

★ Python 3 para impacientes ★

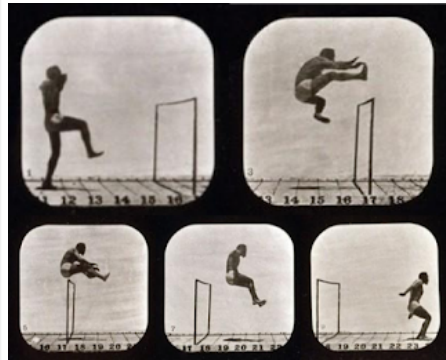


★ "Simple es mejor que complejo" (Tim Peters) ★

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

lunes, 4 de julio de 2016

Animaciones con Pyapng



El módulo **Pyapng** es un desarrollo reciente que permite crear animaciones **APNG** a partir de una secuencia de imágenes.

El formato **APNG** es una ampliación del formato de imágenes **PNG** que soporta animaciones. Normalmente, la extensión **.png** se suele utilizar con los dos tipos de archivos aunque las animaciones tienen reservada la extensión **.apng**.

Los archivos **PNG** con respecto a los **JPG** tienen la ventaja de utilizar un algoritmo de comprensión que consigue reconstruir las imágenes originales sin sufrir pérdidas. También, como sucede con los **GIFs**, soportan transparencias aunque en este caso admiten hasta 256 niveles de opacidad.

Instalación de Pyapng

Para construir animaciones basadas en imágenes **.png** sólo es necesario instalar el módulo **pyapng**:

```
$ pip install apng
```

Sin embargo, para poder crear animaciones que utilicen imágenes **.jpg** y de otros formatos instalar además el módulo [Pillow](#).

```
$ pip install pillow
```

Si tiene alguna versión de **Pillow** instalada desde hace tiempo se recomienda su actualización a la versión más reciente:

```
$ pip install pillow --upgrade
```

El navegador Internet Firefox reproduce las animaciones APNG sin necesidad de instalar ningún complemento. Para ver las animaciones de los ejemplos se recomienda abrir esta página con Firefox.

Crear una animación APNG

El siguiente ejemplo crea una animación **APNG** con las imágenes de la lista. Entre una imagen y la siguiente se introduce un retardo de 0,1 segundos (**delay=100**):

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. [], [..], ... 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...

[Tkinter: interfaces gráficas en Python](#)

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

[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...

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

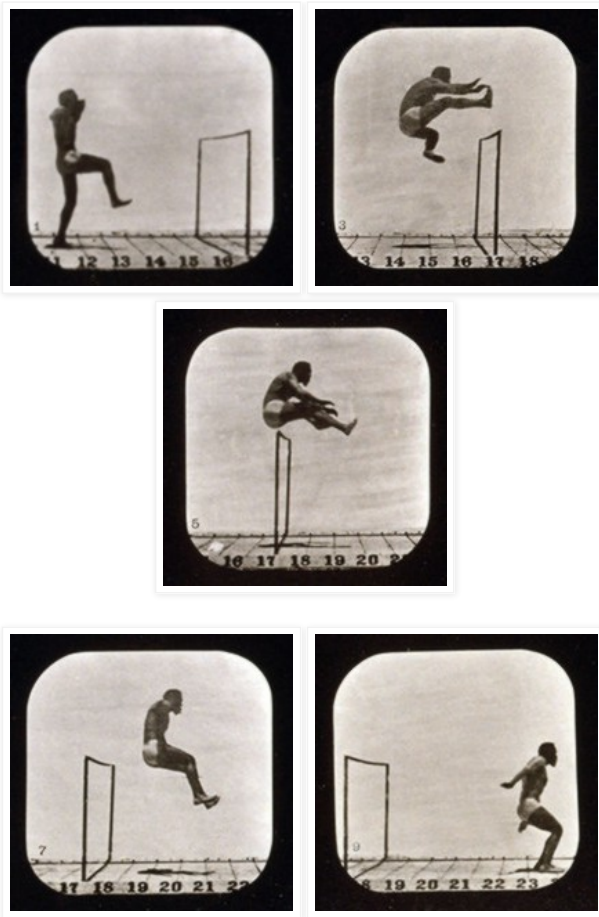
Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

[Tkinter: Tipos de ventanas](#)

Ventanas de aplicación y de diálogos En la entrada anterior tratamos los distintos gestores de geometría que se utilizan para di...

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

En programación, la técnica que permite que una aplicación ejecute



(Para probar ejemplos, descargar imágenes)

```
from apng import APNG
APNG.from_files(['atleta-01.jpg',
                 'atleta-02.jpg',
                 'atleta-03.jpg',
                 'atleta-04.jpg',
                 'atleta-05.jpg'],
               delay=100).save('animatleta1.png')
```



Crear una animación APNG con retardos diferentes

A continuación, un ejemplo similar al anterior pero que establece dos retardos distintos. El primero, de 0,2 segundos se utiliza con las cuatro primeras imágenes y el segundo, de 1 segundo, con la última imagen:

```
t1 = 200
t2 = 1000
atletas = [('atleta-01.jpg', t1),
           ('atleta-02.jpg', t1),
           ('atleta-03.jpg', t1),
           ('atleta-04.jpg', t1),
           ('atleta-05.jpg', t2)]

anima = APNG()
for archivo, delay in atletas:
```

simultáneamente varias operaciones en el mismo espacio de proceso se...

Archivo

julio 2016 (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](#)

```
anima.append(archivo, delay=delay)
anima.save('animatleta2.png')
```



Extraer imágenes de un archivo APNG

Por último, un ejemplo para mostrar el modo de extraer y guardar, una a una, la imágenes de una animación **APNG**:

```
anim = APNG.open('animatleta1.png')
cont = 1
for png, control in anim.frames:
    png.save('img-{cont}.png'.format(cont=cont))
    cont += 1
```

[Ir al índice del tutorial de Python](#)

Publicado por Pherkad en [12:13](#)



Etiquetas: [Python3](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)