

oef-mode

mode majeur d'Emacs pour éditer des exercices et des documents interactifs
au format « online exercise format » (version alpha "20240513.1955")

Raoul Hatterer

This is the **oef-mode** manual, version alpha (in french) "20240513.1955"

OEF veut dire "Online Exercise Format", un format pour d'exercices et de documents interactifs destinés à l'enseignement assisté par ordinateur.

Le format d'un fichier OEF est très similaire à LaTeX. Il est composé d'instructions suivies d'un ou plusieurs paramètres. Le mode **oef-mode** pour Emacs a pour ambition d'être un outil efficace pour la création et la mise en ligne de documents ou d'exercices sous ce format.

Copyright © 2017 - 2024 Raoul Hatterer.

Table des matières

1	Introduction	1
1.1	Résumé	1
1.2	Installations	1
1.2.1	Installation d'emacs	1
1.2.1.1	Sous Windows	1
1.2.1.2	Sous Mac OSX	1
1.2.1.3	Sous Linux	1
1.2.2	Fichier d'initialisation	1
1.2.3	Gestionnaires de packages	2
1.2.4	Configurer le clavier du Mac	3
1.2.5	Installer oef-mode	3
1.2.6	Installer oef-mode manuellement	4
1.2.7	Autres installations	4
1.2.7.1	Lier emacs à la corbeille	4
1.2.7.2	EMMET-MODE (Zen-coding)	5
1.3	Activation	5
2	Le compte Modtool	6
2.1	Posséder un compte développeur	6
2.2	Ajout d'un exercice OEF	6
2.3	Utiliser des fichiers de données	6
2.3.1	Référence par index	6
2.3.2	Dictionnaire	7
3	Les documents Wims	8
4	Les exercices Wims	10
4.1	L'en-tête	10
4.2	Les commentaires	10
4.3	Les variables	10
4.4	Le texte de l'énoncé	10
4.5	Les aides (partie facultative)	10
4.6	Les réponses	11
	Index	12

1 Introduction

Le présent manuel est disponible au format PDF, mais il est également intégré, une fois le package oef-mode installé, au système d'information d'Emacs accessible par le raccourci `C-h i`.

1.1 Résumé

`oef-mode` est un mode majeur permettant d'éditer des exercices et des documents au format « online exercise format ».

Il propose un menu et des raccourcis permettant d'accéder aux commandes, fonctions, scripts... spécifiques au format OEF.

1.2 Installations

1.2.1 Installation d'emacs

Cette partie est à destination des lecteurs du document PDF qui, connaissant peut-être déjà Wims et le format OEF, souhaiteraient tester `oef-mode`. Ils trouveront ici des indications pour installer Emacs.

1.2.1.1 Sous Windows

- Déterminez si votre Microsoft Windows est 32 ou 64 bits: pour cela, appuyez sur la touche `Window` ; commencez à taper “Afficher le système d'exploitation exécuté par votre ordinateur” dans la zone de recherche et sélectionnez ce choix. Cela fait apparaître une fenêtre dans laquelle le type de système (32 ou 64 bits) est indiqué.
- Rendez-vous ensuite à la page de téléchargement: <https://ftp.gnu.org/pub/gnu/emacs/windows/emacs-29/> et téléchargez l'installateur:
 - `emacs-$VERSION-x86_64-installer.exe` si votre système d'exploitation est 64 bits
 - `emacs-$VERSION-i686-installer.exe` si votre système d'exploitation est 32 bits
- Lancez le programme d'installation pour installer Emacs sur votre ordinateur.

1.2.1.2 Sous Mac OSX

Sous mac OSX, vous pouvez installer la version que vous trouverez ici: <https://emacsformacosx.com>

1.2.1.3 Sous Linux

```
sudo apt-get install emacs
```

1.2.2 Fichier d'initialisation

Le fichier d'initialisation est un fichier qu'Emacs charge lorsqu'il démarre. Il est utilisé pour personnaliser Emacs.

Sous Linux et Mac, par défaut, Emacs cherche le fichier d'initialisation aux emplacements suivants :

- `~/ .emacs`

- `~/ .emacs.el`
- `~/ .emacs.d/init.el` (depuis 2010 il s'agit de l'emplacement à privilégier)

Sous Microsoft Windows:

- `%HOMEPATH%\AppData\Roaming\ .emacs.d\init.el` (emplacement à privilégier)
- `%HOMEPATH%\AppData\Roaming\ .emacs`

1.2.3 Gestionnaires de packages

Le paquet (package) `oef-mode` ne faisant pas partie des paquets pré-installés avec emacs, il vous faudra l'installer. De plus, `oef-mode` n'étant pas répertorié dans `elpa`, le dépôt de packages accessible par défaut dans emacs, je recommande d'installer `el-get` <http://wikemacs.org/wiki/El-get> un gestionnaire de package alternatif car le package `oef-mode` s'y trouve répertorié. Pour cela, nous utiliserons un fichier d'initialisation chargé en tout premier appelé `early-init.el`.

Dans Emacs saisir la commande `C-x C-f` Find file: `~/ .emacs.d/early-init.el` (ou `%HOMEPATH%\AppData\Roaming\ .emacs.d\early-init.el` sous Microsoft Windows) pour créer un nouveau buffer nommé `early-init.el` et placez y le code suivant (`C-y` permet de coller(yank) dans Emacs) destiné à installer le gestionnaire de package `el-get`:

```
(add-to-list 'load-path "~/ .emacs.d/el-get/el-get")

(unless (require 'el-get nil 'noerror)
  (require 'package)
  (add-to-list 'package-archives
    '("melpa" . "http://melpa.org/packages/"))
  (package-refresh-contents)
  (package-initialize)
  (package-install 'el-get)
  (require 'el-get))

(add-to-list 'el-get-recipe-path "~/ .emacs.d/el-get-user/recipes")
(el-get 'sync)
```

- Sélectionner l'ensemble du code avec `C-x h`
- Indenter le code avec `M-x: indent-region`
- Sauvegardez avec `C-x C-s`
- Quittez Emacs avec `C-x C-c`

Microsoft Windows:

- Lancez Emacs. Patientez le temps de l'installation de `el-get`. Quittez Emacs. Relancez Emacs.

Mac OSX:

- Lancez Emacs à partir du Terminal:
(`/Applications/Emacs.app/Contents/MacOS/Emacs &`).
- Patientez le temps de l'installation de `el-get`. Quittez Emacs. Relancez Emacs.

1.2.4 Configurer le clavier du Mac

Vous vous apercevrez que, par défaut, les deux touches *option* ont été configurées en touche Meta *M-* pour emacs ce qui fait que vous avez perdu l'accès aux accolades (*option parenthèse*), aux crochets (*option shift parenthèse*) et aux autres caractères auxquels vous aviez accès par le biais de la touche *option*.

Éditer le fichier d'initialisation (`~/ .emacs.d/init.el`) et rajouter les lignes suivantes qui permettent d'avoir la touche *option* de gauche qui agit comme *Meta* tandis que la touche *option* de droite reste inchangée pour pouvoir accéder à des caractères comme les accolades ou les crochets.

Pour cela, tapez `C-x C-f` (c'est-à-dire *control* x suivi de *control* f) dans Emacs.

Find file: apparaît sous la fenêtre. Saisir `~/ .emacs.d/init.el` et valider.

Coller le code suivant, sans indentation:

```
;; mac specific bindings
(setq mac-right-command-modifier nil ; cmd de droite = cmd
      mac-command-modifier nil ; cmd de gauche = cmd
      mac-option-modifier 'meta ; option de gauche = meta
      mac-right-option-modifier nil ; option de droite = car. spéciaux
      mac-control-modifier 'control ; control de gauche = control
      ns-function-modifier 'hyper ; fn key = hyper
      ns-right-alternate-modifier nil); cette touche n'existe pas

;; INDENT-REGION on OSX
(global-set-key (kbd "C-:") 'indent-region)

(load-library "iso-transl") ; pour avoir l'accent circonflexe

;; exec-path
(add-to-list 'exec-path "/usr/local/bin")

;; load-path
(add-to-list 'load-path "~/ .emacs.d/lisp")
```

Sauvegardez avec `C-x C-s` puis quitter Emacs (`C-x C-c`), relancez Emacs et vérifiez que vous pouvez avoir les accolades.

Si un fichier `~/ .emacs` a été créé dans votre dossier personnel, il faut l'effacer car il prend le pas sur le fichier `~/ .emacs.d/init.el`.

1.2.5 Installer oef-mode

- Créer ou éditer le fichier d'initialisation (`~/ .emacs.d/init.el`) et rajouter les lignes suivantes qui permettent d'installer *oef-mode*, et *use-package*, s'ils ne le sont pas déjà:

```
(setq my:el-get-packages
      '(cl-lib
        el-get
          oef-mode
          use-package
        ))

(el-get 'sync my:el-get-packages)

(use-package company
  :ensure t
  :config
  (add-hook 'after-init-hook 'global-company-mode)
)
;; global activation of the unicode symbol completion

;; Permet d'activer le mode oef-mode quand on visite les
;; fichiers avec l'extension .oef
(load "oef-mode.el")
```

- Quitter et relancer emacs

La dernière version `oef-mode` vient d'être installée. Pour faire des mise à jour dans le futur, il y a la commande `M-x: el-get-update oef-mode`

1.2.6 Installer oef-mode manuellement

Si vous utilisiez déjà Emacs, sans `el-get` ni `use-package`, une alternative est d'installer `oef-mode` manuellement:

- S'il n'existe pas encore, créer le répertoire `~/ .emacs.d/lisp/`
- Télécharger l'archive zip <https://github.com/raoulhatterer/oef-mode/archive/master.zip>, la décompresser dans le répertoire `~/ .emacs.d/lisp/`
- Ajouter le code suivant à votre fichier `~/ .emacs.d/init.el` ou `.emacs` avant la ligne `'(package-initialize)'`.

```
;; Tell emacs where is your personal elisp lib dir
(add-to-list 'load-path "~/ .emacs.d/lisp/oef-mode")
(load "oef-mode.el") ;; load the package named oef-mode
```

- Quitter et relancer emacs.

Par convention, le répertoire `~/ .emacs.d/lisp/` est la destination des bibliothèques installées de façon manuelle.

1.2.7 Autres installations

1.2.7.1 Lier emacs à la corbeille

Insérer le code suivant dans votre fichier `init` pour que les fichiers supprimés dans emacs aillent dans la corbeille du Mac:

```
;;; Les fichiers qu'on supprime dans emacs vont dans la corbeille
(use-package osx-trash
  :ensure t
  :config
    (osx-trash-setup)
    (setq delete-by-moving-to-trash t)
)
```

1.2.7.2 EMMET-MODE (Zen-coding)

EMMET-MODE est une méthode puissante pour coder rapidement du HTML

Par exemple, le snippet `ul>li.item$*3` suivi de `C-j` permet de transformer le snippet en tags HTML:

```
<ul>
  <li class="item1"></li>
  <li class="item2"></li>
  <li class="item3"></li>
</ul>
```

Consulter la page suivante pour découvrir d'autres possibilités: <https://github.com/smhica/emmet-mode>

Pour disposer de raccourcis claviers, placer le code suivant dans votre fichier init d'emacs:

```
;;; EMMET-MODE
(defvar emmet-mode-keymap
  (let
    ((map (make-sparse-keymap)))
    (define-key map (kbd "C-j") 'emmet-expand-line)
    (define-key map (kbd "<C-M-right>") 'emmet-next-edit-point)
    (define-key map (kbd "<C-M-left>") 'emmet-prev-edit-point)
    (define-key map (kbd "C-c w") 'emmet-wrap-with-markup)
    map)
  "Keymap for emmet minor mode.")
```

Raccourcis disponibles:

`C-j` Transforme un snippet en tags

`<C-M-right>`
 Point d'édition suivant

`<C-M-left>`
 Point d'édition précédent

`C-c w` Transforme le mot sélectionné en tag

1.3 Activation

- Pour activer et désactiver le mode: `M-x: oef-mode`
- Quand vous créez un fichier avec l'extension `.oef` le mode s'active automatiquement.

2 Le compte Modtool

2.1 Posséder un compte développeur

Pour pouvoir développer des modules avec Modtool il faut demander un compte développeur par email au gestionnaire du site.

Vous recevrez un login et un mot de passe, vous permettant de vous connecter pour créer des ressources.

Vous pouvez alors créer un module en cliquant sur le lien *créer un nouveau module*.

2.2 Ajout d'un exercice OEF

Pour mettre un nouvel exercice dans le module :

1. Utilisez un mot court et significatif. Évitez les caractères spéciaux et les noms trop longs. Cela simplifie les processus de contrôle du code. par exemple `'toto'`.
2. Dans le champ **'Ajouter un autre fichier'**, mettez ce nom précédé de `'src/'` et suivi de l'extension `'.oef'`, dans notre exemple cela donne: `'src/toto.oef'`.
3. Dans la fenêtre qui apparaît, copiez le source de votre exercice.
4. Cliquez sur **'Enregistrer'**.

Attention : `'src'` est un nom imposé de répertoire dans lequel doivent être toutes les sources d'exercices OEF : il est important que le nom du fichier se termine par `'.oef'` et qu'il se trouve dans le répertoire `'src'`, c'est-à-dire que l'adresse soit `'src/xxx.oef'`

Vous pouvez alors tester votre exercice en cliquant **'Tester le module'**.

2.3 Utiliser des fichiers de données

On peut stocker des données dans un fichier que nous appellerons ici **'tableau'** pour les utiliser ensuite dans un ou plusieurs exercices.

2.3.1 Référence par index

Plutôt que d'avoir les phrases dans le fichier d'exercice, créez un fichier d'adresse tableau (attention, cette fois-ci ne pas mettre src) et mettez-y le texte suivant:

```
:Je suis grand ??et,est,ai?? brun.
Le café ??est,et,ai?? chaud.
La fleur ??est,et,ai?? rouge.
Le soleil ??est,et,ai?? chaud
J'??ai,est,et?? chaud

:L'élève ??est,et?? sérieux ??et,est?? appliqué.
Paul se ramasse ??et,est?? se relève.
Le pull ??est,et?? chaud ??et,est?? léger.
La rue ??est,et?? étroite ??et,est?? en pente.
Pierre ??est,et?? fort en mathématiques ??et,est?? en français.
La lumière sur le port ??est,et?? très belle en hiver ??et,est?? en été.
```

Remarquez le signe de ponctuation ‘:’. Il va être possible maintenant dans un exercice de charger ces données soit en entier, soit sélectivement pour la partie entre deux signes ‘:’ consécutifs. Donnons les premières commandes permettant cette utilisation:

```
\text{Tableau = wims(record 1 of tableau)}
\text{a = randomrow(\Tableau)}
```

record 1 chargera les cinq premières lignes du tableau, plus exactement celles entre le premier : et le second.

```
\text{Tableau = wims(record 2 of tableau)}
\text{a = randomrow(\Tableau)}
```

record 2 chargera les six lignes suivantes (donc ici les 6 dernières) du tableau (il faudra bien sûr changer la suite de l’exercice qui ne fonctionnera plus)

```
\text{Tableau = wims(record 0 of tableau)}
\text{a = randomrow(\Tableau)}
```

record 0 chargera toutes les lignes du fichier tableau

```
\integer{m = wims(recordcnt tableau)}
\matrix{Tableau = }
\for{ i = 1 to \m}{
\text{Tableau = \Tableau
randrow(wims(record \i of tableau))}
```

Ici, on compte le nombre d’enregistrements avec **recordcnt** (ici 2, mais il pourrait y en avoir plus). Et on prend une ligne par type. On pourra donc poser une question de chaque type.

2.3.2 Dictionnaire

Donnons un autre exemple d’utilisation de tableau extérieur aux exercices. Il s’agit de tableau de type "dictionnaire" où des clés sont associées à des valeurs (comme en python) de la forme suivante :

```
azote:.....
hydrogene:....
```

On peut alors aller chercher la ligne associée à azote avec la commande **lookup** suivie de la clé

```
\text{ligne= wims(lookup azote in tableau)}
```

et récupérer ensuite les informations qui suivent et dont vous avez besoin.

3 Les documents Wims

Il s'agit d'un ensemble de pages HTML (il vous faut donc connaître la syntaxe HTML et CSS) dans lesquelles peuvent être incluses des données calculées ou dessinées par les logiciels attendant à WIMS (des commandes supplémentaires étendant les possibilités de HTML permettent de faire cela). Les documents peuvent être utilisés pour proposer des cours avec des exemples à caractère aléatoire et des liens directs vers les exercices.

L'intérêt à faire un document WIMS plutôt que des pages html est donc la possibilité d'inclure:

- des exemples qui peuvent être renouvelés à la volée par le lecteur;
- des liens directs sur des exercices, ou sur des outils;
- des zones à déplier et replier (avec `\fold`) qui permettent à l'élève de se concentrer sur le plan, sur un théorème sans sa démonstration ou au contraire sur sa démonstration.

Pour homogénéiser la présentation des documents vous pouvez ajouter votre style CSS valable pour tout le document : pour cela, aller à Propriétés du document et copier votre fichier de style ou son adresse url.

Par exemple,

```
<style>
div.cours{
background-color : #faffe6;
margin : 0 5%;
padding : 3%;
border-width : thin;
}
</style>
```

Vous pouvez aussi ajouter de la même manière du code JavaScript.

Pour intégrer des applets présents sur un autre site à un document, il suffit de mettre l'adresse url des applets.

Comment classer les pages d'un document ?

L'option parent permet d'établir une hiérarchie entre les pages d'un document en plus des options usuelles précédent et suivant (par défaut, le parent est main). C'est commode pour gérer des "chapitres" et des tables de matières de chapitre.

Les formules TeX débordent. Que faire ?

Il vaut mieux couper les formules TeX un peu longues, $\backslash(...) = \backslash(...)$, plutôt que $\backslash(... = ...)$. Cela laisse au navigateur le choix de les mettre sur la même ligne ou non selon la taille de la fenêtre.

Dans un document, comment demander au lecteur d'entrer un paramètre ?

Voici un exemple où le paramètre à entrer est un entier :

```
\def{integer n = randint(2..9)}
\form{.}{exemple}{ n =
<input size = "6" name = "parm1" value = "\parm1">
<input type = "hidden" value = "OK">
}
```

```
\def{integer value = \parm1}  
\def{integer n = \value issametext NaN ? \parm1: \value}
```

Par défaut, l'entier n sera un entier entre 2 et 9.

J'ai fait un document dans ma classe, comment le citer ?

Dans un autre document, si le document cité a le numéro c9 : `\link{c9/main}` On peut aussi remplacer main le nom d'un autre bloc si on veut aller à une page particulière.

J'ai chargé un fichier pdf, ai fait un copier-coller du lien indiqué dans le document et le lien n'apparaît pas sur le nom du fichier.

Si vous avez fait un copier-coller du lien indiqué, il est possible que le blanc ait été mal codé. Enlevez les blancs et remettez-les !

4 Les exercices Wims

4.1 L'en-tête

Le début du fichier doit comporter un en-tête.

Exemple d'en-tête:

```
\title{SEUL SUR MARS}
\language{fr}
\range{-5..5}
\author{hatterer.med@laposte.net}
\computeanswer{no}
<!-- l utilisateur doit lui-même faire les calculs et entrer la valeur finale -->■
```

Les différentes commandes dont celles utilisées dans l'en-tête figurent dans le menu OEF>Commands.

4.2 Les commentaires

Dans la spécification W3C, un commentaire commence par '`<!--`' et se termine par '`-->`'

`M-`; permet d'insérer un commentaire.

4.3 Les variables

Après l'en-tête, on initialise les différentes variables introduites ou calculées.

C'est la partie qui fera appel à vos qualités de programmeur car elle correspond à «l'intelligence» de l'exercice.

4.4 Le texte de l'énoncé

```
\statement{...}
    pour écrire l'énoncé (cette commande est indispensable mais ne doit apparaître
    qu'une seule fois)
```

Pour ceux qui connaissent Python, c'est un peu comme si l'on avait le droit d'utiliser qu'une seule fois l'instruction `print()` mais qu'en revanche le texte à afficher pouvait être très très long et mis en forme au format HTML.

4.5 Les aides (partie facultative)

Cette partie facultative est à placer entre le texte de l'énoncé (Voir Section 4.4 [Le texte de l'énoncé], page 10) et les réponses (Voir Section 4.6 [Les réponses], page 11).

```
\hint {...}
    pour écrire une indication (ce qui apparaît lorsqu'on clique sur le lien
    Indications placé sous l'énoncé)
```

```
\help {...}
    pour un texte qui apparaîtra lorsque l'utilisateur clique sur Aide de la barre de
    menu en haut de l'exercice.
```

4.6 Les réponses

Après l'énoncé (ou les aides s'il y en a) on mettra la partie concernant la saisie et l'analyse des réponses.

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

pour analyser une réponse

Puisque généralement plusieurs réponses sont attendues, cette commande sera à écrire plusieurs fois (une fois par réponse attendue).

Vous pouvez bien sûr taper cette commande au clavier, mais sachez qu'oef-mode offre plusieurs possibilités pour insérer une réponse:

- menu OEF > Commands > **answer**
- *M-x* **oef-insert-answer**
- le raccourci *C-o-a*

Et pour sélectionner les paramètres entre accolades avec oef-mode:

- *M-x* **oef-select-parameter**
- le raccourci *C-o C-p*

Dans la première accolade, on peut mettre du texte qui apparaît devant le champ réservé à la réponse. Cel peut être un résumé de la question que l'on a peut-être posée de façon plus détaillée dans l'énoncé (Voir Section 4.4 [Le texte de l'énoncé], page 10). Pour ceux qui connaissent Python, c'est l'équivalent du texte que vous mettez entre les parenthèses de l'instruction `input()`. La réponse de l'utilisateur à la première question est automatiquement stockée dans la variable `\reply1` et ainsi de suite pour les autres réponses. Mais, généralement, vous n'allez pas faire appel à ces variables pour analyser les réponses de l'utilisateur. À la place, vous allez préciser entre les accolades qui suivent la réponse que vous attendez.

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 la troisième accolade.

Le menu OEF > **Answers Types and Options** liste et permet d'insérer les différents types et options que vous pouvez utiliser.

Pour une analyse plus détaillée sachez que la variable `\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.

Index

C

Compte développeur 6

E

exercice, OEF 6

G

gestionnaire, package 2

I

initialisation, init 1

installation 1

installation, emacs 1

installation, emacs, Linux 1

installation, emacs, Mac, OSX 1

installation, emacs, Windows 1

L

lookup 7

R

record, recordent 6