

# Sesión XI

## LaTeX avanzado II

- Creando class y package
- Control de flujo
- datatools
- Frágiles y robustos
- Entrega de ejercicios

# Sección 1

## Creando class y package

## Mi propia documentclass

Para crear una documentclass basta crear un archivo llamado <nombre>.cls e indicar que es una clase. Por ejemplo

### Código (otroarticulo.cls)

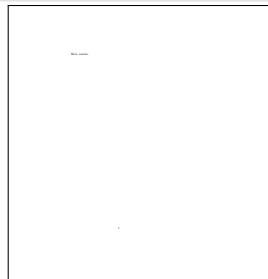
```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{otroarticulo} [2024/01/01 Articulo muy español]

\LoadClass{article}
```

### Código (main.tex)

```
\documentclass{otroarticulo}

\begin{document}
  Hola, mundo.
\end{document}
```



main.pdf

Los archivos `.cls` suelen ser bastante complejos.

Por ejemplo, puede encontrarse `article.cls` escribiendo en terminal:

```
kpsewhich article.cls
```

# ¿Qué puedo hacer en una class?

En un `.cls` podemos:

- Basarnos en clases existentes usando `\LoadClass`
- Incluir paquetes usando `\RequirePackage`
- Crear opciones para el paquete con `\DeclareOption`

Muchos ejemplos en

[https://www.overleaf.com/learn/latex/Writing\\_your\\_own\\_class](https://www.overleaf.com/learn/latex/Writing_your_own_class)

# Creando un package

Los paquetes se guardan en archivos `<nombre>.sty`.

## Código (`examplepackage.sty`)

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesPackage{examplepackage}[2014/08/24 Example LaTeX  
↪ package]
```

Leer más: [https:](https://www.overleaf.com/learn/latex/Writing_your_own_package)

[//www.overleaf.com/learn/latex/Writing\\_your\\_own\\_package](https://www.overleaf.com/learn/latex/Writing_your_own_package)

## Sección 2

# Control de flujo



# Condicionales nativos

<https://en.wikibooks.org/wiki/TeX/if>

*The `\if` command denotes the start of an if-then-else control structure. The forms `<token-1>` and `<token-2>` must expand to tokens. `<token-1>` and `<token-2>` can be either a character or a control sequence. If `<token-1>` and `<token-2>` both expand to the same character code then `<tex-code-1>` is processed; otherwise it is ignored. If the `\else` section is included and `<token-1>` and `<token-2>` expand to different character codes, then `<tex-code-2>` is processed; otherwise it is ignored.*

## Código

```
\documentclass[varwidth]
  {standalone}

\begin{document}

\def \mycmd {1}
\if\mycmd1
  Es 1
\else
  No es 1, es \mycmd
\fi

\def \mycmd {2}
\if\mycmd1
  Es 1
\else
  No es 1, es \mycmd
\fi

\end{document}
```

Es 1

No es 1, es 2

## `\ifx`

`\if` “expande” (es decir, evalúa) los tokens antes de compararlos. Detalles sobre expansión en este link.

Un comando similar es `\ifx`. La diferencia es complicada: StackOverflow.

Un ejemplo habitual de uso es

```
\ifx\mycmd\undefined
```

## `\ifthenelse`

El paquete `ifthen` (mantenido por el equipo de  $\text{\LaTeX}$ ) nos permite un `if` más “moderno”:

```
\ifthenelse{<expression>}{<>true code>}{<>false code>}
```

Las condiciones utilizan

<code>==</code>	equality		<code>!=</code>	inequality
<code>&lt;</code>	less than		<code>&gt;</code>	greater than
<code>&amp;&amp;</code>	AND		<code>  </code>	OR
<code>!</code>	NOT			

Podemos utilizar contadores

```
\newcounter{value}
\setcounter{value}{10}

\ifthenelse{\value{value}>5}
{\color{green}Large value}
{\color{red}Small value}
```

E incluso se introduce valores `bool`

```
\newboolean{draft}
\setboolean{draft}{True}
\ifthenelse{\boolean{draft}}
{Draft version}
{}
```

Fuente de los ejemplos

Ver también <https://ctan.org/pkg/xifthen>

# Bucles

El comando `\foreach` nos permite hacer bucles for

## Código

```
\foreach\x in  
{1,2,3,0}{\x,}
```

1,2,3,0,

Admite notaciones de rango como `\{1,2,\dots,6\}`

El paquete `forloop` añade el comando

```
\forloop{counter}{startValue}{condition}{body}
```

Ver también el paquete `multido`.

## Sección 3

`datatools`

# datatool

Vamos a utilizar los siguientes datos:

## Archivo datos.csv

```
título, precio  
El Quijote, 100  
El Lazarillo de Tormes, 50
```

## Código

```
\documentclass{standalone}  
  
\usepackage{datatool}  
  
\begin{document}  
\DTLloaddb{datos}{datos.csv}  
\DTLdisplaydb{datos}  
\end{document}
```

título	precio
El Quijote	100
El Lazarillo de Tormes	50

# datatool

## Código

```
\documentclass{standalone}

\usepackage{datatool}
\usepackage{eurosym}

\begin{document}

\DTLloaddb{datos}{datos.csv}

\DTLforeach*
{datos} % database
{\Title=título,\Price=precio}
→ % assignments
{\Title{ } (\Price \euro), } %
→ body
\end{document}
```

El Quijote (100€), El Lazarillo de Tormes (50€),



# datatool

## Código

```
\documentclass{standalone}

\usepackage{datatool}
\usepackage{eurosym}

\begin{document}

\DTLloaddb{datos}{datos.csv}

\DTLforeach*
[\equal{\Price}{100}]% condition
{datos} % database
{\Title=título,\Price=precio} %
→ assignments
{\Title{} (\Price \euro), } %
→ body

\end{document}
```

El Quijote (100€),

## Sección 4

# Frágiles y robustos

# Comandos frágiles y robustos, expansión y ejecución

StackOverflow

## La documentación de beamer

*fragile* tells beamer that the frame contents is “fragile.” This means that the frame contains text that is not “interpreted as usual.” For example, this applies to verbatim text, which is, obviously, interpreted somewhat differently from normal text. If a frame contains fragile text, different internal mechanisms are used to typeset the frame to ensure that inside the frame the character codes can be reset. The price of switching to another internal mechanism is that either you cannot use overlays or an external file needs to be written and read back (which is not always desirable).

...

If you wish to use a `{verbatim}` environment in a frame, you have to add the option `[fragile]` to the `{frame}` environment. The `\end{frame}` must be alone on a single line (except for any leading whitespace). Using this option will cause the frame contents to be written to an external file and then read back. See the description of the `{frame}` environment for more details. You must also use the `[fragile]` option for frames that include any “fragile” text, which is any text that is not “interpreted the way text is usually interpreted by TEX.” For example, if you use a package that (locally) redefined the meaning of, say, the character `&`, you must use this option.

## Sección 5

### Entrega de ejercicios

## Ejercicio a entregar (opcional)

Entregar un documento `semana11.tex` que resuelva alguno o todos los ejercicios en

<https://web.mit.edu/jgross/www/LaTeX/Really%20Hard%20Exercises%20Become%20a%20LaTeXer.pdf>