

Apuntes de L^AT_EX

Capítulo 17: Gestión de tipos

Abstract

En éste capítulo trataremos las posibilidades existentes para manejar tipos (ó caracteres) en L^AT_EX. Existe una amplia variedad de tipos a nuestra disposición, cada uno de ellos almacenado en un paquete específico. Los tipos por defecto son los llamados “Computer Modern Fonts”, creados por D. E. Knuth empleando el programa METAFONT. Este programa permite construir los tipos utilizando un lenguaje gráfico especial. A continuación describiremos cómo utilizar otras familias de tipos a nuestra disposición.

1 Fuentes postscript

Además de las fuentes Computer Modern creadas por Knuth, podemos utilizar otros muchos diferentes tipos. Existen dos procedimientos para el cambio de fuentes; si queremos cambiar el tipo de letra para TODO EL DOCUMENTO, se debe cargar uno de los siguientes paquetes de la tabla en el preámbulo (por ejemplo, para éste documento se ha usado el paquete mathptmx, que proporciona el tipo de letra times):

Paquete	Código	roman	sansserif	typewriter	formulas
mathpazo	ppl	Palatino			≈ Palatino
mathptmx	ptm	Times			≈ Times
helvet	phv		Helvetica		
avant	pag		Avant Garde		
courier	pcr			Courier	
bookman	pbk	Bookman			
newcent	pnc	New Century			
charter	bch	Charter			
chancery	pzc	<i>Zapf Chancery</i>			

Si por el contrario deseamos cambios puntuales de tipo de letra, debemos emplear los códigos de tres letras (ptm, phv, etc...) en combinación con los siguientes comandos de bajo nivel que especifican las propiedades de la fuente:

- `\fontencoding{Codificación}` Tipo de codificación de la fuente: OT1, T1, OT2... (no es necesario especificarla en general, si ya lo hemos hecho en el preámbulo)
- `\fontfamily{Familia}` La familia se especifica a través de los códigos de tres letras mencionados en la tabla, que identifican a la fuente
- `\fontseries{Serie}` Se pueden emplear los valores m (medio ó normal), b (negrita), bx (negrita extendida), sb (seminegrita) y c (condensada). Puede que no todos ellos estén disponibles, éso depende de la familia en particular
- `\fontshape{Perfil}` n (normal ó recto), it (itálico), sl (inclinado), sc (versalita)

- `\fontsize{Tamaño}{Interlínea}` Tamaño es una longitud rígida, mientras que Interlínea es otra longitud que puede admitir valores elásticos; Para un resultado correcto es aconsejable que la interlínea sea aproximadamente un 20% mayor que el tamaño del tipo de letra. Debe hacerse notar que, en ciertos casos (las fuentes Computer Modern, por ejemplo), se pueden haber fijado valores admisibles del tamaño, con lo que L^AT_EX sustituirá el valor elegido para Tamaño por el más cercano entre los admisibles. En tales situaciones, se debe optar por emplear los comandos `\resizebox` ó `\scalebox`
- `\selectfont` Tras haber configurado los parámetros anteriores (ó alguno de ellos), se debe declarar este comando para hacer efectivos los cambios. Si queremos que tales cambios sean locales (por ejemplo para cambiar el tipo de un determinado párrafo) deberían encerrarse todos los comandos y el texto al que afectan entre llaves (a fin de delimitar un grupo)

por ejemplo:

Ejemplo:

```
\newdimen\tamanyo
\newdimen\interlinea
\def\letra#1#2{%
\tamanyo=#1%
\interlinea=1.2\tamanyo%
\fontfamily{pbk}
\fontsize{\the\tamanyo}%
{\the\interlinea}\selectfont#2}
\letra{1pt}{Hola} \letra{5pt}{Hola}
\letra{10pt}{Hola}
\letra{20pt}{Hola} \\
\letra{1cm}{Hola} \\
\letra{2cm}{Hola}
```

..Hola Hola Hola

Hola

Hola

2 Los paquetes pifont y marvosym

Los paquetes pifont y marvosym proporcionan herramientas para manejar con más comodidad las fuentes Symbol, ZapfDingbats y MarvoSym. Por ejemplo, con el paquete pifont, podemos acceder a las fuentes Symbol y ZapfDingbats mediante el comando

```
\Pisymbol{Fuente}{Número}
```

donde Fuente admite como opciones psy ó pzd, y Número tiene el mismo significado que para el comando `\symbol`.

```
\Pifill{Fuente}{Número} \Piline{Fuente}{Número}
```

rellenan un espacio extensible, en el primer caso, y una línea entera, en el segundo, con el símbolo escogido.

Tenemos también entornos para construir listas análogas a `itemize` y `enumerate`, en las cuales cada ítem viene precedido de un símbolo obtenido de las fuentes Symbol ó ZapfDingbats:

```
\begin{Pilist}{Fuente}{Número} \item xxx ... \end{Pilist}
```

```
\begin{Piautolist}{Fuente}{Número} \item xxx ... \end{Piautolist}
```

donde para el primero, el símbolo se mantiene constante (como en `itemize`), y en el segundo, va cambiando a partir de un valor inicial descrito por Número

Como lo más frecuente es utilizar la fuente ZapfDingbats, existen versiones simplificadas de los anteriores comandos y entornos:

```
\ding{Número}          \dingfill{Número}      \dingline{Número}  
\begin{dinglist}{Número} ... \begin{dingautolist}{Número} ...
```

Ejemplo:

El paquete babel permite gestionar, entre otros, los idiomas:

```
\begin{dingautolist}'{266}  
\item Español  
\item Catalán  
\item Gallego  
\end{dingautolist}
```

El paquete babel permite gestionar, entre otros, los idiomas:

- ❶ Español
 - ❷ Catalán
 - ❸ Gallego
-

Finalmente, el paquete marvosym implementa un comando básico `\mvchrNúmero`, de significado análogo a `\char`. Además, proporciona una serie de comandos, en inglés, que nombran los diversos símbolos de la tabla de la fuente (más información en la documentación del paquete). Por ejemplo:

```
\Letter   ✉          \Mobilefone  📞          \Faxmachine  📠          etc...
```