

Principios del modo matemático

Composición de textos científicos

18 de septiembre de 2020

1 Un vistazo más a la cabecera

La cabecera de un documento es lo que aparece antes de `\begin{document}`. En nuestros primeros ejemplos, tendremos algo tan breve como

```
\documentclass{article}
\usepackage[utf8]{inputenc}
```

(de hecho esto es lo que pone por defecto Overleaf). La primera línea indica que se toman los parámetros por defecto de un formato llamado `article`. Con otros editores se especifican algunos de estos parámetros entre corchetes. Por ejemplo

```
\documentclass[a4paper,10pt]{article}
```

que indica que el formato del papel es A4 y que las fuentes son de tamaño diez puntos. Podemos jugar a cambiar `10pt` por `11pt` o `12pt`, o `a4paper` por `letterpaper` (los folios usados en EE.UU.) y ver los resultados.

La segunda línea indica que la codificación es UTF8 con un parámetro que se pasa al paquete `inputenc`. Parámetros aparte, un paquete es algo que se carga con `\usepackage{...}` y que extiende de cierta forma las capacidades de \LaTeX . Hay algunos muy específicos y otros que hacen cosas muy tontas. Por ejemplo, incluyendo `\usepackage{lipsum}` en la cabecera dispondremos de la instrucción `\lipsum` que genera el típico texto *Lorem ipsum* usado en pruebas de imprenta, con las variantes `\lipsum[n-m]` que dan los párrafos entre `n` y `m`. Por ejemplo `\lipsum[11-11]`, que se puede abreviar con `\lipsum[11]`, produce:

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

En matemáticas casi todo el mundo carga tres paquetes de la AMS (*American Mathematical Society*) incluso si no los usa y así lo haremos nosotros. Nuestras cabeceras incluirán

```
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
```

En la práctica, casi ningún matemático tenemos mucha idea de qué comandos dejarían de funcionar si no los cargásemos y la realidad es que en textos simples no necesitaríamos ninguno. Es más una costumbre y una precaución. Un físico o un químico seguro que hará lo mismo con otros paquetes.

2 Dos tipos de fórmulas

Las fórmulas en un texto matemático pueden aparecer dentro de una línea *in-line* o resaltadas y centradas *displayed*. Las primeras en \LaTeX se encierran entre $\$$ y las segundas entre $\[$ y $\]$. Por ejemplo la fórmula $x + y = 2$ se ha escrito como $\$x+y=2\$$ mientras que $\[x+y=2 \]$ (que se suele escribir en tres líneas) daría lugar a

$$x + y = 2$$

Los espacios en las fórmulas matemáticas son en general irrelevantes, así que $\[x +y = 2 \]$ daría el mismo resultado. Cabe preguntarse en el primer caso por qué no tecleamos la fórmula como texto, sin $\$$, el resultado sería entonces $x+y=2$ que es más feo y posiblemente incoherente con el tipo de letra de otras fórmulas que involucren x e y .

Para introducir un subíndice en una fórmula usaremos $_$ y para un superíndice $\^$. Cuando estos tengan más de un carácter o comando, los agruparemos con llaves. Por ejemplo, $E = mc^2$ es $\$E=mc^2\$$ y $E = mc^{1+1}$ es $\$E=mc^{1+1}\$$. Esto lleva a la duda de cómo teclear una llave si queremos que aparezca de verdad, basta poner delante \backslash . Así tenemos

$$\{x_n\} \leftrightarrow \backslash\{x_n\} \quad \text{y} \quad \{x_n\}_{n=1}^{\infty} \leftrightarrow \backslash\{x_n\}_{n=1}^{\infty}$$

Además hemos aprendido que el comando ∞ da lugar a ∞ .

La regla general es que en \LaTeX los símbolos matemáticos se reemplazan por sus nombres en inglés o por abreviaturas suyas precedidos de la barra hacia atrás \backslash . Un ejemplo son las letras griegas en las que se usan el nombre completo (para las mayúsculas se pone la primera letra en mayúsculas). Así

```
\[
\alpha^{\beta},
\quad
\Omega^{\lambda^2},
\quad
```

`\theta(\pi)`
`\]`

da lugar a

$$\alpha^\beta, \quad \Omega^{\lambda^2}, \quad \theta(\pi)$$

y además hemos aprendido que `\qqquad` introduce cierta separación entre fórmulas.

Las funciones trigonométricas básicas son `\sin`, `\cos`, `\tan`, etc. Si utilizamos el paquete de idioma `babel` con la opción `spanish` tendremos la traducción¹ `\sen`. Otras funciones son `\log`, `\exp`, `\min`, `\max`, `\inf`, `\sup` y seguro que puedes imaginar más ejemplos y comprobarlos en un manual.

De nuevo surge la pregunta de para qué necesitamos escribirlo de esta forma si podríamos teclearlo como texto. Seguramente los siguiente respuesta a la pregunta:

<code>\cos 0 + \tan(0+0)</code>	\rightarrow	<code>cos0 + tan(0 + 0)</code>	Mal
<code>cos 0 + tan(0+0)</code>	\rightarrow	<code>cos 0 + tan(0+0)</code>	Mal
<code>\cos 0 + \tan(0+0)</code>	\rightarrow	<code>cos 0 + tan(0 + 0)</code>	Bien

De acuerdo, en la del centro quizá habría que decir que está regular. Su problema es que, siendo un poco exigente, el espaciado no es el adecuado.

3 Fracciones y raíces

Hay algunos comandos `LATEX` que necesitan argumentos. Los más comunes son las fracciones. Su formato es `\frac{}{}` donde entre las primeras llaves se pone el numerador denominador y entre las segundas el denominador. Omitir las llaves solo funciona con números de un dígito. La mayor parte de los editores completan el comando para que sea más fácil de usar. `LATEX` toma decisiones “inteligentes” acerca del tamaño de la fracción. Por ejemplo, con `E=\frac{3}{4} mc^2` obtenemos $E = \frac{3}{4}mc^2$ en una fórmula *in-line* pero como *displayed* sería

$$E = \frac{3}{4}mc^2$$

(por cierto, aunque suene a broma esta fórmula apareció en la física del siglo pasado para la energía del electrón antes de la llegada de la relatividad).

También las fracciones de fracciones siguen normas sensatas acerca del tamaño:

$$\frac{x + \frac{1}{2}}{y} \leftrightarrow \frac{\frac{x + \frac{1}{2}}{y}}{1}$$

Más adelante veremos como cambiar esto a nuestro gusto.

¹En el curso veremos cómo definir `\sen` u otros comandos nosotros mismos.

Las raíces cuadradas responden a `\sqrt{...}` y se generalizan a raíces de índice n añadiendo un parámetro opcional como `\sqrt[n]{...}`. Por ejemplo

$$\sqrt{\frac{1+x}{1-x}} \leftrightarrow \sqrt{\frac{1+x}{1-x}}, \quad \sqrt[3]{1+2^4} \leftrightarrow \sqrt[3]{1+2^4}$$

Yo prefiero leer en un libro $(1+x)^{1/5}$ que $\sqrt[5]{1+x}$ o $(1+x)^{\frac{1}{5}}$ pero es solo una opinión.

4 Operadores con límites

Algunas de las funciones que hemos visto y en general algunos operadores (comandos) admiten límites. Como regla general los superiores se indican como si fueran superíndices y los inferiores como subíndices. Estos límites se colocan automáticamente dependiendo de si la fórmula es *in-line* o *displayed*. El primer ejemplo básico son los límites, en sentido matemático, que como cabía esperar se indican con `\lim` y también como cabía esperar `\to` indica el “tiende a”. Por ejemplo

$$\lim_{x \rightarrow 0^+} x^{\tan x} = 1$$

se obtiene con

```
\[
\lim_{x \to 0^+} x^{\tan x}=1
\]
```

y en una línea veríamos $\lim_{x \rightarrow 0^+} x^{\tan x} = 1$.

Algo similar se puede decir de máximo, mínimo, supremo e ínfimo. Por ejemplo, ¿sabrías dar una fuente para las siguientes fórmulas?

$$\inf_{x>0} \frac{\cos x}{1+x^2} \quad \text{y} \quad \max_{n>1} \{\exp(10n - n^2)\}.$$

Si no es así, consulta la fuente de esta guía.

Los operadores llamados “grandes” admiten también límites superiores. Los más utilizados son

`\sum`, `\prod`, `\bigcup` y `\bigcap`

que sin sus límites dan lugar a

$$\sum, \quad \prod, \quad \cup \quad \text{y} \quad \cap.$$

Por ejemplo, la famosa identidad del problema de Basilea y una de las leyes de De Morgan

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \text{y} \quad \bigcup_{j \in J} A_j^c = \left(\bigcap_{j \in J} A_j \right)^c$$

se obtendrían con el código

```

\sum_{n=1}^{\infty}
\frac{1}{n^2}
=
\frac{\pi^2}{6}

```

y

```

\bigcup_{j \in J} A_j^c
=
\big(
\bigcap_{j \in J} A_j
\big)^c

```

Más adelante veremos los diferentes delimitadores pero el último ejemplo ya muestra que `\big(` y `\big)` producen paréntesis un poco más grandes de lo normal.

Las integrales siguen la misma filosofía, así podríamos escribir la consecuencia del teorema fundamental del cálculo

$$\frac{d}{dx} \int_0^x f = f(x)$$

con

```

\frac{d}{dx}
\int_0^x f = f(x)

```

Hay un punto con las integrales sobre el que volveremos más adelante y es cómo espaciar el diferencial. Por ejemplo, si uno es un poco purista la fórmula

$$\int_0^1 x dx = \frac{1}{2}$$

que se obtendría con `\int_0^1 x dx = \frac{1}{2}` queda un poco fea porque x y dx están pegados. Casi todo el mundo se contenta con $x dx$ y algunos quieren ver $x dx$.

5 Operadores binarios y relaciones binarias

Aparte de cosas como `=`, `>` y alguna más que podemos teclear directamente con nuestro teclado los operadores que suelen relacionar dos términos de una expresión matemática son

\leq	<code>\le</code>	\geq	<code>\ge</code>	\cup	<code>\cup</code>	\cap	<code>\cap</code>
\in	<code>\in</code>	\equiv	<code>\equiv</code>	\subset	<code>\subset</code>	\supset	<code>\supset</code>
\times	<code>\times</code>	\cdot	<code>\cdot</code>	\vee	<code>\vee</code>	\wedge	<code>\wedge</code>
\sim	<code>\sim</code>	\simeq	<code>\simeq</code>	\approx	<code>\approx</code>	\circ	<code>\circ</code>
\pm	<code>\pm</code>	\setminus	<code>\setminus</code>	$ $	<code>\mid</code>	\perp	<code>\perp</code>

Muchos de ellos se pueden negar al precederlos de `\not`. Así $\not\approx$ se obtienen con `\not\approx`. Están las abreviaturas particulares `\notin`, `\neq` y `\nmid` para \notin , \neq y \nmid . La última es casi obligatoria porque `\not\mid` no da el resultado esperado.

Aunque la aplicación web `detexify` es más un juguete nos puede sacar de algún apuro si tenemos pulso y no queremos buscar en la documentación.

Los editores de \LaTeX suelen ofrecer una lista de los símbolos más comunes u otros más extraños. Incluso los usuarios más experimentados de vez dan vistazo a estas listas.