Ortotipografía de programas informáticos

Javier Bezos

Versión 0.1. 2007-03-14.

Este documento tiene como propósito proporcionar una serie de reglas generales sobre la composición de código informático destinado a técnicos, autores, editores y correctores que tienen que tratar con obras o artículos de este tipo.

Código

El código escrito en algún lenguaje de programación deberá distinguirse tipográficamente del texto, normalmente con letra sin remates o mecanográfica.

Listados de código

En los bloques de código que ocupan varias líneas, se deberá prestar atención a la posibilidad de que la medida del texto sea inferior a la longitud de las líneas del código. Es ese caso, no se dejará que el código pase simplemente a la línea siguiente (y menos sin sangrar), sino que se organizará siguiendo las reglas sintácticas del lenguage, normalmente con sangrado adicional, y en ocasiones con el apoyo de algún signo especial (ninguno en C, Java o Pascal, pero \ en Python, _ en Visual Basic y ¬ en AppleScript, por ejemplo).

```
if (png_info (img)->valid & PNG_INFO_pHYs) {
    img_xres (img) = round (0.0254 *
        png_get_x_pixels_per_meter (png_ptr (img), png_info (img)));
    img_yres (img) = round (0.0254 *
        png_get_y_pixels_per_meter (png_ptr (img), png_info (img)));
}
```

y no

```
if (png_info (img)->valid & PNG_INFO_pHYs) {
    img_xres (img) = round (0.0254 *
png_get_x_pixels_per_meter (png_ptr (img), png_info (img)));
    img_yres (img) = round (0.0254 *
png_get_y_pixels_per_meter (png_ptr (img), png_info (img)));
}
```

También se puede emplear un signo especial que indique que una línea es continuación de la anterior y que debe considerarse que en el código es en realidad una sola línea:

```
if (png_info (img)->valid & PNG_INFO_pHYs) {
    img_xres (img) = round (0.0254 *
       ➥ png_get_x_pixels_per_meter (png_ptr (img), png_info (img)));
    img_yres (img) = round (0.0254 *
       ➥ png_get_y_pixels_per_meter (png_ptr (img), png_info (img)));
}
```

Alineación

Cuando la letra sea mecanográfica, se ajustarán las líneas de forma que los caracteres resulten alineados verticalmente, por lo que el espacio ha de ser fijo:

```
for f in fndf.split('\n'):
    f = f.strip()
end
```

y no

```
for f in fndf.split('\n'):
    f = f.strip()
end
```

Código en texto

Contra la norma general, los signos de puntuación pegados al código pero que no forman parte de él, deben ser de la letra principal, aunque conservando la figura y el trazo. Además, los espacios del código, incluso si forman parte de él, deben ser los que corresponden al texto (en la letra mecanográfica suelen ser fijos y más anchos, lo que resta uniformidad a los párrafos).

Entre los métodos de la clase `String` están: *`count`*, *`find`*, *`index`*, *`join`*, *`split`*, *`strip`* y otros.

y no

Entre los métodos de la clase `String` están: *`count`*, *`find`*, *`index`*, *`join`*, *`split`*, *`strip`* y otros.

La orden `a.put(x,i)` asigna `x` al valor `i`-ésimo de `a`.

y no

La orden `a.put (x, i)` asigna `x` al valor `i`-ésimo de `a`.

Puntuación

Los listados de código, es decir, el código que no está en el texto sino dispuesto aparte, no llevarán ningún signo de puntuación que no les corresponda, incluso si les pudiera corresponder por su posición en el texto (coma, punto, etc.). Si se trata de un código breve del texto, que cita una orden, la explica, etc., sí se pondrá la puntuación que corresponde al texto.

Contacto

Para errores, comentarios y sugerencias, puede ponerse en contacto conmigo a través de:

`http://www.tex-tipografia.com/contact.html`

La última versión de este documento está disponible en:

`http://www.tex-tipografia.com/typo.html`

Licencia