

PYTHON: UN LENGUAJE DEL FUTURO

PONENTE: **JOSÉ RAMÓN ESPINOSA MUÑOZ**
(DATA ANALYST, DATA SCIENTIST, PYTHON DEVELOPER)





► Sesión única:

- ¿Qué es Python?
 - En pocas palabras...
 - ¿Por qué se creo, quién y de dónde tomaron ideas?
 - Características diferenciales
- Casos de éxito con Python
- Campos en los que se aplica Python.
 - Módulos implicados en cada campo
 - Resumen de uso en cada campo

- ¿Qué es Python?:
 - En pocas palabras...





PYTHON: UN LENGUAJE DEL FUTURO

- ¿Qué es Python?:
 - En pocas palabras...





PYTHON: UN LENGUAJE DEL FUTURO

- ¿Qué es Python?:
 - En pocas palabras...





■ ¿Por qué se creo, quién y de dónde tomaron ideas?

- Python fue creado por Guido Van Rossum a finales de los ochenta, principios de los noventa



- A pesar de que el logotipo hace referencia a la serpiente Python, el autor afirma que el nombre es un homenaje a los Monty Python.



PYTHON: UN LENGUAJE DEL FUTURO

- ¿Por qué se creo, quién y de dónde tomaron ideas?
 - Python se basa en una antiguo lenguaje de programación llamado “**abc**”, lenguaje que a su vez pretendió ser una alternativa a “Basic”. En ambos casos se pretendía sencillez en su desarrollo y un lenguaje más claro. También podríamos decir que ha tomado prestadas ideas de **Perl** y **Modula-3**

Recuperar las palabras de un documento en ABC

```
HOW TO RETURN words document:
  PUT {} IN collection
  FOR line IN document:
    FOR word IN split line:
      IF word not.in collection:
        INSERT word IN
collection
RETURN collection
```

Recuperar las palabras de un documento en Python

```
def words(document):
    collection = set()
    for line in document:
        for word in line.split():
            if word not in collection:
                collection.add(word)
    return collection
```





- ¿Por qué se creo, quién y de dónde tomaron ideas?
 - La necesidad de Python viene de la complejidad de algunos lenguajes como C.
 - Se pretendía tener la potencia de C, pero con una sintaxis sencilla y que se pudiera ejecutar en máquinas Linux.
 - Desde 2018, Guido Van Rossum está fuera del proyecto Python Software Foundation por decisión propia. En la actualidad 5 miembros coordinan la labor de dirigir la comunidad de programadores.



■ Características diferenciales

- Es un lenguaje de propósito general.
- Es multiparadigma, pues, a pesar de ser orientado a objetos, también es funcional y imperativo.
- Se basa en 2 versiones incompatibles entre si: Python 2 y Python 3
- Es un lenguaje interpretado.
- Es multiplataforma, sirve para Windows, Unix, Linux, Mac, etc..
- Es de tipado dinámico. No hay que definir las variables. También es de tipado fuerte, por lo que no se podrán, por ejemplo, sumar dos variables de tipo de dato distinto.
- La comunidad de desarrolladores de Python es enorme.
- Es de distribución libre, aunque existe una distribución llamada **Anaconda** que es de Microsoft.



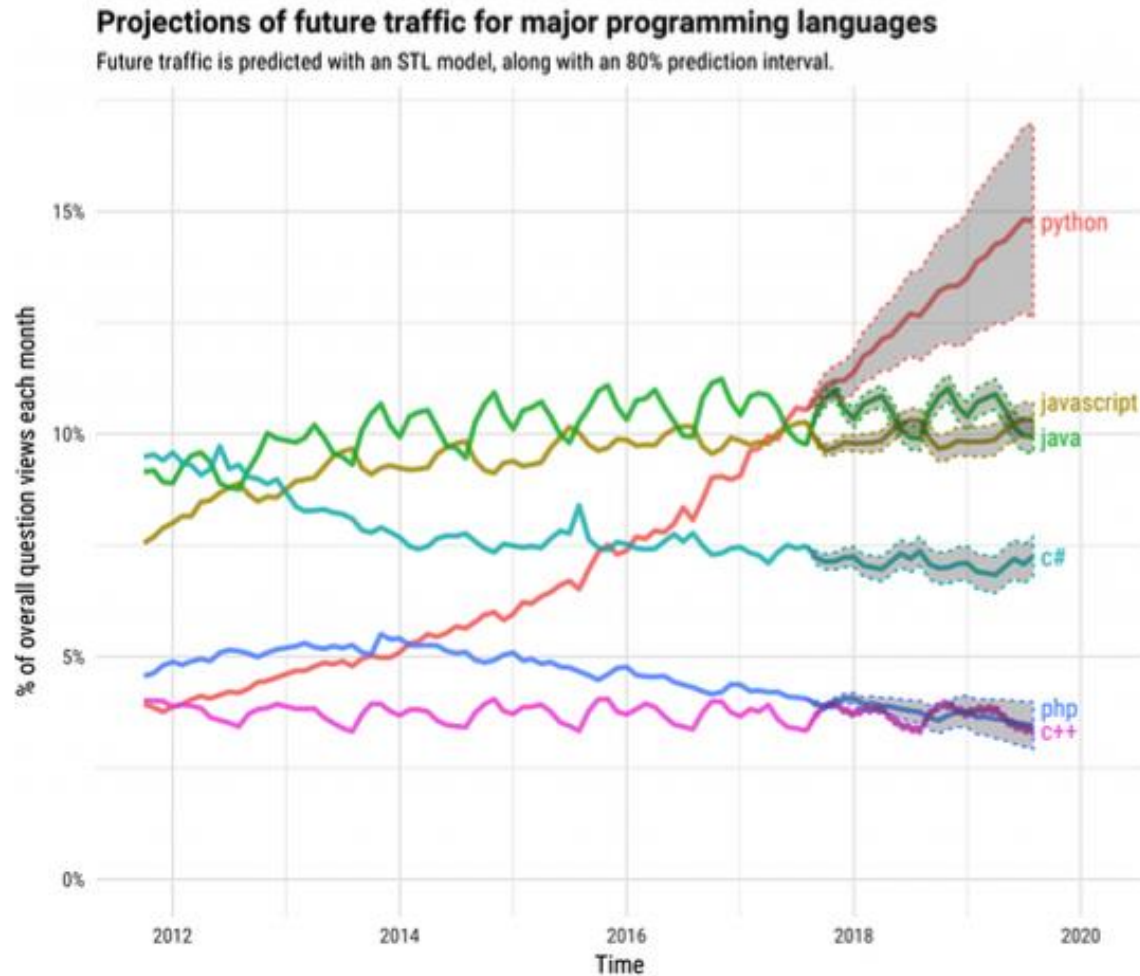
■ ...Y ahora, lo que interesa

- Python es más eficaz que otros lenguajes por:
 1. Ser gratuito y multiplataforma.
 2. No tener símbolos “raros” entre las instrucciones ayuda a que los programadores tarden menos en “picar” el código. (¡¡Java... tiembla!!!)
 3. Puede ser orientado a objetos, pero también funcional, pero también imperativo... Depende de lo que necesitemos en cada caso. ¿Y eso en que influye?
 4. La comunidad de diseñadores de módulos es muy grande y está por todo el mundo. Todo lo hacen por amor al arte. ¡¡Tenemos el departamento de I+D+i más grande del mundo y Gratis!!
 5. Así pasa, que tenemos esta gráfica:



PYTHON: UN LENGUAJE DEL FUTURO

- ...Y ahora, lo que interesa





- ...Y ahora, lo que interesa
 - Solo... algunas “cosinhas” negativas:
 - Como es **interpretado** y además el lenguaje es más cercano al humano que al de la máquina, le “cuesta” un poquito más hacer las cosas.
 - Que no hace mucho que se ha hecho popular, y es posible que sea más **difícil encontrar programadores** de Python que de Java.
 - Una comunidad tan grande es difícil de manejar. Por eso el que fundó Python se fue...algo cansado de ser el malo de la película. Lo cierto es que tenemos por ejemplo 2 python distintos que no son compatibles entre sí. Además **ciertas versiones se han abandonado**... Vamos, lo que tienen todas las democracias...



-
- The timeline illustrates the evolution of Python versions from 1994 to 2020. It is divided into three main sections: Python 1 (1994-2000), Python 2 (2000-2020), and Python 3 (2008-2020). Python 1 versions range from 1.0 to 1.6. Python 2 versions range from 2.0 to 2.7, with 2.7 being the final release. Python 3 versions range from 3.0 to 3.8, with 3.8 being the final release. The timeline also shows the release of Python 2.7.1, 2.7.2, 2.7.3, 2.7.4, 2.7.5, 2.7.6, 2.7.7, 2.7.8, 2.7.9, 2.7.10, 2.7.11, 2.7.12, 2.7.13, 2.7.14, 2.7.15, 2.7.16, 3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8.



■ ...Conservemos la calma: El ZEN de Python

- *Hermoso es mejor que feo.*
- *Explícito es mejor que implícito.*
- *Simple es mejor que complejo.*
- *Complejo es mejor que complicado.*
- *Sencillo es mejor que anidado.*
- *Escaso es mejor que denso.*
- *La legibilidad cuenta.*
- *Los casos especiales no son lo suficientemente especiales para romper las reglas.*
- *Lo práctico le gana a la pureza.*
- *Los errores no deben pasar en silencio. A menos que sean silenciados.*
- *En cara a la ambigüedad, rechazar la tentación de adivinar.*
- *Debe haber una - y preferiblemente sólo una - manera obvia de hacerlo. Aunque esa manera puede no ser obvia en un primer momento a menos que seas holandés.*
- *Ahora es mejor que nunca. Aunque "nunca" es a menudo mejor que "ahora mismo".*
- *Si la aplicación es difícil de explicar, es una mala idea.*
- *Si la aplicación es fácil de explicar, puede ser una buena idea.*





■ Casos de éxito con Python

- Python es muy usado por empresas dedicadas al sector de animación y de internet. Aunque actualmente los casos de éxito se están diversificando.



- ILM la empresa animadora de Star Wars. Aprovecha su fácil interoperabilidad con C y C++, siendo el lenguaje que mantiene el streaming con las bases de datos de imágenes.



- Google emplea Python en casi todos sus desarrollos de lado del servidor, dejando a C++ cuando dichos desarrollos lo imponen.



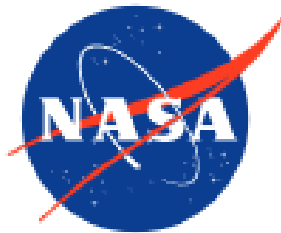
■ Casos de éxito con Python



- Python es actualmente responsable de múltiples servicios en gestión de infraestructura. Estos incluyen el uso de TORconfig, FBOSS y Dapper para programar y ejecutar tareas de mantenimiento.



- Utilizan Django, escrito en Python, para la implementación web de su plataforma.



- Python es utilizado en el Centro de Control de nuevas Misiones



PYTHON: UN LENGUAJE DEL FUTURO

■ Campos en los que se aplica Python.

■ **En desarrollo Web**

- Como lenguaje de Scripting
- Networking
- WebServices
- Front y Backend (Django)

■ **En Criptografía-Espionaje**

■ **En Ciberseguridad. Hacking ético**

■ **En Videojuegos – Cine**

■ **En Finanzas**

■ **En Pruebas**

■ **En Bases de datos**

- Relacionales
- NoSQL

■ **En Sistemas**

- Procesamiento de la señal
- Robótica y Automatización
- IoT

■ **En otras aplicaciones Científicas**

- Medicina-bioinformática
- Científico-matemática
- Estadística
- SIG
- Enseñanza

■ **En Machine Learning**

- Computer Vision - Facial
- Conducción autónoma
- Procesamiento del lenguaje / del Habla

■ **En Data Science**

- Análisis Redes sociales
- ETL-Data Wrangling
- Web -Scrapping

■ **En Big Data**



PYTHON: UN LENGUAJE DEL FUTURO

- Campos en los que se aplica Python.
 - Desarrollo Web
 - Como lenguaje de Scripting
 - Desde sus inicios, python fue lenguaje de Scripting en diseños HTML.
 - Actualmente ese uso no es tan común, ya que los módulos aportan encapsulamiento muy necesario en aras de la seguridad.
 - Networking
 - **asynchoro** - Marco de programación asíncrono y concurrente con líneas paralelas con interfaz similar a hilos. Junto con las librerías **threading** y **multiprocessing**
 - **requests**. Librería para realizar llamadas http (get, post, put, delete, etc.)
 - **Gevent** - biblioteca de red basada en Coroutine
 - **TwistedMatrix** - Marco de red impulsado por eventos
 - **RPyC** - RPC transparente. Marco de computación distribuida
 - **PyRO** – potente RPC OO
 - **HTTPLib2** : una biblioteca completa de clientes HTTP que admite muchas características que quedan fuera de otras bibliotecas HTTP, como httplib en la biblioteca estándar.



- Campos en los que se aplica Python.
 - Desarrollo Web
 - WebServices
 - **Flask:**
 - Flask es un framework escrito en Python que permite crear aplicaciones web rápidamente con modelo MVC y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.
 - Incluye un servidor web para ir viendo sobre la marcha el resultado.
 - Su curva de aprendizaje es sencilla, aunque las posibilidades de la herramienta pueden parecer lo contrario.
 - Instalación: `pip install flask`
<http://flask.pocoo.org/>
 - **Flask-Babel.**
 - Es un complemento de Flask que permite localizar los proyectos Flask internacionalmente.
 - Instalación: `pip install flask-babel`



PYTHON: UN LENGUAJE DEL FUTURO

■ Campos en los que se aplica Python.

■ Desarrollo Web

• Front y Backend

■ Django:

- Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–template
- Facilitar la creación de sitios web complejos ya que Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés Don't Repeat Yourself).
- Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.
- Otras características de Django son:
 1. Aplicaciones "enchufables" que pueden instalarse en cualquier página gestionada con Django.
 2. Una API de base de datos robusta.
 3. Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
 4. Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
 5. Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Instalación: `pip install django`

<https://www.djangoproject.com/>



- Campos en los que se aplica Python.
 - En Criptografía-Espionaje.
 - Criptografía básica
 - Python hace uso de la librería incluida en la librería estándar: **hashlib**
 - **hashlib** implementa una interfaz común para muchos algoritmos de resumen de hash y resumen de mensajes.
 - Se incluyen los algoritmos hash seguros SHA1, SHA224, SHA256, SHA384 y SHA512 de FIPS (definidos en FIPS 180-2), así como el algoritmo MD5 de RSA
 - Criptografía “strong” (AES y RSA)
 - Python permite utilizar el módulo **pycryptodome** que sustituye a **pycrypto**
<https://pycryptodome.readthedocs.io/en/latest/src/installation.html>
 - Esta librería permite proteger la información, con cifrado simétrico, asimétrico e híbrido.
 - Las funcionalidades específicas de este modulo son las siguientes:



- Campos en los que se aplica Python.
 - En Criptografía-Espionaje.

Paquete	Descripción
Crypto.Cipher	Módulos para proteger la confidencialidad , es decir, para cifrar y descifrar datos (ejemplo: AES).
Crypto.Signature	Módulos para asegurar la autenticidad , es decir, para crear y verificar firmas digitales de mensajes (ejemplo: PKCS # 1 v1.5).
Crypto.Hash	Módulos para crear resúmenes criptográficos (ejemplo: SHA-256).
Crypto.PublicKey	Módulos para generar, exportar o importar <i>claves públicas</i> (ejemplo: RSA o ECC).
Crypto.Protocol	Módulos para facilitar las comunicaciones seguras entre las partes, en la mayoría de los casos mediante el uso de primitivas criptográficas de otros módulos (por ejemplo, el esquema de intercambio secreto de Shamir).
Crypto.IO	Módulos para tratar las codificaciones comúnmente utilizadas para datos criptográficos (ejemplo: PEM).
Crypto.Random	Módulos para generar datos aleatorios.
Crypto.Util	Rutinas de propósito general (ejemplo: XOR para cadenas de bytes).



- Campos en los que se aplica Python.
 - En Criptografía-Espionaje.
 - Criptografía en IoT
 - El auge de los dispositivos IoT supone un problema de seguridad importante.
 - La criptografía es utilizada en las capas 2, 3 y 4 del modelo OSI.
 - Python tiene mucho que decir ya que, por un lado, va a poder programar dispositivos IoT (Ver sección) y por otro aplicarle las funciones de sus módulos de criptografía.
 - Python en el Espionaje.
 - Un dato curioso es conocer que Python se está convirtiendo en una herramienta muy útil en el nuevo espionaje del siglo XXI.
 - Existe un libro muy curioso editado por Packt Publishing: **Python for Secrets Agents**, donde se tratan temas variados desde scrapping hasta criptografía.
<https://www.packtpub.com/hardware-and-creative/python-secret-agents>



- Campos en los que se aplica Python.
 - En Ciberseguridad. Hacking ético
 - Python también se encuentra en el mundo del hacking blanco o ético.
 - Existen una serie de módulos que podemos importar para Python. Vamos a dar una ligera descripción de cada uno de ellos y su uso:
 - **python-nmap**: Este modulo descubre host y servicios en una red de ordenadores, creando un mapa de la red. Su uso a ambos lados (hacking blanco y negro) es habitual. En hacking blanco su uso permite detectar vulnerabilidades ya que encuentra puertos abiertos.
 - La instalación en python es: **pip install python-nmap**
 - **hashlib**: Esta librería que ya hemos visto en la parte de criptografía es también usada para evidenciar si ha habido modificación de la información en la recogida de pruebas tras un ataque.
 - **pexpect**: Este módulo se suele usar para automatizar aplicaciones interactivas. Permite responder a lo que los programas proponen dando sensación de ser un humano el que responde en la líneas de comando.
 - Uno de los scripts más utilizados de pexpect, es **pxssh**, que interactúa con sesiones SSH con métodos de login, etc..



- Campos en los que se aplica Python.
 - En Ciberseguridad. Hacking ético
 - **_winreg**: Esta es una librería propia de Python. Nos permite leer el registro de Windows. Con esto se puede saber qué páginas se abren con una wifi, y si conseguimos las direcciones MAC además podemos conocer la localización geográfica del atacante.
 - **pygeoip**: librería que permite relacionar la IP con un lugar físico.
<http://code.google.com/p/pygeoip/>
 - **scapy**: Es una de las librerías más peligrosas del espectro python. Si se usa para bien, permite detectar muchas vulnerabilidades, incluso detectar intrusos, si se usa mal, se pueden realizar diversos ataques como el sniffing (escucha de paquetes)



► GUI Python

■ Librerías para crear interfaces de usuario. Características

- Tkinter: Es la interfaz de facto, preinstalada con Python. Buena para proyectos personales o de aprendizaje.
 - Es simple y fácil de aprender
 - Tiene una documentación completa
 - Sin embargo:
 - No tiene demasiados elementos gráficos. Sólo los básicos
 - Control limitado.

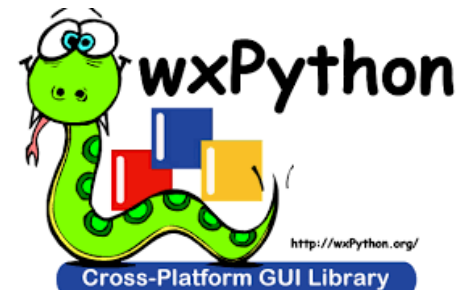




► GUI Python

■ Librerías para crear interfaces de usuario. Características

- WxPython: Está basada en WxWidgets de C / C++, sería la interfaz por defecto si no existiese Tkinter.
 - Es más completa que Tkinter en elementos gráficos
 - Control flexible de la interfaz.
 - Apariencia nativa
 - Es independiente del entorno en el que se ejecuta.
 - Es sencillo crear componentes personalizados
 - Tiene buena documentación y ejemplos
 - Sin embargo
 - Hay que instalarla
 - Tiene una curva de aprendizaje más alta
 - Es complejo realizar interfaces multiplataforma.





► GUI Python

■ Librerías para crear interfaces de usuario. Características

- PyQT: Está basada en la librería QT de C / C++
 - Es más completa que Tkinter en elementos gráficos
 - Control flexible de la interfaz. Se pueden definir los eventos más sencillos en el propio GUI
 - Puede trabajar con entornos móviles
 - Es Rápida y de Apariencia nativa
 - Sin embargo
 - Hay que instalarla
 - Tiene una curva de aprendizaje más alta
 - A veces hay que hacer cast entre tipos de datos
 - Falla en documentación





■ Campos en los que se aplica Python.

■ En Videojuegos – Cine

- Dentro del mundo de videojuegos en python, el módulo **pygame** es la estrella.



- Se instala con : **pip install pygame**
- Pygame permite manejar sprites, crear animaciones.
- Suele estar especializado en Juegos 2D.
- Sin embargo, en combinación con SDL, que puede controlar el tema gráfico a través de su acceso a la aceleradora gráfica. Los juegos de pygame no se quedan solo en 2D

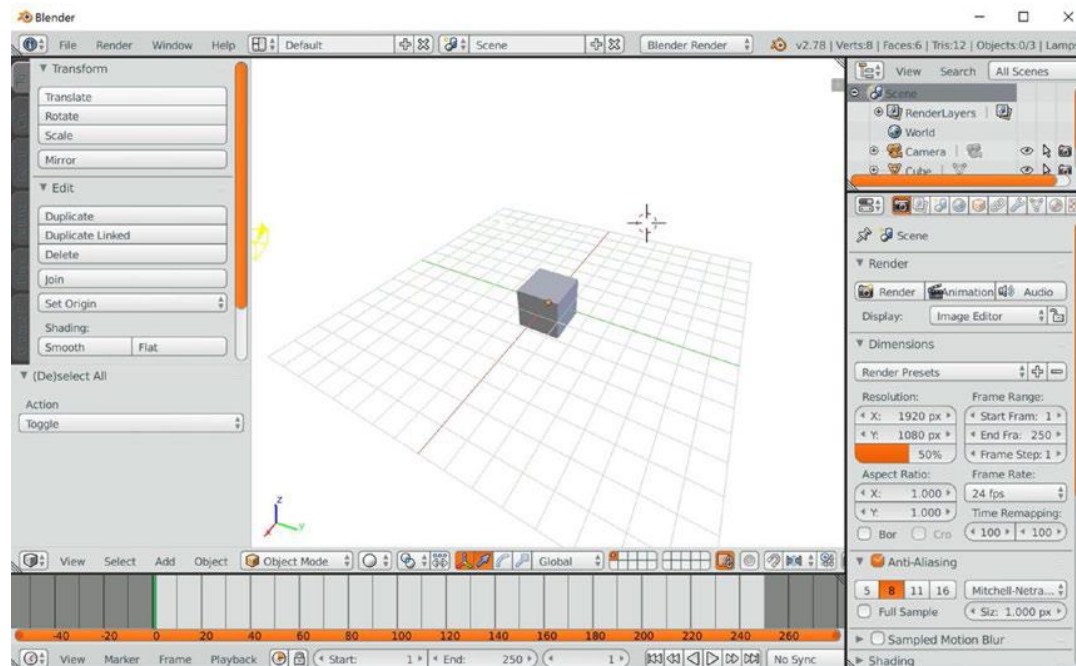


- Campos en los que se aplica Python.
 - En Videojuegos – Cine
 - Otras alternativas:
 - [Pyglet](#): Pyglet es la alternativa a Pygame+PyOpenGL permitiendo aceleración gráfica y trabajar en 2d/3d, todo ello en una sola librería, y sin tener que recurrir a la engorrosa dificultad de PyOpenGL.
 - Es más difícil de usar que Pygame.
 - Pyglet es una librería con un desarrollo actual más rápido que pygame
 - pyglet.org*
 - `pip install pyglet`*
 - [Panda3d](#): Librería libre centrada en videojuegos 3D escrita para Python y C++.
 - Algo compleja pero profesional.
 - Algunos juegos de Disney han usado esta librería.
 - [Cocos2d](#): Este framework es una extensión de Pyglet que añade gran cantidad de efectos, rotar y manipular sprites, transiciones entre escenas con estilos, etc..



PYTHON: UN LENGUAJE DEL FUTURO

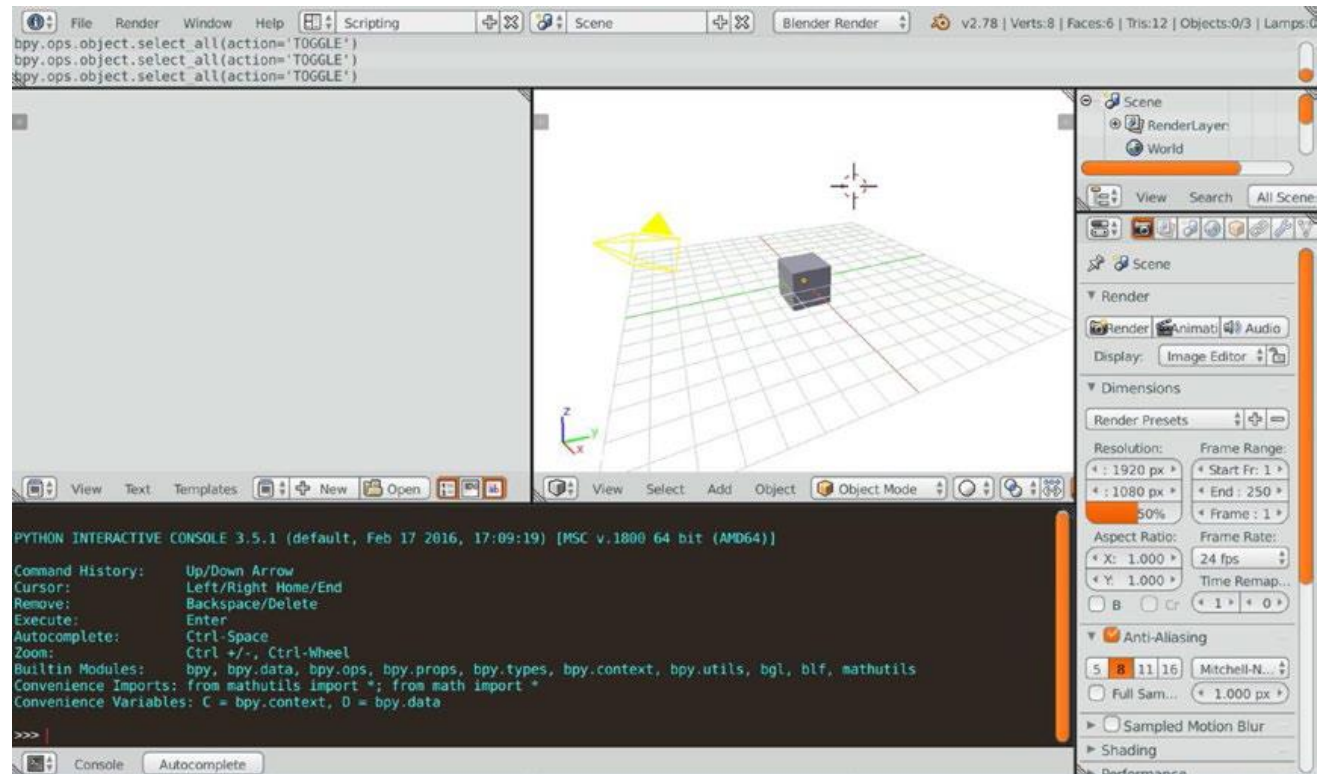
- Campos en los que se aplica Python.
 - En Videojuegos – Cine
 - Otra serie de módulos aplicados, a la producción de Videojuegos y de escenas diseñadas por ordenador en el cine son :
 - **El API para Python de Blender:** Blender es una herramienta de modelado 3D de gran potencia y fama dentro de los diseñadores de VideoJuegos o de escenas de animación en el cine.





PYTHON: UN LENGUAJE DEL FUTURO

- Campos en los que se aplica Python.
 - En Videojuegos – Cine
 - **El API para Python de Blender:**
 - La forma de crear scripts que supongan acciones automatizadas en blender se puede realizar en Python:





- Campos en los que se aplica Python.
 - En Videojuegos – Cine
 - **El API para Python de Blender:**
 - La librería que permite hacer todo eso es **bpy**
 - Su instalación es sencilla: ***pip install bpy***
 - Como contrapartida, es necesario tener instalado Visual Studio para poder realizar el modelo.
 - Algunos sub-módulos de bpy son la siguiente:
 - **bpy.ops**: Contiene operadores para manejar objetos de juego.
 - **bpy.data**: Sirve para acceder a los datos internos. Por ejemplo, la posición de un objeto.
 - Otros módulos son: **bpy.types**, **bpy.utils**, **bpy.props** y **bpy.path**.
 - Con estos módulos y Python, podemos ser capaces de manipular la posición, rotación, etc. de los objetos.



- Campos en los que se aplica Python.
 - En Videojuegos – Cine
 - **El API para Python de Maya:**
 - Al igual que para Blender, Python puede acceder a la api de Maya.
 - Maya es un programa dedicado a realizar objetos 3D, animaciones y efectos especiales tanto para el mercado de Video como para el cine.
 - Para importarla en un problema de python, importamos: **import maya.cmds;**
 - Para instalarla: **pip install maya-cmds**
 - Con esta librería podemos programar fácilmente los objetos 3D por código.
 - Tal vez la curva de aprendizaje de esta herramienta sea muy pronunciada.



■ Campos en los que se aplica Python.

■ En Finanzas

- Cada día que pasa, el mundo de las finanzas se automatiza más y más.
- Python desde su aparición ha tenido grandes vinculaciones con el mundo de las finanzas.
- Actualmente, esas vinculaciones se mantienen en forma de una gran cantidad de librerías que nos ayudan a realizar aplicaciones.
- **pandas_datareader**: Accede de forma remota a información de carácter bursátil, a través de terceros y la almacena en forma de DataFrames muy útiles en entornos Python.
 - Instalación: `pip install pandas-datareader`
 - Su página web: <https://pandas-datareader.readthedocs.io/en/latest/whatsnew.html>
 - Es necesario estar al corriente de sus noticias porque sus fuentes suelen dar problemas.



■ Campos en los que se aplica Python.

■ En Finanzas

- **numpy-lib-financial**: Es otra de las grandes librerías dedicadas al tema financiero. Sirve como calculadora financiera de una serie de conceptos.
 - Forma parte de la librería numpy.
- **statsmodels**: Es una librería de análisis estadístico. Muy útil para lo que se denominan series temporales. Los gráficos de bolsa se comportan como una serie temporal.
 - Dispone de una gran cantidad de modelos estadísticos para implementar: ANOVA, GAM, series temporales, etc..
 - **statsmodels.tsa**: Módulo de serie temporales. Incorpora los modelos AR, ARMA, ARIMA, SARIMAX y muchas operaciones con datos de series temporales.
 - **statsmodels.tsa.statespace**: contiene clases y funciones que son útiles para el análisis de series de tiempo usando métodos de espacio de estado.
 - Los state spaces son un modelo matemático de un sistema físico como un conjunto de variables de entrada, salida y estado relacionadas mediante ecuaciones diferenciales de primer orden.



■ Campos en los que se aplica Python.

■ En Finanzas

- **stats**: Es un módulo con un gran número de funciones estadísticas ya listas.
 - La lista completa de funciones se encuentra en:
 - <https://www.statsmodels.org/stable/stats.html>
- **scipy**: Es un módulo especializado en matemáticas e ingeniería. Está muy relacionado con **numpy**. Almacena las constantes más importantes de los ámbitos matemático, físico, químico, etc..
- **finance**: Cálculos de riesgos financieros. optimizado para aportar facilidad de uso.
 - Instalación: `pip install finance`
- **tradingmachine**: Un backtester para algoritmos financieros.
 - Instalación: `pip install tradingmachine`
- **economics**: Librería de funciones de manipulación de datos económicos.
 - Instalación: `pip install economics`
 - <https://github.com/trickvi/economics>



■ Campos en los que se aplica Python.

■ En Pruebas

- Python además de ser un lenguaje fácil de programar, es un lenguaje fácil de probar. Para ello utiliza una serie de herramientas
 - **assert**: Es la forma más sencilla de crear pruebas unitarias, esta herramienta sólo espera que le introduzcamos la función a evaluar y el resultado a esperar.
 - **unittest**: Es un módulo que se encuentra en el núcleo estándar de Python. En Python 2, unittest se llaman unittest2.
 - **nose2**: Totalmente integrado con unittest, también puede automatizar toda la ejecución de pruebas desarrollada en unittest. Busca todas las pruebas definidas por unittest y las ejecuta. Permite filtrar pruebas
 - Instalación: `pip install nose2`
 - https://nose2.readthedocs.io/en/latest/getting_started.html
 - **pytest**: También soporta todos los casos de prueba definidos con unittest. Filtra por casos de prueba. Permite volver a ejecutar desde la última prueba fallida. Contiene unos plugins que aumentan funcionalidad. Es la herramienta más usada
 - Instalación: `pip install pytest`
 - <https://docs.pytest.org/en/latest/>



- Campos en los que se aplica Python.
 - En acceso a Bases de datos
 - Python dispone de librerías para la gestión de algunas de las bases de datos relacionales como MySQL, sqlite3, etc. y para todas las nuevas Bases de Datos NoSQL.
 - Bases de datos relacionales: Todos los módulos siguientes, consiguen desarrollar consultas sql sobre las bases de datos.
 - **pysqlite3**: Módulo gestor de bases de datos sqlite 3 en Python.
 - Instalación: `pip install pysqlite3`
<https://github.com/coleifer/pysqlite3>
 - **pymysql**: Cliente MySQL en Python
 - Instalación: `pip install pymysql`
<https://pymysql.readthedocs.io/en/latest/>
 - **pandas-oracle**: Modulo que permite trabajar con BBDD Oracle a través e Panda.
 - Instalación: `pip install pandas-oracle`
https://github.com/cwade/pandas_oracle

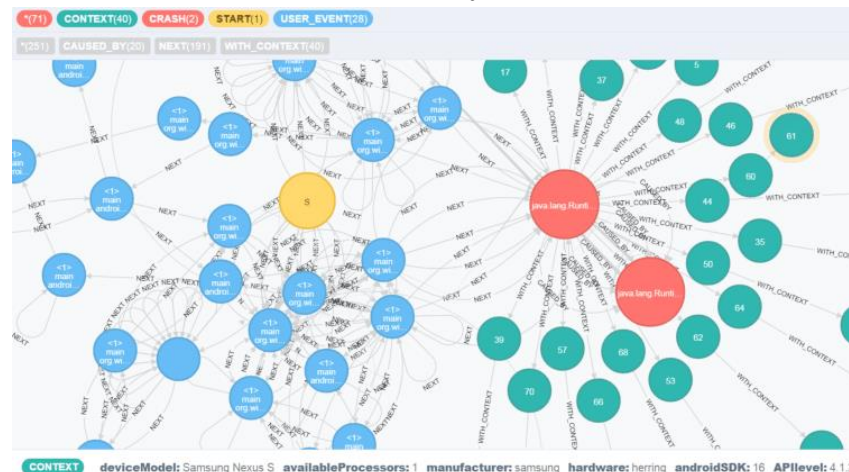


- Campos en los que se aplica Python.
 - En acceso a Bases de datos
 - Otros módulos genéricos:
 - **SQLAlchemy**: Proporciona un conjunto completo de patrones de persistencia de nivel empresarial, diseñados para el acceso a bases de datos eficientes y de alto rendimiento, adaptados a un lenguaje de dominio simple y de Pythonic.
 - Instalación: `pip install SQLAlchemy`
<https://www.sqlalchemy.org/>
 - **SQLObject**: Es un administrador relacional de objetos para proporcionar una interfaz de objeto a su base de datos, con tablas como clases, filas como instancias y columnas como atributos.
 - SQLAlchemy incluye un lenguaje de consulta basado en objetos Python que hace que el SQL sea más abstracto y proporciona una importante base de datos de independencia para las aplicaciones.
 - Soporta MySQL, PostgreSQL, SQLite, Firebird, Sybase, MSSQL y MaxDB (SAPDB).
 - Instalación: `pip install SQLObject`
<http://sqlobject.org/>



PYTHON: UN LENGUAJE DEL FUTURO

- Campos en los que se aplica Python.
 - En acceso a Bases de datos
 - Bases de datos NoSQL:
 - **pymongo**: Driver de acceso al cluster de MongoDB, base de datos de tipo documental.
 - Instalación: `pip install pymongo`
<https://api.mongodb.com/python/current/>
 - **py2neo**: Driver de manipulación de la base de datos en grafo Neo4J.
 - Instalación: `pip install py2neo`





- Campos en los que se aplica Python.
 - En acceso a Bases de datos
 - Bases de datos NoSQL:
 - **redis-py**: Interfaz de acceso a todos los comandos de la base de datos Clave-valor Redis.
 - Instalación: `pip install redis`
<https://redis-py.readthedocs.io/en/latest/#>
 - **cassandra**: Driver de uso de Cassandra, la más conocida de las BBDD columnares. Cassandra se utiliza cuando tengamos necesidad de gran velocidad de respuesta en BBDD.
 - Instalación: `pip install cassandra-driver`
<https://datastax.github.io/python-driver/>



PYTHON: UN LENGUAJE DEL FUTURO

■ Campos en los que se aplica Python.

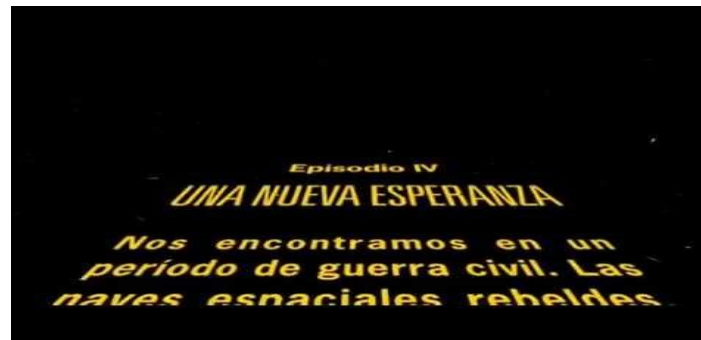
■ En Sistemas

- Un campo donde Python ha destacado sorprendentemente es en Sistemas.

Hace mucho tiempo, en una galaxia
muy, muy lejana....



- **Sistemas** siempre ha estado bajo el **imperio del lenguaje C**. Pero un grupo de programadores hartos de las difíciles instrucciones del emperador C, se están rebelando contra él, apoyadas por las naves especiales de Python.





■ Campos en los que se aplica Python.

■ En Sistemas

- Procesamiento de la señal

- Hay varios módulos que permiten capturar información de sensores e instrumentos. Los más importantes son:

- **pyVisa**: Permite controlar todo tipo de dispositivos de medición independientemente de la interfaz (por ejemplo, GPIB, RS232, USB, Ethernet)

Instalación: `pip install pyvisa`

<https://pyvisa.readthedocs.io/en/latest/introduction/getting.html>

Su funcionamiento es muy intuitivo, en un campo tan complejo:

1. Se importa
2. Se crea un `resourceManager`
3. Se le da la lista de dispositivos a controlar
4. Se le asigna uno y se le hacen “Queries”



■ Campos en los que se aplica Python.

■ En Sistemas

- Procesamiento de la señal

- **Familia de productos Native Instruments:**

- NI-DCPower (módulo Python: **nidcpower**)
 - NI-DMM (módulo Python: **nidmm**)
 - NI-FGEN (módulo Python: **nifgen**)
 - NI-SCOPE (módulo Python: **niscope**)
 - NI-SWITCH (módulo Python: **niswitch**)
 - NI-ModInst (módulo Python: **nimodinst**)
 - NI Switch Executive (módulo Python: **nise**)

<https://nimi-python.readthedocs.io/en/master/>

- **Familia de productos CompactDAQ:**

Esta popular familia de sensores se puede trabajar en Python con el módulo **ni-daqmx**

<https://nidaqmx-python.readthedocs.io/en/latest/>



■ Campos en los que se aplica Python.

■ En Sistemas

• IoT

- En pocos años, el fenómeno Internet of Things ha revolucionado los electrodomésticos caseros. Python está ahí... también.
- En este campo, están involucrados ciertos módulos, algunos que tienen que ver con las propias placas y otros con el control de Internet.

■ mraa

- Es una biblioteca de GPIO para la mayoría de las placas que admiten Python.
- Es una biblioteca de alto nivel: leer y escribir en pines se hace en una sola línea
- También brinda soporte para protocolos de comunicación como I2C, UART y SPI.
- Instalación: <https://github.com/intel-iot-devkit/mraa>
<https://github.com/intel-iot-devkit/mraa/tree/master/examples/python>



■ Campos en los que se aplica Python.

■ En Sistemas

- IoT

- **sockets**

- Paquete que facilita la creación de redes a través de TCP / IP y UDP utilizando Python. Proporciona acceso a las API de socket de Berkeley para acceder a Internet. Tanto TCP / IP como UDP son protocolos de capa de transporte, son ideales para la comunicación con dispositivos en la misma red WiFi.

- Instalación: `pip install sockets`

<https://github.com/eshiofune/sockets>

- **bs4**

- Beautiful Soup es una de las librerías más potentes de Python, sirve para parsear páginas web.

- Instalación: `pip install beautifulsoup4`

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>



■ Campos en los que se aplica Python.

■ En Sistemas

- IoT

- **request**

- Es el paquete de solicitudes se utiliza en Python para realizar llamadas HTTP y analizar respuestas. Este paquete es útil cuando se trata de servicios de nube de terceros basados en HTTP.

- Instalación: `pip install request`

- <https://github.com/looking-for-a-job/request.py>

- **paho-mqtt**

- MQTT es un protocolo desarrollado exclusivamente para el paradigma de Internet de las cosas. Su especialidad es la comunicación de alta velocidad.

- Las solicitudes de MQTT se pueden hacer directamente dentro de Python, sin necesidad de realizar ninguna configuración adicional.

- Instalación: `pip install paho-mqtt`

- <http://www.eclipse.org/paho/>

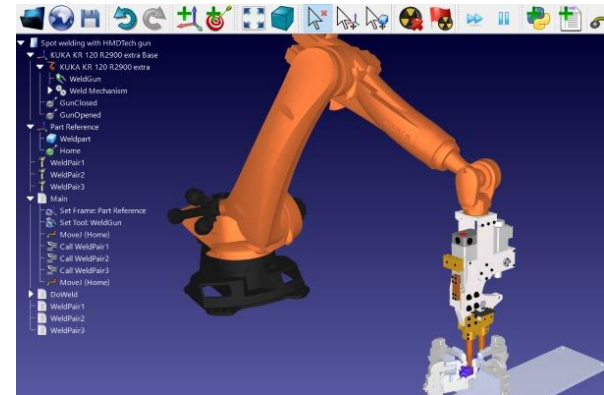


- Campos en los que se aplica Python.
 - En Sistemas
 - Robótica
 - **Robot Framework**
 - Robot Framework es un marco de automatización de código abierto genérico para pruebas de aceptación, desarrollo dirigido por pruebas de aceptación (ATDD) y automatización de procesos robóticos (RPA). Tiene una sintaxis sencilla de texto sin formato
 - Instalación: `pip install robotframework`
<https://robotframework.org/>
 - **pybotics**
 - pybotics es un paquete de herramientas para el control de automatización de robots.
 - Instalación: `pip install pybotics`
<https://github.com/nnadeau/pybotics>



PYTHON: UN LENGUAJE DEL FUTURO

- Campos en los que se aplica Python.
 - En Sistemas
 - Robótica
 - **RoboDK**
 - RoboDK permite crear simulaciones con robots y mecanismos, así como generar programas específicos para robots. Se puede usar la interfaz gráfica de usuario para crear programas, y ampliar sus capacidades con Python. La API de RoboDK para Python integra todas las funciones de programación fuera de línea del simulador de RoboDK y le permite implementar aplicaciones automatizadas para una gran variedad de robots y mecanismos.
 - Instalación: `pip install robodk`



- <https://robodk.com/doc/en/PythonAPI/index.html>



- Campos en los que se aplica Python.
 - En otras aplicaciones Científicas
 - Python no se detiene en las máquinas. Su espectro de posibilidades se extiende al ámbito científico y de ciencias de la salud. Vamos a ver la posibilidades en los siguientes campos
 - Medicina
 - Bioinformática
 - Matemática y Estadística
 - SIG
 - Enseñanza
 - Medicina-bioinformática
 - Nota: A pesar de mostrar algunos ejemplos de librerías Python, hay que aclarar que lo que más está avanzando a Python en este campo es su uso con Redes Convolucionales.
 - **medpy**
 - Es un módulo dedicado al procesamiento de imágenes para medicina.
 - Instalación: `pip install MedPy`
- <http://loli.github.io/medpy/>



PYTHON: UN LENGUAJE DEL FUTURO

■ Campos en los que se aplica Python.

■ En otras aplicaciones Científicas

- Medicina-bioinformática

- **pymedtermino**

- Es un módulo dedicado a la búsqueda de términos relacionados con medicina. Se admiten las siguientes terminologías: iconos SNOMED CT, ICD10, MedDRA, CDF, UMLS y VCM
 - Instalación: pip instalar PyMedTermino

- http://www.lesfleursdunormal.fr/static/informatique/pymedtermino/index_en.html

- **pygeno**

- pyGeno es una base de datos bioinformática. La característica más importante de pyGeno, es que permite trabajar sin problemas con genomas de referencia y personalizados .Instalación:
 - pip instalar pygeno

- <http://pygeno.irc.ca/>

- Enlace sobre StartUps que usan python:

- <https://belitsoft.com/python-development-services/python-healthcare>



- Campos en los que se aplica Python.
 - En otras aplicaciones Científicas
 - Matemática y Estadística
 - Al igual que su medio hermano R, Python fue inicialmente pensado para Estadística y Matemática. Por ello tiene muchos módulos pensados para tal fin. Vamos a citar algunos:
 - **statistics**
 - Este módulo proporciona funciones para calcular estadísticas y otras funciones matemáticas con datos numéricos.
 - Instalación: `pip install statistics`
<https://docs.python.org/3/library/statistics.html>
 - **statsmodels**
 - Este módulo proporciona clases y funciones para la estimación de muchos modelos estadísticos diferentes, así como para realizar pruebas estadísticas y la exploración de datos estadísticos.
 - Instalación: `pip install statsmodels`
<http://www.statsmodels.org/devel/>



PYTHON: UN LENGUAJE DEL FUTURO

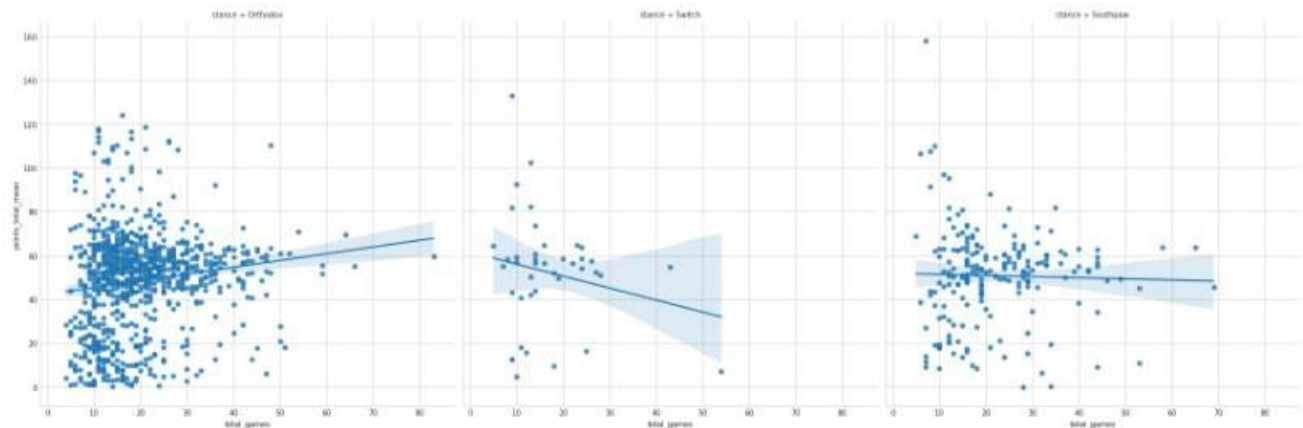
■ Campos en los que se aplica Python.

■ En otras aplicaciones Científicas

- Matemática y Estadística

- **Seaborn + Matplotlib + plotly**

- Estos tres módulos proporcionan visualización de datos. Juntos aportan un gran número de posibilidades gráficas, tanto 2d como 3d



- Instalación: `pip install matplotlib`
- Instalación: `pip install seaborn`
- Instalación: `pip install plotly` (<https://plot.ly/python/>)



- Campos en los que se aplica Python.
 - En otras aplicaciones Científicas
 - Matemática y Estadística
 - Otras librerías interesantes en el ámbito científico:
 - **sympy**
 - Módulo de tratamiento de Algebra
 - Instalación: `pip install sympy`
 - **numpy & scipy**
 - Las conocidas librerías de referencia en Python
 - Instalación: `pip install numpy`
 - Instalación: `pip install scipy`
 - **patsy**
 - Módulo de descripción de modelos estadísticos.
 - Instalación: `pip install patsy`
 - <https://patsy.readthedocs.io/en/latest/>



- Campos en los que se aplica Python.
 - En otras aplicaciones Científicas
 - Educación
 - La librería por excelencia para educación en Python es:
 - **Jupyter notebook**
 - Antiguo Ipython.
 - Permite crear notebooks que incluyen código Python, R, Scala etc. totalmente ejecutable además de textos, audio , enlaces, imágenes.
 - Como novedad permite total integración con Spark, la plataforma más utilizada actualmente en Big Data.
 - Está optimizada también para el debugging de código.
 - Instalación: `pip install jupyter`
- <https://jupyter.org/>



- Campos en los que se aplica Python.
 - En otras aplicaciones Científicas
 - SIG
 - Un campo muy en boga es el análisis geoespacial. Python trabaja una serie de módulos que pasamos a describir.
 - **RTree**: Wrapper Python de ctypes de libspatialindex que proporciona una serie de características avanzadas de indexación espacial.
 - Instalación: `pip install rtree`
<http://toblerity.org/rtree/>
 - **GDAL** : Biblioteca de traducción para formatos de datos geoespaciales ráster y vectoriales
 - Instalación: `pip install GDAL`
<https://gdal.org/>
 - **Fiona** : Lectura y escritura de archivos de datos espaciales.
 - Instalación: `pip install fiona`
<https://fiona.readthedocs.io/en/latest/>



- Campos en los que se aplica Python.
 - En otras aplicaciones Científicas
 - SIG
 - **Shapely** : Manipulación y análisis de objetos geométricos.
 - Instalación: pip install Shapely
 - <https://shapely.readthedocs.io/en/latest/>
 - **GeoPandas** : El objetivo de GeoPandas es facilitar el trabajo con datos geoespaciales en python. Combina las capacidades de **pandas**, proporcionando operaciones geoespaciales en pandas y una interfaz de alto nivel para múltiples geometrías.
 - Instalación: pip install GeoPandas
 - <http://geopandas.org/>
 - **PySAL** : una biblioteca de funciones de análisis espacial escritas en Python destinadas a soportar el desarrollo de aplicaciones de alto nivel.
 - Instalacion: pip install pysal
 - **Missingno** : Módulo de visualización de datos perdidos para Python
 - Instalación: pip install missingno



■ Campos en los que se aplica Python.

■ En Machine Learning

- Tal vez es el campo en el que Python se ha hecho más famoso.
- La librería por excelencia es
 - **Scikit-Learn:**
 - Incluye varios algoritmos de clasificación, regresión y análisis de grupos entre los cuales están máquinas de vectores de soporte, bosques aleatorios, Gradient boosting, K-means y DBSCAN.
 - Está diseñada para interoperar con las bibliotecas numéricas y científicas NumPy y SciPy.
 - Permite en una sola línea definir los conjuntos train y test
 - En otra línea el entrenamiento del modelo
 - En otra línea la aplicación del modelo a nuevos datos
 - En otra línea su evaluación.
 - Además puede deducir las características principales a utilizar, etc.
- Instalación: `pip install sklearn`

<https://scikit-learn.org/stable/>



■ Campos en los que se aplica Python.

■ En Deep Learning

- Deep learning es un término vinculado directamente a la utilización de Redes neuronales.
- Cuando hablamos de estas redes, surgen tres grandes librerías utilizadas en python:
 - TensorFlow
 - Keras
 - Pytorch
- Las tres hacen lo mismo, pero se diferencia en ciertos rasgos:
 - TensorFlow es la más potente y estándar
 - Keras es más sencilla de manejar
 - Pytorch intenta cubrir ambos aspectos, aunque aún está por debajo en uso de las dos anteriores.
- A estas tres grandes librerías se unen otras más pequeñas y que son particulares de cada problema a tratar en Deep Learning.



PYTHON: UN LENGUAJE DEL FUTURO

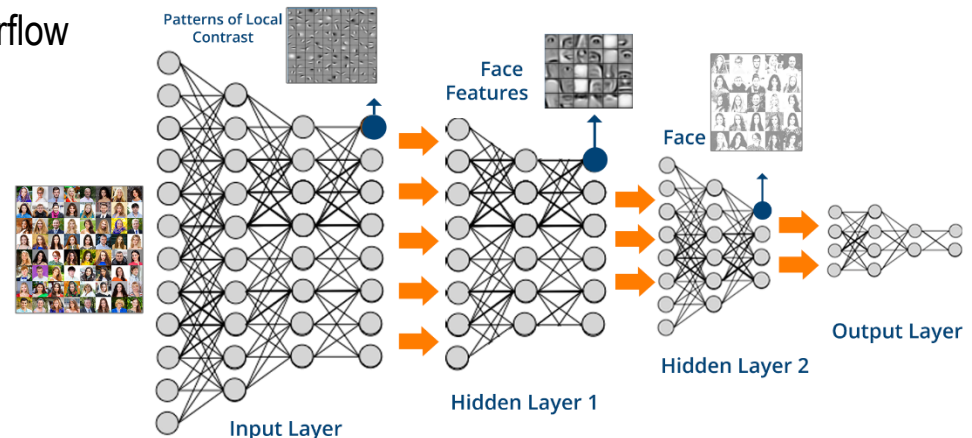
■ Campos en los que se aplica Python.

■ En Deep Learning

• Tensorflow:

- Es un framework muy utilizado en el aprendizaje profundo y automático.
- Fue inicialmente desarrollado en Google Brain. Liberado en 2015
- Entre las aplicaciones TensorFlow más populares se encuentran la identificación de objetos, el reconocimiento de voz y más.
- También hay diferentes ayudantes de capa sobre TensorFlow, como tflearn, tf-slim, skflow, etc..
- Permite aceleración de procesamiento utilizando GPUs.
- En 2019 se va a lanzar TensorFlow 2.0 con mejoras, entre otras, en la sencillez de uso.
- Instalación: `pip install tensorflow`

■ <https://www.tensorflow.org/>





■ Campos en los que se aplica Python.

■ En Deep Learning

• Keras:

- Es una biblioteca de Redes Neuronales de Código Abierto escrita en Python. Es capaz de ejecutarse sobre TensorFlow, Microsoft Cognitive Toolkit o Theano.1.
- Está especialmente diseñada para posibilitar la experimentación en más o menos poco tiempo con redes de Aprendizaje Profundo. Sus fuertes se centran en ser amigable para el usuario, modular y extensible.
- Contiene varias implementaciones de los bloques constructivos de las redes neuronales como por ejemplo los layers, funciones objetivo, funciones de activación, optimizadores matemáticos
- Instalación: `pip install keras`

<https://keras.io/>

• Pytorch:

- Es una biblioteca de aprendizaje automático para el lenguaje de programación Python , basado en la biblioteca Torch
- Utilizado para aplicaciones como el procesamiento de lenguaje natural .
- Se desarrolla principalmente por Facebook
- Instalación: `pip install pytorch`

<https://pytorch.org/>



■ Campos en los que se aplica Python.

■ En Deep Learning

- Otras librerías más específicas:
- Computer Vision – Reconocimiento Facial

■ **OpenCV:**

- OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión de computadora y de aprendizaje automático de código abierto.
- OpenCV fue construido para proporcionar una infraestructura común para las aplicaciones de visión artificial y para acelerar el uso de la percepción de la máquina en los productos comerciales.
- La biblioteca cuenta con más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de aprendizaje por ordenador y de aprendizaje por ordenador tanto clásicos como de vanguardia.
- Estos algoritmos se pueden detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D desde cámaras estéreo, unir imágenes para producir una alta resolución imagen de una escena completa, encuentre imágenes similares de una base de datos de imágenes, elimine los ojos rojos de las imágenes tomadas con flash, siga los movimientos de los ojos, reconozca paisajes y establezca marcadores para superponerlos con realidad aumentada, etc..
- <https://opencv.org/>



PYTHON: UN LENGUAJE DEL FUTURO

■ Campos en los que se aplica Python.

■ En Deep Learning

- Otras librerías más específicas:
- Procesamiento del lenguaje / del Habla
- NLTK
 - NLTK es un conjunto de bibliotecas, una plataforma completa para el procesamiento de lenguaje natural.
 - Puede procesar y analizar texto de diversas maneras, personalizarlo y etiquetarlo, extraer información, etc..
 - También se utiliza para crear prototipos y construir sistemas de investigación.
 - Instalación: `pip install nltk`

<https://www.nltk.org/>

■ SpaCy

- Es una biblioteca de procesamiento de lenguaje natural con excelentes ejemplos, documentación de API y aplicaciones de demostración.
- La biblioteca está escrita en el lenguaje Cython, que es la extensión C de Python.
- Es compatible con casi 30 idiomas, proporciona una fácil integración de aprendizaje profundo y promete robustez y alta precisión.
- Instalación: `pip install spacy`

<https://spacy.io/>



■ Campos en los que se aplica Python.

■ En Deep Learning

- Otras librerías más específicas:
- Procesamiento del lenguaje / del Habla
- Gensim
 - Gensim es una biblioteca de Python para el análisis semántico robusto, el modelado de temas y el modelado de espacios vectoriales, y se basa en **Numpy** y **Scipy**.
 - Proporciona una implementación de algoritmos de PNL populares, como word2vec.
 - Aunque gensim tiene su propia implementación de models.wrappers.fasttext, la biblioteca de texto rápido también se puede usar para un aprendizaje eficiente de representaciones de palabras.
 - Instalación: `pip install gensim`

<https://radimrehurek.com/gensim/>



■ Campos en los que se aplica Python.

■ En Data Science

- Análisis Redes sociales
 - En este tipo de análisis se suelen usar librerías como la ya descrita **Request** con su submódulo **Oauth**. A través de este módulo se puede acceder a la API de Twitter, Facebook o cualquier otra red social, y a través de **params** acceder a la información.
 - Una vez que se accede a la información, se puede realizar los análisis necesarios con Pandas, con otras librerías de Machine Learning, etc..
- ETL-Data Wrangling
 - Data Wrangling es la forma anglosajona de decir Tortura de datos.
 - Se refiere a los procesos de modificación para limpieza, adecuación de la información a nuestros propósitos.
 - La librería más utilizada es Pandas:
 - Pandas:
 - Carga ficheros de diversos formatos.
 - Ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.
 - Instalación: `pip install pandas`

<https://pandas.pydata.org/>



■ Campos en los que se aplica Python.

■ En Data Science

• Web –Scrapping

- Este término se refiere a las diversas operaciones a realizar para obtener información de Web.
- Existen dos librerías principales:
 - **Beautiful soup:** De la que ya hemos hablado en otro lugar
 - **Scrapy:**
 - Crea web spiders, realiza web-scrapping, etc..
 - Su uso facilita lo que cuesta en Beautiful soup: Distintos tipos de web, distintas formas de acceso.
 - Instalación: `pip install scrapy`

<https://scrapy.org/>



■ Campos en los que se aplica Python.

■ En Big Data

- En el campo de BigData sólo hay un ganador: pyspark
- **pyspark:**
 - Es la librería, la API, que permite trabajar con Spark en python
 - Para poder realizar toda su tarea, se subdivide en submódulos:
 - ***pyspark.sql module***: Permite utilizar SQL en Spark además de DataFrames.
 - ***pyspark.streaming module***: Permite realizar operaciones Spark en streaming para implementar BigData en tiempo real.
 - ***pyspark.ml package***: Permite crear Pipeline para procesamiento de Machine Learning
 - ***pyspark.mllib package***: Contiene la definición de todos los algoritmos de Machine Learning a aplicar por Spark.

- Instalación: pip install pyspark

<https://spark.apache.org/docs/2.2.0/api/python/pyspark.html>



► Fin de la Sesión