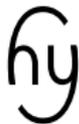


Diverses choses que je sais... sur les fichiers de style pour L^AT_EX

Le T_EXnicien de surface

U.S.T.L. & GUTenberg

Dunkerque 2008
version 1.1, 2008-07-03



STRUCTURE D'UN FICHIER `module.sty`

▸ `identification;`

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

STRUCTURE D'UN FICHIER `module.sty`

- ▶ `identification ;`
- ▶ `code d'initialisation ;`

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

STRUCTURE D'UN FICHIER `module.sty`

- ▶ `identification` ;
- ▶ `code d'initialisation` ;
- ▶ `déclaration des options` ;

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

STRUCTURE D'UN FICHER `module.sty`

- ▶ **identification ;**
- ▶ **code d'initialisation ;**
- ▶ **déclaration des options ;**
- ▶ **exécution des options ;**

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

STRUCTURE D'UN FICHER `module.sty`

- ▶ `identification` ;
- ▶ `code d'initialisation` ;
- ▶ `déclaration des options` ;
- ▶ `exécution des options` ;
- ▶ `chargement de modules` ;

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

STRUCTURE D'UN FICHER `module.sty`

- ▶ identification ;
- ▶ code d'initialisation ;
- ▶ déclaration des options ;
- ▶ exécution des options ;
- ▶ chargement de modules ;
- ▶ code principal.

STRUCTURE D'UN FICHIER `module.sty`

- ▶ **identification ;**
- ▶ code d'initialisation ;
- ▶ déclaration des options ;
- ▶ exécution des options ;
- ▶ chargement de modules ;
- ▶ code principal.

[Structure d'un fichier de style](#)[Partie d'identification](#)**Commandes pour s'identifier**Exemple avec **engrec**

Exemple minimaliste

[Code d'initialisation](#)[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

IDENTIFICATION

```
\ProvidesPackage{<nom>}{<informations>}
```

Convention : [*<informations>*] = [2008/05/15 v1 Style personnel...]

Structure d'un fichier de style

Partie d'identification

Commandes pour s'identifier

Exemple avec **engrec**

Exemple minimaliste

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

IDENTIFICATION

```
\ProvidesPackage{<nom>}{<informations>}
```

Convention : [*<informations>*] = [2008/05/15 v1 Style personnel...]

```
\NeedsTeXFormat{<format>}{<date de parution>}
```

```
\NeedsTeXFormat{LaTeX2e}[2001/06/01]
```

[Structure d'un fichier de style](#)[Partie d'identification](#)[Commandes pour s'identifier](#)[Exemple avec `engrec`](#)[Exemple minimaliste](#)[Code d'initialisation](#)[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

IDENTIFICATION

```
\ProvidesPackage{<nom>}{<informations>}
```

Convention : [*<informations>*] = [2008/05/15 v1 Style personnel...]

```
\NeedsTeXFormat{<format>}{<date de parution>}
```

```
\NeedsTeXFormat{LaTeX2e}[2001/06/01]
```

Ces commandes sont optionnelles mais elles facilitent la maintenance des fichiers et donc des distributions.

EXEMPLE D'IDENTIFICATION

LE MODULE **engrec**

```
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\def\PackageName{engrec}
\def\fileversion{v1.1}
\def\filedate{2008/05/07}
\def\fileinfo{fichier engrec.sty
               par Le TeXnicien de surface}
\ProvidesPackage{engrec}
  [\filedate\space\fileversion\space\fileinfo]
```

Structure d'un fichier de style

Partie d'identification

Commandes pour s'identifier

Exemple avec **engrec**

Exemple minimaliste

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

EXEMPLE MINIMALISTE

Voici tout le contenu de `essai.sty` :

```
\def\@truc#1*#2\@nil{#2 --- #1}
\newcommand\LeTruc [1]{\@truc#1\@nil}
```

Voici un exemple d'utilisation :

```
\documentclass{minimal}
\usepackage{essai}
\usepackage{lmodern}
\begin{document}
\LeTruc{A*Bricole}
\end{document}
```

Ça marche!

Dans le code chargé par `\usepackage`, `@` est une lettre .

STRUCTURE D'UN FICHER `module.sty`

- identification ;
- **code d'initialisation ;**
- déclaration des options ;
- exécution des options ;
- chargement de modules ;
- code principal.

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)**Ce que l'on y peut mettre**Exemple avec `engpron`Exemple avec `breakurl`[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

INITIALISATION

Tout code accepté!

On peut même charger des modules avec `\RequirePackage`.

Modules utiles :

- ▶ `calc`;
- ▶ `ifthen`;
- ▶ `(x)keyval`.

EXEMPLE

LE MODULE `engpron`

```
\RequirePackage{tipa}[1996/06/10]
\RequirePackage{ifthen}
\newboolean{tip@style}
\newboolean{b@wild}
\newboolean{b@jones}
\newboolean{b@hyphenable}
\newboolean{b@visible}
\newboolean{b@final}
\newboolean{b@nice}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Ce que l'on y peut mettre

Exemple avec `engpron`

Exemple avec `breakurl`

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Ce que l'on y peut mettre

Exemple avec [engpron](#)Exemple avec [breakurl](#)

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

EXEMPLE

LE MODULE `breakurl`

```

\@ifpackageloaded{hyperref}{}{
  \PackageError{breakurl}{The breakurl depends on
    hyperref package} {I can't do anything. Please
    type X <return>, edit the source file^~J%
    and add
    \string\usepackage\string{hyperref\string}
    before
    \string\usepackage\string{breakurl\string}.}
  \endinput
}

\RequirePackage{xkeyval}

\newif\if@preserveurlmacro\@preserveurlmacrofalse
\newif\if@burl@fitstrut\@burl@fitstrutfalse
\newif\if@burl@fitglobal\@burl@fitglobalfalse

\newtoks\burl@toks
...

```

STRUCTURE D'UN FICHIER `module.sty`

- identification ;
- code d'initialisation ;
- **déclaration des options ;**
- exécution des options ;
- chargement de modules ;
- code principal.

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

```
Commande  
\DeclareOption  
Commande  
\PassOptionsToPackage  
Commande  
\DeclareOption*
```

Exécution des options

Chargement de modules

Code principal

Colophon

DÉCLARER DES OPTIONS

```
\DeclareOption{option}{code}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

DÉCLARER DES OPTIONS

```
\DeclareOption{<option>}{<code>}
```

Dans `engpron.sty`

```
\DeclareOption{tipastress}  
  {\setboolean{tip@style}{true}}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande

`\DeclareOption`

Commande

`\PassOptionsToPackage`

Commande

`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

DÉCLARER DES OPTIONS

`\DeclareOption{<option>}{<code>}`

Dans `engpron.sty`

```
\DeclareOption{tipastress}
  {\setboolean{tip@style}{true}}
```

Dans `article.cls`

```
\DeclareOption{a4paper}
  {\setlength\paperheight {297mm}%
   \setlength\paperwidth  {210mm}}
```

PASSER DES OPTIONS

On peut passer des options à d'autres modules qui seront chargés plus tard avec `\PassOptionsToPackage{<liste d'options>}{<nom du module>}`.

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)[Déclarations des options](#)[Commande
`\DeclareOption`](#)[Commande
`\PassOptionsToPackage`](#)[Commande
`\DeclareOption*`](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

PASSER DES OPTIONS

On peut passer des options à d'autres modules qui seront chargés plus tard avec `\PassOptionsToPackage{<liste d'options>}{<nom du module>}`.

Dans `amsmath.sty`

```
\DeclareOption{namelimits}{%  
  \PassOptionsToPackage{namelimits}{amsopn}}  
\DeclareOption{nonamelimits}{%  
  \PassOptionsToPackage{nonamelimits}{amsopn}}
```

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)[Déclarations des options](#)[Commande
`\DeclareOption`](#)[Commande
`\PassOptionsToPackage`](#)[Commande
`\DeclareOption*`](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

PASSER DES OPTIONS

On peut passer des options à d'autres modules qui seront chargés plus tard avec `\PassOptionsToPackage{<liste d'options>}{<nom du module>}`.

Dans `amsmath.sty`

```
\DeclareOption{namelimits}{%  
  \PassOptionsToPackage{namelimits}{amsopn}}  
\DeclareOption{nonamelimits}{%  
  \PassOptionsToPackage{nonamelimits}{amsopn}}
```

L^AT_EX ne vérifie pas que les modules sont effectivement chargés plus tard.

OPTIONS INCONNUES

Si un module est chargé avec une option inconnue, il émettra un avertissement. On peut cependant modifier ce comportement avec `\DeclareOptions*{<code>}`.

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

OPTIONS INCONNUES

Si un module est chargé avec une option inconnue, il émettra un avertissement. On peut cependant modifier ce comportement avec `\DeclareOptions*{<code>}`.

Dans `inputenc.sty`

```
\DeclareOption*{\inputencoding{\CurrentOption}}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

OPTIONS INCONNUES

Si un module est chargé avec une option inconnue, il émettra un avertissement. On peut cependant modifier ce comportement avec `\DeclareOptions*{<code>}`.

Dans `inputenc.sty`

```
\DeclareOption*{\inputencoding{\CurrentOption}}
```

Noter l'utilisation de `\CurrentOption`.

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

OPTIONS INCONNUES

Si un module est chargé avec une option inconnue, il émettra un avertissement. On peut cependant modifier ce comportement avec `\DeclareOptions*{code}`.

Dans `inputenc.sty`

```
\DeclareOption*{\inputencoding{\CurrentOption}}
```

Noter l'utilisation de `\CurrentOption`.

Dans `youngtab.sty`

```
\DeclareOption*{\PackageWarning{youngtab}{%
  Unknown option ‘\CurrentOption’ (Known:
  \MessageBreak ‘vcentermath’,
  ‘noautoscale’, ‘stdtext’, ‘enableskew’.)}}
```

Noter l'utilisation de `\MessageBreak`.

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

INFORMER, AVERTIR, SEMONCER

- ▶ `\PackageInfo{<nom>}{<texte informatif>};`

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande

`\DeclareOption`

Commande

`\PassOptionsToPackage`

Commande

`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

INFORMER, AVERTIR, SEMONCER

- ▶ `\PackageInfo{<nom>}{<texte informatif>};`
- ▶ `\PackageWarning{<nom>}{<avertissement>};`

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande

`\DeclareOption`

Commande

`\PassOptionsToPackage`

Commande

`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

INFORMER, AVERTIR, SEMONCER

- ▶ `\PackageInfo{<nom>}{<texte informatif>};`
- ▶ `\PackageWarning{<nom>}{<avertissement>};`
- ▶ `\PackageWarningNoLine{<nom>}{<avertissement>};`

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

INFORMER, AVERTIR, SEMONCER

- ▶ `\PackageInfo{<nom>}{<texte informatif>};`
- ▶ `\PackageWarning{<nom>}{<avertissement>};`
- ▶ `\PackageWarningNoLine{<nom>}{<avertissement>};`
- ▶ `\PackageError{<nom>}{<texte court>}{<texte long>};`

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Commande
`\DeclareOption`

Commande
`\PassOptionsToPackage`

Commande
`\DeclareOption*`

Exécution des options

Chargement de modules

Code principal

Colophon

STRUCTURE D'UN FICHER `module.sty`

- ▶ identification ;
- ▶ code d'initialisation ;
- ▶ déclaration des options ;
- ▶ **exécution des options ;**
- ▶ chargement de modules ;
- ▶ code principal.

EXÉCUTION DES OPTIONS

Après que toutes les options ont été définies avec `\DeclareOption`, on fait exécuter une liste d'options, les options « par défaut » :

```
\ExecuteOptions{<liste d'options>}
```

Dans `book.cls`

```
\ExecuteOptions{letterpaper,10pt,%  
twoside,onecolumn,final,openright}
```

On veillera à ne pas laisser d'espace derrière la virgule!

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

TRAITEMENT DES OPTIONS

On termine la partie concernant les options par :

```
\ProcessOptions\relax
```

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

TRAITEMENT DES OPTIONS

On termine la partie concernant les options par :

```
\ProcessOptions\relax
```

ou

```
\ProcessOptions*
```

TRAITEMENT DES OPTIONS

On termine la partie concernant les options par :

```
\ProcessOptions\relax
```

ou

```
\ProcessOptions*
```

Dans ce 2^e cas, les options passées par `\usepackage` ou `\RequirePackage` sont traitées dans l'ordre.

Un exemple : `babel`.

STRUCTURE D'UN FICHER `module.sty`

- identification ;
- code d'initialisation ;
- déclaration des options ;
- exécution des options ;
- **chargement de modules ;**
- code principal.

CHARGER DES MODULES

On peut désormais charger des modules avec

```
\RequirePackage[<liste d'options>]{<nom du module>}[<date de parution>].
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Colophon

CHARGER DES MODULES

On peut désormais charger des modules avec

```
\RequirePackage[<liste d'options>]{<nom du module>}[<date de parution>].
```

On pensera à charger les modules auxquels on a passé des options ci-dessus.

CHARGER DES MODULES

On peut désormais charger des modules avec

`\RequirePackage[liste d'options]{nom du module}[date de parution]`.

On pensera à charger les modules auxquels on a passé des options ci-dessus.

On dispose aussi de `\RequirePackageWithOptions{nom du module}[date de parution]` : les options passées au module sont celles de la classe ou du module appelant.

Dans le module `combnat.sty` de `combine` :

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{combnat}[2002/08/24 v0.2
    combined natbib package]
\@ifclassloaded{combine}{}{%
  \PackageError{combnat}{The 'combine' class
    is expected}{\@ehc}}
\RequirePackageWithOptions{natbib}
```

STRUCTURE D'UN FICHER `module.sty`

- ▶ identification ;
- ▶ code d'initialisation ;
- ▶ déclaration des options ;
- ▶ exécution des options ;
- ▶ chargement de modules ;
- ▶ **code principal.**

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

```
\CheckCommand  
\AtEndOfPackage  
\AtBeginDocument  
\AtEndDocument  
\InputIfFileExists  
\IfFileExists  
\IfFileExists=2
```

Colophon

CODE PRINCIPAL

C'est la partie principale, non ?

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Commandes spéciales](#)`\CheckCommand``\AtEndOfPackage``\AtBeginDocument``\AtEndDocument``\InputIfFileExists``\IfFileExists``\IfFileExists-2`[Colophon](#)

CODE PRINCIPAL

C'est la partie principale, non ?

Quelques commandes spéciales :

- ▶ `\AtEndOfPackage{code};`

CODE PRINCIPAL

C'est la partie principale, non ?

Quelques commandes spéciales :

- ▶ `\AtEndOfPackage{<code>};`
- ▶ `\AtBeginDocument{<code>}` : le `code` est exécuté dans le *préambule*;

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

CODE PRINCIPAL

C'est la partie principale, non ?

Quelques commandes spéciales :

- ▶ `\AtEndOfPackage{<code>};`
- ▶ `\AtBeginDocument{<code>}` : le `code` est exécuté dans le *préambule*;
- ▶ `\AtEndDocument{<code>};`

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

CODE PRINCIPAL

C'est la partie principale, non ?

Quelques commandes spéciales :

- ▶ `\AtEndOfPackage{<code>};`
- ▶ `\AtBeginDocument{<code>}` : le `code` est exécuté dans le *préambule*;
- ▶ `\AtEndDocument{<code>};`
- ▶ `\CheckCommand` qui prend les mêmes arguments que `\newcommand` : si la commande existe déjà, rien ne se passe, sinon on a un avertissement ;

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

CODE PRINCIPAL

C'est la partie principale, non ?

Quelques commandes spéciales :

- ▶ `\AtEndOfPackage{<code>;}`;
- ▶ `\AtBeginDocument{<code>}` : le `code` est exécuté dans le *préambule*;
- ▶ `\AtEndDocument{<code>;}`;
- ▶ `\CheckCommand` qui prend les mêmes arguments que `\newcommand` : si la commande existe déjà, rien ne se passe, sinon on a un avertissement ;
- ▶ `\@ifdefinable\NomMacro{<code>}` : le code est exécuté si `\NomMacro` n'est pas définie ou égale à `\relax` et si son nom ne commence pas par `end`.

CODE PRINCIPAL

C'est la partie principale, non ?

Quelques commandes spéciales :

- ▶ `\AtEndOfPackage{<code>};`
- ▶ `\AtBeginDocument{<code>}` : le `code` est exécuté dans le *préambule*;
- ▶ `\AtEndDocument{<code>};`
- ▶ `\CheckCommand` qui prend les mêmes arguments que `\newcommand` : si la commande existe déjà, rien ne se passe, sinon on a un avertissement ;
- ▶ `\@ifdefinable\NomMacro{<code>}` : le code est exécuté si `\NomMacro` n'est pas définie ou égale à `\relax` et si son nom ne commence pas par `end`.

On pourra lire `ltxdefs.dtx` à propos des commandes de définition de commandes.

INCLUSION DE FICHIERS

On peut contrôler l'inclusion de fichiers : fichiers de configuration, fichiers auxiliaires...

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Commandes spéciales](#)[\CheckCommand](#)[\AtEndOfPackage](#)[\AtBeginDocument](#)[\AtEndDocument](#)[\InputIfFileExists](#)[\IfFileExists](#)[\IfFileExists-2](#)[Colophon](#)

INCLUSION DE FICHIERS

On peut contrôler l'inclusion de fichiers : fichiers de configuration, fichiers auxiliaires...

- ▶ `\IfFileExists{<NomFichier>}{<code si-oui>}{<code sinon>}`

INCLUSION DE FICHIERS

On peut contrôler l'inclusion de fichiers : fichiers de configuration, fichiers auxiliaires...

- ▶ `\IfFileExists{<NomFichier>}{<code si-oui>}{<code sinon>}`
- ▶ `\InputIfFileExists{<NomFichier>}{<code si-oui>}{<code sinon>}` : si le fichier existe, il est chargé après que le `code si-oui` a été exécuté.

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

EXEMPLE D'UTILISATION DE `\CheckCommand`

Dans `pdfcolmk.sty` par Heiko OBERDIEK :

```
\CheckCommand{\@leftmark}[2]{#1}
\CheckCommand{\@rightmark}[2]{#2}
\CheckCommand*{\leftmark}{%
  \expandafter\@leftmark\botmark\@empty\@empty
}
\CheckCommand*{\rightmark}{%
  \expandafter\@rightmark\firstmark\@empty\@empty
}
\long\def\@leftmark#1#2#3{#2}
\long\def\@rightmark#1#2#3{#3}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

EXEMPLE D'UTILISATION DE `\AtEndOfPackage`

Dans `csquotes.sty` :

```
\AtEndOfPackage{%  
  \def\do#1#2{\catcode'#1=#2\relax}  
  \csq@docatcodes  
  \let\csq@docatcodes \@undefined  
  \let\do\noexpand}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

EXEMPLE D'UTILISATION DE `\AtBeginDocument`

Dans `beton.sty`, dernière ligne :

```
\AtBeginDocument{\normalsize}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

EXEMPLE D'UTILISATION DE `\AtEndDocument`

Tout le code de `lastpage.sty` :

```

\def\filename{lastpage}
\def\fileversion{v0.1b}
\def\filedate{1994/06/25}
\def\docdate {1994/07/20}
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{lastpage}[\filedate\space\fileversion\space
    LaTeX2e package for refs to last page number (JPG)]
\def\lastpage@putlabel{\addtocounter{page}{-1}%
    \immediate\write\@auxout{\string
    \newlabel{LastPage}{\the\thepage}}%
    \addtocounter{page}{1}}
\AtEndDocument{%
    \message{AED: lastpage setting LastPage}%
    \clearpage\lastpage@putlabel}%
\endinput

```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

EXEMPLE D'UTILISATION DE `\InputIfFileExists`

Dans `euro.sty`

```
\InputIfFileExists{euro.cfg}%  
  {\PackageInfo{euro}{Local config file %  
    euro.cfg used}}{}
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

EXEMPLE D'UTILISATION DE `\IfFileExists`

Dans `minitoc.sty`, le premier de beaucoup d'autres :

```
\if@mtcpatchmemoir@
\IfFileExists{mtcpatchmem.sty}{%
  \@ifclasslater{memoir}{2005/09/25}{}{%
    \RequirePackage{mtcpatchmem}}}{%
\mtePackageError[<E0028>]{minitoc}%
  {Unable to patch the memoir class}%
  {So it remains incompatible. Sorry.}}
\fi
```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

EXEMPLE D'UTILISATION DE `\IfFileExists-2`

Dans `pagenote.sty` :

```

\newcommand*{\printnotes}{%
  \@ifstar{\@sprintnotes}{\@printnotes}}
\newcommand*{\pnofilewarn}{%
  \PackageWarning{pagenote}{There is no .ent file}}
\newcommand*{\@sprintnotes}{%
  \ifmakingpagenotes
    \notedivision
    \IfFileExists{\jobname.ent}{%
      \immediate\closeout\@notefile
      \input{\jobname.ent}%
      \immediate\openout\@notefile=\jobname.ent%
    }{\pnofilewarn}
  \fi}

\newcommand*{\@printnotes}{%
  \ifmakingpagenotes
    \notedivision
    \IfFileExists{\jobname.ent}{%
      \immediate\closeout\@notefile
      \input{\jobname.ent}%
    }{\pnofilewarn}
  \fi}

```

Structure d'un fichier de style

Partie d'identification

Code d'initialisation

Déclarations des options

Exécution des options

Chargement de modules

Code principal

Commandes spéciales

`\CheckCommand`

`\AtEndOfPackage`

`\AtBeginDocument`

`\AtEndDocument`

`\InputIfFileExists`

`\IfFileExists`

`\IfFileExists-2`

Colophon

[Structure d'un fichier de style](#)[Partie d'identification](#)[Code d'initialisation](#)[Déclarations des options](#)[Exécution des options](#)[Chargement de modules](#)[Code principal](#)[Colophon](#)

Cette présentation est réalisée avec
`beamer` et `[expert,oldstyle]fourier`.

2008-05-15 — 2008-05-19
version 1.1 corrigée 2008-07-03