
\real{b = randint(-20..20)/10}
\real{c = randint(-20..20)/10}
\function{f = \a*x^2+simplify(\b*x) + \c}
\real{x0 = randint(-40..40)/10}
\real{y0 = evaluate(\f, x = \x0)}
\function{df = diff(\f,x)}
\real{df0 = evaluate(\df,x = \x0)}
\function{D = \df0*x - simplify(\df0*\x0 - \y0)}
\integer{xmin = min(-3, \x0 - 2)}
\integer{xmax = max(3, \x0 + 2)}
\text{A = slib(function/bounds \f, x,\xmin,\xmax)}
\integer{ymin = min(-3, \A[1] - 2)}
\integer{ymax = max(3, \A[2] + 2)}
\text{rangex = \xmin, \xmax}
\text{rangey = \ymin, \ymax}

\text{dessin = rangex \rangex
rangey \rangey
arrow \xmin,0,\xmax,0,10, black
arrow 0,\ymin,0,\ymax,10, black
plot navy,\f

```

```

        circle \x0,\y0,5, red
        text black,0,0,roman,0
    }
    \text{url = draw(200,200
        \dessin)}
    \text{dessinc = \dessin
        plot green,\D
    }
    \text{urlc = draw(200,200
        \dessinc)}
    \real{x1 = \x0+1}
    \real{y1 = \df0*\x1 - \df0*\x0 + \y0}
    \text{P0 = slib(draw/convpixel \x0,\y0,200,200,\rangex,\rangey,0,pixels)}
    \text{P1 = slib(draw/convpixel \x1,\y1,200,200,\rangex,\rangey,0,pixels)}
    \text{rep = \url;line,\P0[1],\P0[2],\P1[1],\P1[2]}

    \statement{En utilisant la souris, positionner la tangente à la courbe d'équation
    \(\text{y} = \text{f}\) au point d'abscisse \x0.<br>
    NB : <i> le tracé sera
    obtenu en définissant deux points appartenant à cette tangente</i>
    <center>\embed{reply1,200x200}</center></p>
    }
    \answer{}{\rep}{type=javacurve}

    \text{dessinc = \dessin
        plot green,\D
    }
    \solution{La droite tangente au point d'abscisse \x0 est dessinée en vert :
    <center></center> }

```

Exercice 3.10

```

\title{Réfraction par le dessin}
\author{Julien M}
\computeanswer{no}
\precision{1000}
\integer{i1 = randint(10..50)}
\integer{n1 = 1}
\real{n2 = randint(110..170)/100 }
\integer{i2 = (asin(\n1/\n2*sin(\i1*pi/180)))*180/pi}
\text{rangex = -2,2}
\text{rangey = -2,2}
\text{SIZE = 501,501}

\text{dessin =
    xrange \rangex
    yrange \rangey
    hline 0, 0, black
    vline 0, 0, black

```



```

    line 0,0, \rangey[1]*tan(\i1*pi/180), \rangey[2], red
    dline 0,0,\rangey[2]*tan(\i1*pi/180), \rangey[1], red
    text black,1.5,1,giant,n1
    text black,1.5,-1,giant,n2
    arc 0,0, 2*\n1, 2*\n1, 180,360, black
    arc 0,0, 2*\n2, 2*\n2, 180,360, black
}
\text{origine = slib(draw/convpixel 0, 0, \SIZE, \rangex, \rangey, 0, pixels)}
\text{sol = \rangey[2]*tan(\i2*pi/180),\rangey[1], \SIZE, \rangex, \rangey}
\text{sol = slib(draw/convpixel \sol, 0, pixels)}
\text{url = draw(\SIZE
\dessin)}}

\statement{Dessiner le rayon réfracté, connaissant l'angle d'incidence
\((i_1) = \i1^\circ\), et les indices de réfraction \((n_1) = \n1\) et \((n_2) = \n2\).
    <p align="center">\embed{reply1,\SIZE[1] x \SIZE[2]}</p>

}

\answer{{\url;sline,\origine[1],\origine[2],\sol[1],\sol[2]}}{type=javacurve}

```

Exercice 4.1

```

\title{Maladies infectieuses 3}
\language{fr}
\author{Sophie Lemaire}
\email{sophie.lemaire@math.u-psud.fr}
\format{html}

\matrix{liste = Tuberculose,bactéries
    Tétanos,bactéries
    Typhoïde,bactéries
    Lèpre,bactéries
    Rage,virus
    Poliomyélite,virus
    Rougeole,virus
    Hépatite,virus
    Grippe,virus
    Bronchiolite,virus
    Paludisme,parasites
    Toxoplasmose,parasites}
\text{nom = random(bactéries,virus,parasites)}
\text{mix = shuffle(rows(\liste))}
\text{listchoix = \liste[\mix[1..6];]}
\text{choix = \listchoix[;1],ces maladies ne sont pas dues à des \nom}
\text{rep = position(\nom,\listchoix[;2])}
\if{\rep=}{\text{rep = 7}}

\statement{Les maladies infectieuses peuvent être dues à des

```

```

\special{help virus, virus}, des \special{help bact,bactéries}
ou des \special{help para,parasites} qui se multiplient
dans l'organisme.
<p> Parmi les maladies suivantes, sélectionner toutes les maladies
qui sont dues à des \nom :
<ul>
  \for{h = 1 to 6}{
    <li> \embed{reply1,\h}</li>
  }
</ul>
\embed{reply1,7}
}
\help{
\if{\help_subject issametext bact or \help_subject issametext }
{<p>Les <b>bactéries</b> sont des organismes vivants unicellulaires
caractérisées par une absence de noyau et d'organites.<br>
Les infections bactériennes peuvent être traitées grâce aux antibiotiques
qui le plus souvent inhibent une de leurs fonctions vitales. </p> }
\if{\help_subject issametext virus or \help_subject issametext }
{<p>Un <b>virus</b> est une entité biologique qui nécessite une
cellule hôte, dont il utilise les constituants pour se multiplier.
Contrairement aux \special{help bact,bactéries}, ce n'est donc pas un
organisme vivant.
Les virus sont le plus souvent de très petite taille (comparée à celle
d'une bactérie par exemple).
Tous les êtres vivants peuvent être infectés par des virus
(les virus affectant des bactéries sont appelés des bactériophages)<br>
Les antibiotiques sont sans effet sur les virus.</p> }
\if{\help_subject issametext para or \help_subject issametext }
{<p> En biologie, un <b>parasite</b> est un organisme vivant qui se nourrit,
s'abrite ou se reproduit en établissant une interaction durable avec
un autre organisme (l'hôte). <br>
En médecine humaine et vétérinaire, on appelle <b>parasite</b> un métazoaire
ou un protozoaire parasitant l'organisme et entraînant une parasitose
(n'incluant donc ni \special{help virus,virus},
ni \special{help bact,bactérie}, ni champignon).}
}
\answer{}{\rep;\choix}{type=checkbox}{option=split}

```

Exercice 4.2

```

\title{Carrés}
\integer{n = randint(4..6)}
\integer{m = randint(3..4)}
\text{rep = wims(values x^2 for x = \m to \n + \m)}
\text{th = wims(makelist \ (x^2) for x = \m to \n + \m - 1)}
\text{ETAPE = wims(makelist r x for x = 1 to \n)}
\steps{\ETAPE}

```

```

\statement{Calculer les carrés des entiers à partir de \m :





```

Exercice 4.3

```

\title{Carré un par un}
\integer{n = randint(4..6)}
\integer{m = randint(3..4)}
\text{question = wims(values x for x = \m to \n + \m)}
\text{rep = wims(values x^2 for x = \m to \n + \m)}
\text{th = wims(makelist \ (x^2) for x = \m to \n + \m - 1)}
\text{ETAPE = wims(makelist r x for x = 1 to \n)}
\text{ETAPE = wims(replace internal , by ; in \ETAPE)}
\steps{\ETAPE}

\statement{Calculer le carré de \question[\step] :

```

Exercice 4.4

```

\title{Lettre de l'alphabet}
\text{alphabet = a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}
\integer{n = randint(1..20)}
\integer{try = 0}
\text{REP=}
\text{NETAPE = r1}
\nextstep{\NETAPE}

\statement{Ecrire la lettre de l'alphabet qui suit la lettre \alphabet[\n] :

```

```

    \if{ r1 isitemof \NETAPE}{
      \embed{r1,5} <font color=red>\REP</font>
    }{
      <font color=green>\alphabet[\n + 1]</font>
    }
  </p>
}

\answer{lettre suivant \alphabet[\n]}{\A}{type=nocase}
\condition{{\A issametext \alphabet[\n + 1]}}
\text{REP = \reply1}
\text{NETAPE=}
\text{NETAPE = \reply1 notsametext \alphabet[\n + 1] and \try < 3 ? r1}

\feedback{\try > 1}{Vous avez fait \try essais}
\integer{try = \try + 1}

```

Exercice 4.4

```

\title{Deux lettres de l'alphabet (plusieurs essais)}
\text{alphabet = a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}
\integer{n = randint(1..20)}

\text{NETAPE = r1,r2}
\nextstep{\NETAPE}
\integer{try = 1}
\text{ordinal = Premier essai : , Deuxième essai : , Troisième essai : }
\text{REP=}
\statement{Ecrire les deux lettres de l'alphabet suivant la lettre \alphabet[\n] :
<p>
\ordinal[\try]
<table align=center><tr><td> \alphabet[\n]</td>
  \for{a = 1 to 2}{
    <td>
      \if{ r\ a isitemof \NETAPE}{
        \embed{r\ a,5} <font color=red>\REP[\a]</font>
      }{
        <font color=green>\alphabet[\n + \a]</font>
      }
    </td>
  }
</table>
}

\answer{suivant de \alphabet[\n]}{\A}{type=raw}
\answer{suivant de \alphabet[\n+1]}{\B}{type=raw}
\condition{Le suivant de \alphabet[\n] est \A ? }{\A issametext \alphabet[\n + 1]}
\condition{Il y a ensuite \B}{\B issametext \alphabet[\n + 2]}

```

```

\text{REP = \reply1,\reply2}
\text{NETAPE = }
\for{b = 1 to 2}{
  \text{NETAPE = \REP[\b] notsametext \alphabet[\n+\b] and \try < 3 ?
    wims(append item r\b to \NETAPE)}
}
\integer{try1 = \try -1}
\feedback{\try1 > 1}{Vous avez fait \try1 essais}
\integer{try = \try + 1}

```

Exercice 4.4

```

\title{Lettres (échecs visibles dans l'analyse)}
\text{alphabet = a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}
\integer{n = randint(1..20)}
\text{texte = Première lettre après \alphabet[\n], Deuxième lettre après \alphabet[\n],
  Première lettre après \alphabet[\n], Deuxième lettre après \alphabet[\n],
  Première lettre après \alphabet[\n], Deuxième lettre après \alphabet[\n]}

\text{NETAPE = r1,r2}
\nextstep{\NETAPE}
\integer{try = 0}
\text{REP = }
\text{COND = 1,2}
\conditions{\COND}

\statement{
  Ecrire dans l'ordre les deux lettres de l'alphabet
  qui suivent la lettre <font size=+2>\alphabet[\n]</font> :
  <ul>
    \for{a = 1 to 6}{
      \if{ r\a isitemof \NETAPE}{
        <li>
          \if{\step>1}{Votre réponse précédente
            <font color=red>\REP[\a-2]</font> est fausse, recommencez :}
          \texte[\a] : \embed{r\a,5}
        </li>
      }
    }
  </ul>
}

\answer{1-1}{\A1}{type=raw}
\answer{1-2}{\B1}{type=raw}
\answer{2-1}{\A2}{type=raw}
\answer{2-2}{\B2}{type=raw}
\answer{3-1}{\A3}{type=raw}
\answer{3-2}{\B3}{type=raw}
\condition{Premier essai lettre 1}{\A1 issametext \alphabet[\n + 1]}

```

```

\condition{Premier essai lettre 2}{\B1 issametext \alphabet[\n + 2]}
\condition{Deuxième essai lettre 1}{\A2 issametext \alphabet[\n + 1]}
\condition{Deuxième essai lettre 2}{\B2 issametext \alphabet[\n + 2]}
\condition{Troisième essai lettre 1}{\A3 issametext \alphabet[\n + 1]}
\condition{Troisième essai lettre 2}{\B3 issametext \alphabet[\n + 2]}

\text{REP = \A1,\B1,\A2,\B2,\A3,\B3}

\text{NETAPE=}
\for{b = 1 to 2}{
  \integer{u = 2*(\step-1) + \b}
  \integer{v = 2*(\step-2) + \b}
  \if{\REP[\v] != \alphabet[\n + \b]}{
    \text{NETAPE = \try < 2 ? wims(append item r\u to \NETAPE)}
    \text{COND = \try < 2 ? wims(append item \u to \COND)}
  }
}

\feedback{\try > 1}{Vous avez eu besoin de \try essais. }
\feedback{1 = 1}{
  <div style="background:#FF33FF; margin:0% 20%;">Les deux lettres qui suivent
    <font size=+2>\alphabet[\n]</font> sont
    <font size=+2>\alphabet[\n + 1]</font> et
    <font size=+2>\alphabet[\n + 2]</font>.
  </div>
}
\integer{try = \try + 1}

```

Exercice 4.5

```

\title{Carrés (nonstop)}
\integer{n = 3}
\integer{m = randint(3..4)}
\text{question = wims(values x for x = \m to \n + \m)}
\text{rep = wims(values x^2 for x = \m to \n + \m)}
\text{th = wims(makelist \((x^2) for x = \m to \n + \m - 1)}
\text{ETAPE = r1}
\text{cnt_r = 1}
\nextstep{\ETAPE}

\statement{
  \if{\step = 1}{Calculer le carré de \question[1] :
    <p align="center"> \embed{r 1, 5} </p>
  }
  \if{\step = 2}{Calculer le carré de \question[2]
    <p align="center"> \embed{r \ETAPE[1], 5} </p>
    \if{\cnt_r = 2}{
      et le carré de \question[3]
      <p align="center"> \embed{r \ETAPE[2], 5} </p>
    }
  }
}

```

```

    }
  }
}

\answer{\th[1]}\{rep[1]}{type=numeric}{option=nonstop}
\answer{\th[2]}\{rep[2]}{type=numeric}{option=nonstop}
\answer{\th[3]}\{rep[3]}{type=numeric}{option=nonstop}

\if{\step = 2}{
  \text{ETAPE = \rep[1] = \reply1 ? r2 : r2,r3}
}{
  \text{ETAPE=}
}

\feedback{1=1}\{rep[1] = \reply1}

\integer{cnt_r = items(\ETAPE)}

```

Exercice 4.6

```

\title{Cellule 2}
\text{Size = 625,320}
\matrix{coord = 245,102,1,-1,un reticulum endoplasmique
50,107,1,1,un filament d'actine
120,107,-1,-1,une mitochondrie
70,180,-1,1,la membrane plasmique
238,159,1,-1,des nucléoles
298,139,1,-1,l'enveloppe nucléaire
358,190,1,1,l'appareil de Golgi
354,225,-1,1,une microtubule
542,207,-1,1,une vésicule
160,100,0,-1,le cytoplasme
}
\integer{k = randint(1..10)}
\text{co = pari([\coord[\k;1]+20*(\coord[\k;3]),\coord[\k;2]+20*\coord[\k;4],
  \coord[\k;1],\coord[\k;2]])}
\text{dessinprelim=
xrange 0,\Size[1]
yrange 0,\Size[2]
copy 0,\Size[2],-1,-1,-1,-1,cellule.jpg
arrow \co,10,blue}
\text{figure = draw(\Size
\dessinprelim)}
\text{NETAPE = r1}
\nextstep{\NETAPE}
\text{verif=}
\statement{Le schéma ci-dessous représente une cellule eucaryote.
\if{\step = 1}{
  <p><center>

```

```

    
  </center></p>
}
{<p><center></center></p>}
Cliquer sur le nom de l'organite désigné par la flèche bleue :
<p><center>\embed{reply1,100x40x1}</center></p>
}

\answer{}{\var;\coord[;5]}{type=dragfill}
\text{NETAPE=}
\integer{r = position(\reply1,\coord[;5])}
\condition{}{\r = \k}{option=hide}
\text{coul = \r==\k ? blue : red}
\text{co = \coord[\r;1]+20*(\coord[\r;3]),\coord[\r;2]+20*\coord[\r;4],
  \coord[\r;1],\coord[\r;2]}
\text{verif = draw(\Size
\dessinprelim
arrow \co,10,\coul)}

\feedback{\r!=\k }{<font color=red> Erreur ! La flèche bleue n'indique
pas \reply1, mais \coord[\k;5]. Sur le dessin ci-dessus,
la flèche rouge vous montre \reply1.
</font>}}

```

Exercice 4.7

```

\title{Le compte est bon}
\integer{n = 3} Nombre de chiffres choisis
\text{chiffres = wims(makelist x for x = 2 to 10)}
\text{list = slib(data/random \n,item,\chiffres)}
\text{op=+,*}
\integer{c = random(1,2)}
\text{opc1 = wims(randitem \op)}
\text{nb = slib(data/random 3,item,\list)}
\if{c = 1}
  {\integer{val = \nb[1]\opc1\nb[2]}}
  {\text{opc2 = wims(randitem \op)}}
  {\integer{val = (\nb[1]\opc1\nb[2])\opc2\nb[3]}}
}
\if{\val isitemof \list}{\text{opc1=*}\text{opc2=*}}
  {\integer{val = (\nb[1]\opc1\nb[2])\opc2\nb[3]}}
}

\text{NETAPE = r1}
\nextstep{\NETAPE}
\text{COND = 1}
\conditions{\COND}
\statement{

```



```

En additionnant ou en multipliant certains de ces chiffres
  <center><b>\list</b></center>
vous devez obtenir <b>\val</b>.
Chaque chiffre ne pourra être utilisé qu'une fois.<br>
\if{\step = 1}{De combien d'opérations aurez-vous besoin ? \embed{r1,2}}
  {\if{\NETAPE!=}
    { Entrez votre formule :
      \if{\nop = 1}{ (\embed{r2,2} \embed{r3} \embed{r4,2}) }
      { (\embed{r2,2} \embed{r3} \embed{r4,2}) \embed{r5}\embed{r6,2} }
    }
  }
}
\answer{}{\nop}{type=numeric}
\answer{}{\ch1}{type=numeric}
\answer{}{\oper1;\op}{type=menu}
\answer{}{\ch2}{type=numeric}
\answer{}{\oper2;\op}{type=menu}
\answer{}{\ch3}{type=numeric}

\condition{nb d'opérations}{\nop = 1 or \nop = 2}{option=hide}

\if{\step = 2}
  {\if{\nop = 1}{\text{NETAPE = r2,r3,r4}
    \text{COND = 2,3}}
    {\if{\nop = 2}{\text{NETAPE = r2,r3,r4,r5,r6}
      \text{COND = 2,3}}
      {\text{NETAPE = }}
    }
  }
  {\text{NETAPE = }}
\integer{res = \nop = 1 ? \ch1 \oper1 \ch2 : (\ch1 \oper1 \ch2)\oper2 \ch3}
\text{rep = \ch1,\ch2}
\text{rep = \nop = 2 ? \rep,\ch3}
\text{rep = wims(listuniq \rep)}
\text{repcomp = wims(listcomplement \list in \rep)}
\integer{uniq = items(\rep) - 1}
\condition{opération correcte ?}{\res = \val}
\condition{Vous avez respecté la consigne}{\uniq = \nop and \repcomp=}{weight=2}

\feedback{\nop > 2}{Vous n'avez pas à faire plus de deux opérations}
\feedback{\step = 3 and \uniq!=\nop}{Vous n'avez pas respecté la consigne :
  vous avez utilisé plusieurs fois le même chiffre.}
\feedback{\step = 3 and \repcomp!=}{Vous n'avez pas respecté la consigne :
  vous avez utilisé un chiffre qui n'était pas proposé}
\feedback{\step = 3 and \res!=\val}{Votre calcul donne \res et non \val}

```