

Introducción a Python

Versión 3.9 (2021)





¿Qué es Python?

- Lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.
- Código legible.
- Es gratuito. Apareció en 1991, liderado por **Guido van Rossum**
- Robusto para el manejo de datos.
- No muy adecuado para proyectos de bajo nivel.



JAVA VERSUS PYTHON

JAVA	PYTHON
A general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible	An interpreted high-level programming language for general-purpose programming
Requires a semicolon at the end of each statement	Does not require a semicolon at the end of each statement
Compiler converts the Java source code into an intermediate code called a bytecode	Interpreter converts the Python source code into the machine code line by line
Compulsory to declare the data types, so it is statically typed	Data types are dynamic, and it is not necessary to declare data types
A pair of curly braces surrounds a block of statements	A block of statement is indented
Faster	Comparatively slower
Difficult to learn	Easier to read, learn and understand

Comparativa con Java

- Aprender a analizar el proyecto y como cada lenguaje suple las necesidades.

Python Enhancement Proposal (PEP 8)

Normas más relevantes:

- Utilizar 4 espacios por nivel de indentación (No utilizar Tab).
- Utilizar máximo 79 caracteres por línea para bloques de código
- Utilizar máximo 72 caracteres por línea para docstrings o comentarios.
- Dejar 2 líneas en blanco sobre la definición de clases y funciones.
 - Para los métodos dentro de las clases se deja una línea en blanco.
- Utilizar líneas en blanco para separar secciones lógicas dentro de una función.
- Codificación UTF-8, no debe ser declarada.

PEP 8

Normas más relevantes:

- Comentarios y docstrings deben estar en ASCII.
 - Excepto: Pruebas sobre formato no-ASCII y nombres de autores.
- Importar un módulo por línea.
- Para bloques de comentarios se utilizar ‘#’ y un espacio en blanco.
- Para comentarios sobre una línea, se deben dejar al menos 2 espacios.

PEP 8

¿Cuándo ignorar las normas de PEP 8?

- Cuando se pierda legibilidad al aplicar una regla.
- Cuando se encuentra un código que no haya seguido las normas desde un principio y no sea de la prioridad aplicar PEP 8.
- Código realizado previo a la creación de la directriz PEP 8.
- Cuando el código debe permanecer en versiones anteriores, las cuales no soportan algunas de las guías de estilo.

Programa Hola mundo

Python

```
print ('¡Hola mundo!')
```

Java

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("¡Hola mundo!");  
    }  
}
```

Entornos virtuales de Python

<https://www.youtube.com/watch?v=4jt9JPolDpY>

Configurar entornos virtuales

El entorno virtual se debe de crear dentro del directorio raíz del proyecto de python. Sin embargo, los archivos del proyecto no deben estar dentro del entorno virtual.

No se debe hacer commit al entorno virtual. Solo se debe agregar el archivo requirements.txt al repositorio.

https://www.youtube.com/watch?v=Kg1Yvry_Ydk

Python Virtual environments

Algunas herramientas son:

- venv.
- virtualenv.
- pipenv.

Configurar entornos virtuales (Mac y Linux)

Lista de comandos importantes

Instalar venv	<code>sudo apt-get install python3-venv</code>
Crear un entorno virtual.	<code>python3 -m venv <i>[environment_dir]</i></code>
Crear un entorno virtual con acceso a los paquetes del sistema.	<code>python3 -m venv <i>[environment_dir]</i> --system-site-packages</code>
Activar entorno virtual.	<code>source <i>[environment_dir]/bin/activate</i></code>
Desactivar entorno virtual.	<code>deactivate</code>
Mostrar lista de paquetes instalados.	<code>pip list</code>
Mostrar lista de paquetes instalados (Formato para exportar).	<code>pip freeze</code>

Configurar entornos virtuales (Mac y Linux)

Lista de comandos importantes

Lista de comandos importantes	
Exportar lista de paquetes instalados.	<code>pip freeze > [requirements_file].txt</code>
Instalar paquetes desde una lista de requerimientos.	<code>pip install -r [requirements_file].txt</code>
Eliminar un entorno virtual.	<code>rm -rf [environment_dir]</code>



Operadores binarios

Operadores binarios	
Conjunción (AND)	&
Disyunción (OR)	
Disyunción exclusiva (XOR)	^
Negación (NOT)	~
Desplazamiento de bits a la izquierda (Shift left)	<<
Desplazamiento de bits a la derecha (Shift right)	>>



Tipos de datos

Valores lógicos

Verdadero	<i>True</i>
Falso	<i>False</i>

Operadores lógicos

Conjunción	<i>and</i>
Disyunción	<i>or</i>
Negación	<i>not</i>

Operadores de comparación

Igual que	<i>==</i>
Distinto de	<i>!=</i>
Menor que	<i><</i>
Menor o igual que	<i><=</i>
Mayor que	<i>></i>
Mayor o igual que	<i>>=</i>

Datos

Enteros / Flotantes	<i>Booleanos</i>
Strings	<i>Conjuntos</i>



Operadores aritméticos

Operadores aritméticos	
Exponenciación	**
Multiplicación	*
División	/
Módulo (residuo)	%
Suma	+
Resta	-



Variables

El nombre de la variable se le conoce como *identificador*.

identificador = *valor*

Ejemplo:

x = 3

nombre = 'Juan'

finalizado = True



Palabras reservadas

Lista de palabras reservadas en Python				
and	del	for	is	raise
assert	elif	from	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	yield
def	finally	in	print	



Listas, Tuplas y Diccionarios

Sintaxis

Listas

Similares a los arreglos, más su tamaño es dinámico, por lo que permiten ir guardando valores sin límite.

Tuplas

Permiten guardar datos por fila/columna.

Son inmutables, es decir son estáticas.

Diccionarios

Permiten guardar datos, por medio de llaves o keys, permite mapear un valor u objeto en específico.



Funciones

Simples

Tipos

Retorno/Múltiple

Recursivas

Ejemplo:

```
def nombre_usuario(nombre):  
    print("¡Bienvenido! ", nombre)
```

#Llamado de la función.

```
nombre_usuario("Hatake Kakashi")
```

Funciones “de Fábrica”

El intérprete de Python incluye varias funciones “incluidas de fábrica” las cuales están siempre disponibles:

Built-in Functions				
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	apply()
delattr()	help()	next()	setattr()	buffer()
dict()	hex()	object()	slice()	coerce()
dir()	id()	oct()	sorted()	intern()



Elementos dentro de funciones

Sintaxis

Estructuras Selectivas

Se tiene:

```
** if.
```

```
** else.
```

```
** elif (else if).
```

Ciclos/Bucles

Usa:

```
** while
```

```
** for.
```

```
** NO usa do while.
```

POO: Clases/Objetos

Ejemplo:

```
class Ninja():
```

```
    """Atributos"""
```

```
    pais = "Konoha"
```

```
    nombre = "Tobirama"
```

Desarrollo de GUI con python

Algunas bibliotecas para el desarrollo de interfaces de usuario son:

- Tkinter (Integrada).
- PyQt.
- wxPython.
- PyGTK.

Docstrings

Los docstrings se utilizan para describir lo que hace un módulo, una clase o función.

Estos comentarios deben estar después de la línea de definición ('def').

Se utilizan tres-comillas dobles `"""`, al principio y `"""` al final.

PyPI

Python Package Index

- Entorno con módulos y paquetes de utilidad. Para encontrar algún módulo o paquete que cubra una cierta necesidad, puedes consultar la lista de PyPI, con más de 4000 herramientas para automatizar alguna solución.

PyPy

- PyPy es un intérprete y compilador JIT (en tiempo de ejecución) para el lenguaje Python, que se enfoca en la velocidad y eficiencia, que son los puntos flojos más señalados en este lenguaje.

Referencias

- Infografía Comparativa Python-Java: Tomado de:
<https://es.sawakinome.com/articles/technology/what-is-the-difference-between-java-and-python.html>
- González Duque, Raúl. (Marzo, 2011). Python para todos. España: Creative Commons Reconocimiento 2.5. Disponible en:
<http://mundogeek.net/tutorial-python/>
- W3Schools. (2020). Python. Estados Unidos: W3Schools. Ver en:
<https://www.w3schools.com/python/default.asp>