

★ Python 3 para impacientes ★



"Simple es mejor que complejo" (Tim Peters)

Python	IPython	EasyGUI	Tkinter	JupyterLab	Numpy
--------	---------	---------	---------	------------	-------

miércoles, 17 de septiembre de 2014

Markdown para Python (I)



Qué es Markdown

Markdown es un lenguaje de marcado ligero creado por John Gruber y Aaron Swart que permite la escritura rápida de textos. Literalmente, hace posible que no se tengan que despegar los dedos del teclado para que podamos concentrarnos en lo esencial: la escritura.

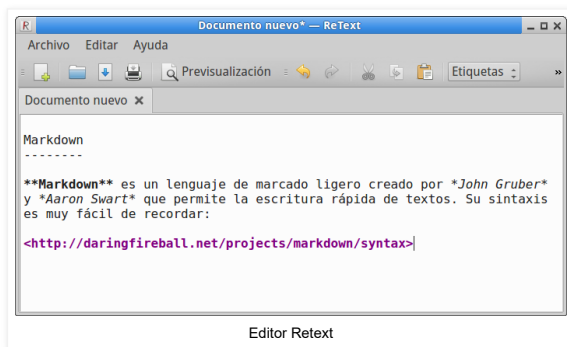
Esta característica lo convierte en un gran aliado a la hora de documentar nuestros proyectos Python (o escritos en otros lenguajes); un trabajo que siempre es necesario abordar pero que suele resultar, a veces, monótono y poco gratificante.

Para mejorar notablemente la productividad Markdown se vale de una serie de caracteres y sentencias que se introducen sutilmente entre el texto con una doble utilidad:

- Hacer que los textos sean más legibles y publicables.
- Aplicar un formato rudimentario a los textos que es aprovechado cuando son convertidos a otros formatos conocidos (.pdf, .odt, .rtf, .doc y .html). Por ejemplo, si en un texto Markdown hemos marcado una palabra en cursiva al convertirlo al formato .odt se le aplicará el formato cursiva a la misma palabra o si se convierte a .html se incluirán las etiquetas de cursiva de ese lenguaje para conseguir el mismo resultado.

Los ficheros de Markdown son de texto plano, un formato muy apropiado para mantener la documentación que no suele acompañar al propio código de los programas. Además, es utilizado en software de publicación web como wikis, blogs, etc. y hay editores de texto Markdown (y plugins) que nos facilitarán la tarea de escribir:

- Linux: gedit, ReText, UberWriter
- Windows: HippoEDIT, MarkdownPad



Referencia rápida de Markdown

En adelante, "e" indica lo que se ha de escribir y "r" el resultado a obtener

Buscar

Python para impacientes

[Python](#)
[IPython](#)
[EasyGUI](#)
[Tkinter](#)
[JupyterLab](#)
[Numpy](#)

Anexos

[Guía urgente de MySQL](#)
[Guía rápida de SQLite3](#)

Entradas + populares

[Dar color a las salidas en la consola](#)

En Python para dar color a las salidas en la consola (o en la terminal de texto) existen varias posibilidades. Hay un método basado ...

[Instalación de Python, paso a paso](#)

Instalación de Python 3.6 A finales de 2016 se produjo el lanzamiento de Python 3.6. El propósito de esta entrada es mostrar, pas...

[Añadir, consultar, modificar y suprimir elementos en Numpy](#)

Acceder a los elementos de un array. [], [i], ... Acceder a un elemento de un array. Para acceder a un elemento se utiliz...

[Variables de control en Tkinter](#)

Variables de control Las variables de control son objetos especiales que se asocian a los widgets para almacenar sus valore...

[Cálculo con arrays Numpy](#)

Numpy ofrece todo lo necesario para obtener un buen rendimiento cuando se trata de hacer cálculos con arrays. Por como está concebido...

[Operaciones con fechas y horas. Calendarios](#)

Los módulos datetime y calendar amplían las posibilidades del módulo time que provee funciones para manipular expresiones de ti...

[Tkinter: interfaces gráficas en Python](#)

Introducción Con Python hay muchas posibilidades para programar una interfaz gráfica de usuario (GUI) pero Tkinter es fácil d...

[Threading: programación con hilos \(I\)](#)

En programación, la técnica que permite que una aplicación ejecute simultáneamente varias operaciones en el mismo espacio de proceso se...

[Convertir, copiar, ordenar, unir y dividir arrays Numpy](#)

Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

[El módulo random](#)

El módulo random de la librería estándar de Python incluye un conjunto de funciones

- Formatos básicos, salto de línea y párrafos

Para escribir en cursiva colocaremos un asterisco o guión bajo delante y detrás del texto; para negrita serán dos y para negrita y cursiva, tres:

e: *Cursiva*
r: *Cursiva*

e: **Negrita**
r: **Negrita**

e: ***Cursiva y negrita***
r: ***Cursiva y negrita***

Dos espacios en blanco al final de una línea fuerzan un salto de línea y una línea en blanco indica el final de un párrafo y el comienzo del siguiente.

- Enlaces externos

Para escribir enlaces en nuestros textos seguiremos la sintaxis usada en los siguientes ejemplos:

e: <<http://www.correos.es>>
r: <http://www.correos.es>

e: [Markdown](<http://es.wikipedia.org/wiki/Markdown> "Wikipedia")
r: [Markdown](http://es.wikipedia.org/wiki/Markdown)

e: <nombre@gmail.com>
r: nombre@gmail.com

e: [Antonio](<mailto:nombre@gmail.com>)
r: [Antonio](mailto:nombre@gmail.com)

- Enlaces de referencia

En cualquier lugar de un texto podemos crear una lista de enlaces que pueden estar numerados o identificados por una etiqueta. Esa lista no será visible si convertimos el documento a .HTML pero se usará como índice para hacer referencia a los enlaces en cualquier lugar del documento.

e: Numerados. [Ubuntu][1] y [Firefox][2]
r: [Ubuntu](http://www.ubuntu.com) y [Firefox](http://www.firefox.org)

e: Etiquetas. Me gusta [Guadalinex][]
r: Me gusta [Guadalinex](http://www.guadalinex.org)

lista:
[1]: <http://www.ubuntu.com> "Ubuntu"
[2]: <http://www.mozilla.org> "Mozilla Firefox"
[Guadalinex]: <http://www.guadalinex.org>

- Citas

Una cita suele ser un texto pequeño que por su importancia queremos resaltar o diferenciarlo del resto. Para escribir una cita tendremos que dejar una línea en blanco y la comenzaremos con el carácter >. A continuación, se añadirá el texto que aparecerá sangrado por el margen izquierdo. Para una cita anidada emplearemos >>

Ejemplo de cita:

e:
> MarkDown es un lenguaje ligero de marcado de texto que permite escribir sin despegar los dedos del teclado.

r:
MarkDown es un lenguaje ligero de marcado de texto
que permite escribir sin despegar los dedos del teclado.

- Formato para código

Es el formato que se aplica a comandos y código fuente de programas que pueden aparecer en un documento.

que permiten obtener de distintos modos números a...

Archivo

septiembre 2014 (2) ▼

python.org



pypi.org



Sitios

- [ActivePython](#)
- [Anaconda](#)
- [Bpython](#)
- [Django](#)
- [Flask](#)
- [Ipython](#)
- [IronPython](#)
- [Matplotlib](#)
- [MicroPython](#)
- [Numpy](#)
- [Pandas](#)
- [Pillow](#)
- [PortablePython](#)
- [PyBrain](#)
- [PyCharm](#)
- [PyDev](#)
- [PyGame](#)
- [Pypi](#)
- [PyPy](#)
- [Pyramid](#)
- [Python.org](#)
- [PyTorch](#)
- [SciPy.org](#)
- [Spyder](#)
- [Tensorflow](#)
- [TurboGears](#)

Una línea de código se puede escribir entre dos acentos graves (') o bien, anteponiendo al código cuatros espacios en blanco o un salto de tabulación:

e: `chown root:root fichero.txt`

r:
`chown root:root fichero.txt`

Para varias líneas tendremos que dejar cuatro espacios en blanco, un tabulador o escribir el código entre dos grupos de acentos graves (```) colocados en la línea anterior y posterior al código.

e:

```
...
a, b = 0, 1
while b < 1000:
    print(b)
    a, b = b, a+b
...
```

r:

```
a, b = 0, 1
while b < 1000:
    print(b)
    a, b = b, a+b
```

- Listas

Para escribir una lista de elementos podemos utilizar los signos *, +, - seguido de un espacio en blanco (lista de viñetas) o comenzar cada elemento de la lista con un número seguido de un punto, 1. 2. ... (lista ordenada). Para escribir una lista dentro de otra dejaremos al principio de cada elemento anidado cuatro espacios o un salto de tabulación.

e:

```
+ Chocos fritos
+ Secreto ibérico
+ Tortilla de patatas
1. Cerveza
2. Refrescos
3. Café
```

r:

```
• Chocos fritos
• Secreto ibérico
• Tortilla de patatas

1. Cerveza
2. Refrescos
3. Café
```

- Encabezados

Con el carácter almohadilla "#" al principio o final de una línea podemos definir los encabezados de los apartados y subapartados de un documento, desde el nivel 1 "#" al nivel 6 "#####".

Cuando el documento se convierta a .odt (OpenDocument Text) o .html estas líneas tendrán asignado el estilo "Encabezado" o el tamaño de cabecera según el nivel indicado. Estos encabezados los utiliza LibreOffice/OpenOffice para generar un índice de forma automática.

e:

```
# Encabezado de nivel 1
## Encabezado de nivel 2
```

r:

Encabezado de nivel 1

Encabezado de nivel 2

Otra forma de definir los encabezados de primer y segundo nivel es "subrayando" con uno o más signos igual "=" o guiones "-" el texto de una línea.

e:

Encabezado Nivel 1

=====

Encabezado Nivel 2

r:

Encabezado Nivel 1

Encabezado Nivel 2

- Línea Horizontal

Para dibujar una línea horizontal escribiremos después de una línea en blanco tres o más asteriscos "*" o signos igual "=" . También, es posible incluir un espacio en blanco entre cada carácter y el siguiente.

e:

- - -

r:

- Imágenes

Para insertar una imagen en un documento tenemos que crear un enlace a la misma (indicando su ubicación y nombre) utilizando la siguiente sintaxis:

```
![Texto Alternativo] (/carpeta/imagen.png|URL "Título")
```

El "Texto Alternativo" se mostrará si la imagen no existe en el lugar indicado. El "Título" es el texto que aparecerá sobre la imagen al pasar el ratón sobre ella, cuando el documento se haya convertido a HTML.

e:

```
![Papel y lápiz](markdown.png "MarkDown")
```

r:



- Documentación oficial

Para conocer más detalles y ampliar la información sobre la sintaxis de Markdown consultar la [documentación ofrecida por John Gruber](#):

En el siguiente capítulo:

Después de esta breve introducción, en el [siguiente capítulo](#) estudiaremos el módulo **Markdown**, un desarrollo escrito en Python del lenguaje Markdown disponible para nuestros programas o para ejecutar desde la línea de comandos.

Publicado por Pherkad en [14:21](#)



Etiquetas: [Markdown](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)