

Sesión V

Listas y tablas

① Listas

② Tablas

③ Entrega de ejercicios

Sección 1

Listas

Listas básicas

Código

```
\documentclass{article}

\begin{document}
  \begin{enumerate}
    \item Primero
    \item Segundo
  \end{enumerate}
  \begin{itemize}
    \item Primero
    \item Segundo
  \end{itemize}
\end{document}
```

1. Primero
 2. Segundo
- Primero
 - Segundo

Cambiando el comportamiento por defecto

El comportamiento por defecto es
`enumerate` es números, y para `itemize` es ●.

El comportamiento de `itemize` se puede modificar

```
\renewcommand{\labelitemi}{\dot{}}}
```

Además, `\item` admite como entrada optativa el texto que queremos poner, p.e. `\item[!]`.

Para configurar `enumerate` es preferible usar uno (y sólo uno) de los paquetes:

`enumitem` (más completo, y recomendado) o `enumerate` (más sencillo).

El paquete enumerate

La documentación en CTAN (4 páginas) es muy clara:

This package gives the enumerate environment an optional argument which determines the style in which the counter is printed. An occurrence of one of the tokens A a l i 1 produces the value of the counter printed with (respectively) `\Alph` `\alph` `\Roman` `\roman` or `\arabic`.

These letters may be surrounded by any strings involving any other TEX expressions, however the tokens A a l i 1 must be inside a `{ }` group if they are not to be taken as special.

Código

```
\documentclass{article}
\usepackage{enumerate}
\begin{document}
  \begin{enumerate}[EX i.]
    \item \label{LA}
      one one
    \item two
      \begin{enumerate}
        [{example} a]
          \item \label{LB}
            one of two
          \item two of two
        \end{enumerate}
    \end{enumerate}
    \begin{enumerate} [{A}-1]
      \item \label{LC}
        one
      \item two
    \end{enumerate}
\end{document}
```

```
EX i. one one
EX ii. two
      example a) one of two
      example b) two of two
A-1 one
A-2 two
```

1

Introducción a contadores

En la documentación anterior habla de `\Alph` `\alph` `\Roman` `\roman` or `\arabic`.

Esto son términos de contadores que veremos más adelante. Baste, por ahora, un pequeño ejemplo:

Código

```
\newcounter{micontador}
\setcounter{micontador}{2}

\arabic{micontador},
\Roman{micontador},
\roman{micontador},
\Alph{micontador},
\alph{micontador}.
```

2, II, II, B, b.

El paquete enumitem

La documentación de este paquete en CTAN tiene 23 páginas.

Tiene varias funcionalidades muy completas:

- 1 Control de márgenes y espaciado
- 2 Control de etiquetas muy detallada (aunque algo farragoso). Por ejemplo
`\begin{enumerate}[label=\emph{\alph*}), ref=\emph{\alpha*}]`
- 3 Permite continuar un la numeración de un enumerate anterior, con la sintaxis
`\begin{enumerate}[resume].`
- 4 Permite crear nuevos tipos de listas, con la sintaxis:
`\newlist{<name>}{<type>}{<max-depth>}`

Sección 2

Tablas

Tablas

Las el entorno `table` es equivalente a `figure`, pero al generar el `caption` obtendremos Cuadro (como recomienda la RAE).

En contenido de la tabla se introduce de manera similar a una matriz

- 1 `<align>`: los básicos son
 - a) l: izquierda
 - b) c: centrado
 - c) r: derecha
 - d) `p{<width>}`: justificado a un ancho dado

Código

```
\begin{tabular}{<align>| ... }  
    cuadro1 & cuadro2 & ... \\  
    \hline\\  
    ...  
\end{tabular}
```

Tablas

Las el entorno `table` es equivalente a `figure`, pero al generar el `caption` obtendremos Cuadro (como recomienda la RAE).

En contenido de la tabla se introduce de manera similar a una matriz

- 1 `<align>`:
- 2 `&` Separación entre cuadros en la misma fila

Código

```
\begin{tabular}{<align>| ... }  
    cuadro1 & cuadro2 & ... \\  
    \hline\\  
    ...  
\end{tabular}
```

Tablas

Las el entorno `table` es equivalente a `figure`, pero al generar el `caption` obtendremos Cuadro (como recomienda la RAE).

En contenido de la tabla se introduce de manera similar a una matriz

- 1 `<align>`:
- 2 `&`
- 3 `|` Si se desea línea vertical entre dos columnas

Código

```
\begin{tabular}{<align>| ... }  
    cuadro1 & cuadro2 & ... \\  
    \hline\\  
    ...  
\end{tabular}
```

Tablas

Las el entorno `table` es equivalente a `figure`, pero al generar el `caption` obtendremos Cuadro (como recomienda la RAE).

En contenido de la tabla se introduce de manera similar a una matriz

- 1 `<align>`:
- 2 `&`
- 3 `|`
- 4 `\hline` Si se desea una línea horizontal.

Código

```
\begin{tabular}{<align>| ... }  
    cuadro1 & cuadro2 & ... \\  
    \hline\\  
    ...  
\end{tabular}
```

Ejemplo básico

Código

```
\begin{table}

\begin{tabular}{c|l r|p{1cm}}
1 & B & 3 & un texto largo\\
\hline
4 & 5 & & un texto largo \\
6 & & 7 & 
\end{tabular}

\caption{Mi tabla}
\end{table}
```

1	B	3	un texto largo
4	5		un texto largo
6		7	

Cuadro: Mi tabla

Más alineaciones

Hay paquete especiales que añaden funcionalidades

Código

% <https://tex.stackexchange.com/questions/35293/p-m-and-b-columns-in-tables>

```
\documentclass{standalone}

\usepackage{array}

\begin{document}
\begin{tabular}{| p{2cm} | m{2cm} | b{2cm} | }
\hline
\centering header p &
\centering header m &
\centering header b \tabularnewline
\hline
text which is considerably longer than ... & &
text which is considerably longer than ... & &
text which is considerably longer than ...
\hline
\hline
\end{tabular}

\end{document}
```

header p	header m	header b
text which is considerably longer than	text which is considerably longer than	text which is considerably longer than
...
...

Líneas horizontales parciales cline

Código

```
\begin{tabular}{llrr}  
  a & b & c & d \\  
  \cline{1-1} \cline{3-4}  
  e & f & g & h  
\end{tabular}
```

a	b	c	d
e	f	g	h

Combinando columnas multicolumn

Código

```
\documentclass{standalone}

\begin{document}
\begin{tabular}{1|1|1|1}
  A & \multicolumn{2}{c|}{B} & \\
  a & b1 & b2 & c
\end{tabular}
\end{document}
```

A	B		C
a	b1	b2	c

Combinando filas multirow

Esto requiere cargar el paquete multirow.

Código

```
\documentclass{standalone}

\usepackage{multirow}

\begin{document}
\begin{tabular}{|l|l|l|}
\multirow{2}{*}{A} & B & C \\
& b & c
\end{tabular}
\end{document}
```

A	B	C
	b	c

Usando multicolumn y multirow

numeric literals	integers	in decimal	8743
		in octal	0o7464
			00103
		in hexadecimal	0x5A0FF
	0xE0F2		
	fractionals	in decimal	140.58
			8.04e7
			0.347E+12
5.47E-12			
		47e22	
char literals			'H'
			'\n'
			'\x65'
string literals			"bom dia"
			"ouro preto\nmg"

Código

% <https://tex.stackexchange.com/questions/167366/combining-multirow-and-multicolumn>

```
\documentclass{standalone}
\usepackage{multirow}

\begin{document}

\begin{tabular}{|l|l|l|l|}\hline
\multirow{10}{*}{numeric literals} & \multirow{5}{*}{integers} & in decimal & \verb|8743| \\ \cline{3-4}
& & \multirow{2}{*}{in octal} & \verb|0o7464| \\ \cline{3-4}
& & \verb|00103| \\ \cline{3-4}
& & \multirow{2}{*}{in hexadecimal} & \verb|0x5A0FF| \\ \cline{3-4}
& & & \verb|0xE0F2| \\ \cline{2-4}
& \multirow{5}{*}{fractionals} & \multirow{5}{*}{in decimal} & \verb|140.58| \\ \cline{3-4}
& & & \verb|8.04e7| \\ \cline{3-4}
& & & \verb|0.347E+12| \\ \cline{3-4}
& & & \verb|5.47E-12| \\ \cline{3-4}
& & & \verb|47e22| \\ \cline{1-4}
\multicolumn{3}{|l|}{\multirow{3}{*}{char literals}} & \verb|'H'| \\ \cline{3-4}
\multicolumn{3}{|l|}{& \verb|'\n'|} \\ \cline{3-4}
\multicolumn{3}{|l|}{& \verb|'\x65'|} \\ \cline{1-4}
\multicolumn{3}{|l|}{\multirow{2}{*}{string literals}} & \verb|"bom dia"| \\ \cline{3-4}
\multicolumn{3}{|l|}{& \verb|"ouro preto\nmg"|} \\ \cline{1-4}
\end{tabular}

\end{document}
```

Algunos comentario más sobre tablas

Más info: <https://www.overleaf.com/learn/latex/Tables>

Hay múltiples páginas que permiten generar tablas con más comodidad

- <https://www.tablesgenerator.com/>
- <https://www.latex-tables.com/>
- <https://tableconvert.com/latex-generator>

Paquete alternativo: booktabs

El paquete `booktabs` ofrece una alternativa a las tablas convencionales, que algunos autores valoran positivamente:

`https:`

`//nhigham.com/2019/11/19/better-latex-tables-with-booktabs/`

Tablas en ecuaciones

El modo matemático incluye array.

Código

```
\[  
f(x) =  
\begin{array}{rl}  
  1 & \text{si } x > 0, \\  0 & \text{si } x \le 0.  
\end{array}  
\]
```

$$f(x) = \begin{array}{ll} 1 & \text{si } x > 0, \\ 0 & \text{si } x \leq 0. \end{array}$$

En general, para una función definida a trazos se recomienda utilizar el entorno `cases` (si no hay fracciones), o `dcases` del paquete `mathtools` (si hay fracciones).

Sección 3

Entrega de ejercicios

Ejercicios a entregar

Entregar en Moodle un documento `semana05.tex` tal que su correspondiente `.pdf` produzca con la mayor similitud posible:

Números en julia

El lenguaje de programación `julia` estructura tiene tipos “abstractos” y tipos “concretos”. Los tipos abstractos admiten tipos “hijo”, mientras que los concretos no. Los tipos concretos pueden ser de tipo “primitivo” o “compuesto”.

Los números se estructuran de la siguiente manera. Existe un tipo abstracto `Number` tiene la siguiente jerarquía de tipos, todos ellos abstractos.

```
A Complex
B Real
  I AbstractFloat
  II Integer
  III Rational
  IV AbstractIrrational
```

Los tipos hijos de `Integer` (item B.II) vienen descritos por el Cuadro 1.

		Número de bits	Menor valor	Mayor valor
Integer	Signed	Int8	-2^7	$2^7 - 1$
		Int16	-2^{15}	$2^{15} - 1$
		Int32	-2^{31}	$2^{31} - 1$
		Int64	-2^{63}	$2^{63} - 1$
		Int128	-2^{127}	$2^{127} - 1$
	Unsigned	UInt8	0	$2^8 - 1$
		UInt16	0	$2^{16} - 1$
		UInt32	0	$2^{32} - 1$
		UInt64	0	$2^{64} - 1$
		UInt128	0	$2^{128} - 1$
Bool	Bool	8	false(0)	true(1)

Cuadro 1: Tipos primitivos de números enteros