



Actualizado Febrero 2021

# Introducción a Python

Alejandro Ortiz Vega  
High Touch Engineer  
Febrero 2021

# Agenda

- ¿Que es Python?
- Estructuras de datos
- Listas
- Diccionarios y sets
- Condicionales if/else
- Bucles for/while
- Funciones

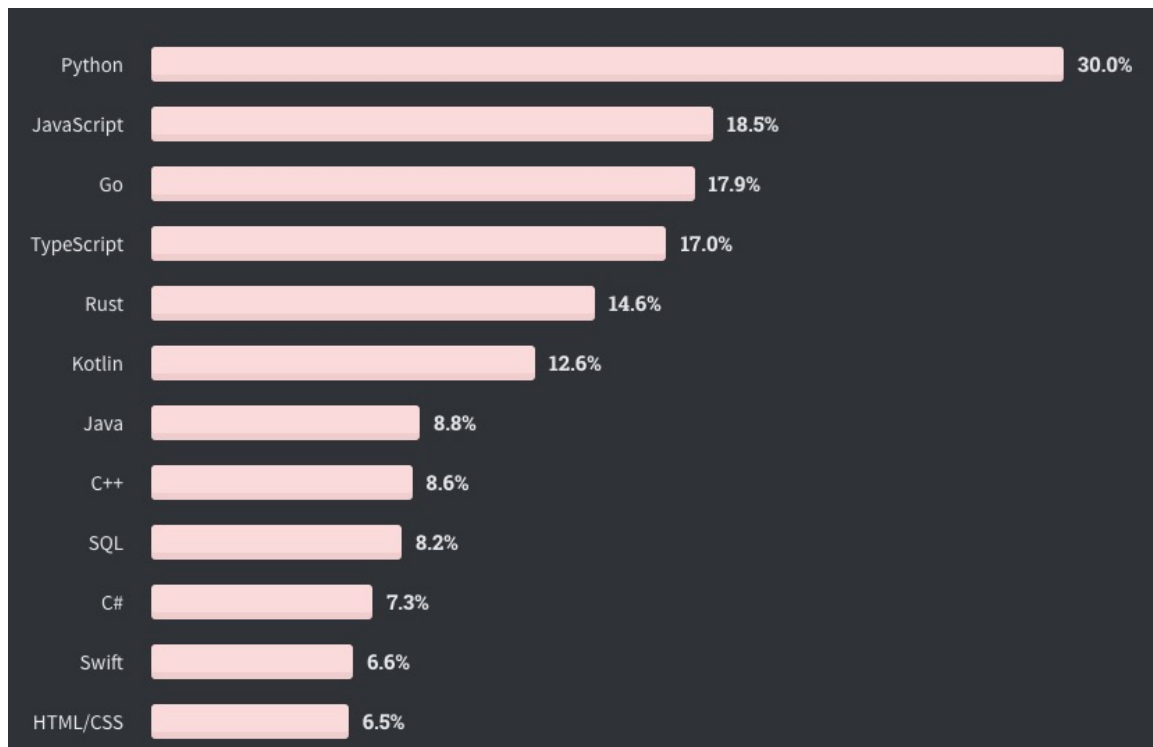


# ¿Que es Python?

Es un lenguaje de programación interpretado de propósito general, orientado a objetos.



# ¿Quiénes quieren aprender Python?



<https://insights.stackoverflow.com/survey/2020>

# Python vs C++

Code	Python has less lines of code.	C++ tends to have large lines of code.
Garbage Collection	Python supports garbage collection	C++ doesn't support garbage collection.
Syntax	Python is easy to remember almost similar to human language.	C++ has a stiff learning curve as it has lots of predefined syntaxes and structure
Compilation	Python uses interpreter.	C++ is pre; compiled.
Speed	Python is slower since it uses interpreter and also determines the data type at run time.	C++ is faster in speed as compared to python.
Efficiency	Easier to maintain, object-oriented and simpler to use	Less clean and manageable in comparison to python
Nature	Python is dynamically typed.	C++ is statically typed.

# Python Input, Output (I/O)

- **Input:**

```
nombre = input()
```

- **Output:**

```
print("Mi nombre es"+nombre)
```

# Tipos de variables

- Numbers -> se utilizan para todo tipo de números enteros, decimales etc.
- String -> se utilizan para guardar cadenas de caracteres y texto.
- List -> es una secuencia ordenada de objetos.
- Tuple -> Las tuplas son objetos de tipo *secuencia*, específicamente es un tipo de dato inmutable.
- Dictionary -> Un objeto *mapping* mapea valores *hashable* a objetos arbitrariamente.

<https://docs.python.org/3/tutorial/datastructures.html>

# Operadores aritméticos

Operador	Caracter
Suma	$a + b = 30$
Resta	$a - b = -10$
Multiplicación	$a * b = 200$
División	$b / a = 2$
Modulo	$b \% a = 0$
Exponente	$a ** b = 100$
División entera	$9 // 2 = 4$



# Operadores de comparación

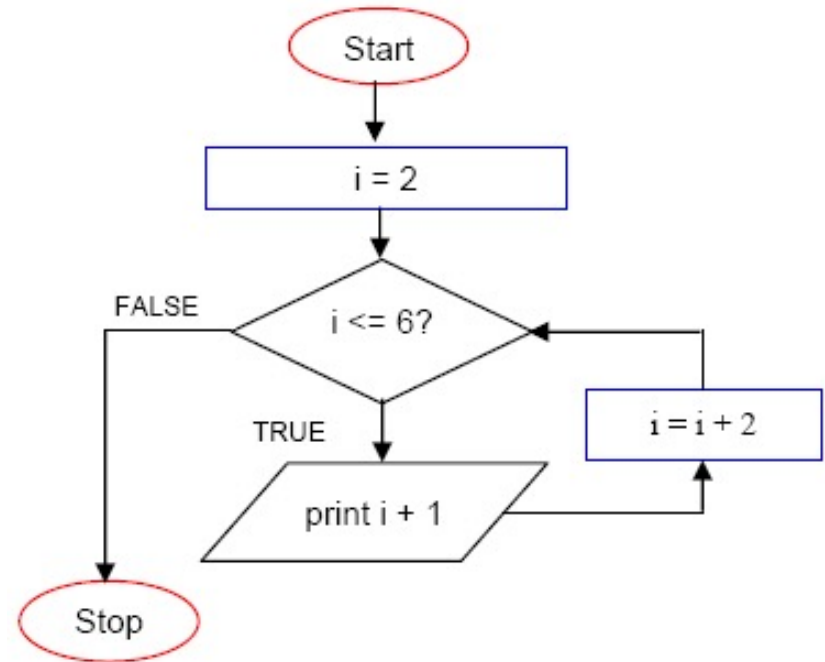
Operador	Caracter
Igual	==
Diferente	!=
Mayor que	>
Menor que	<
Mayor igual que	>=
Menor igual que	<=

# Ejercicio

- Escribir un programa que pregunte al usuario su edad y muestre por pantalla si es mayor de edad o no.

# Bucles en Python

- Los bucles "For" iteran sobre una secuencia que esta definida.
- Un bucle "while" permite repetir la ejecución de un grupo de instrucciones mientras se cumpla una condición.



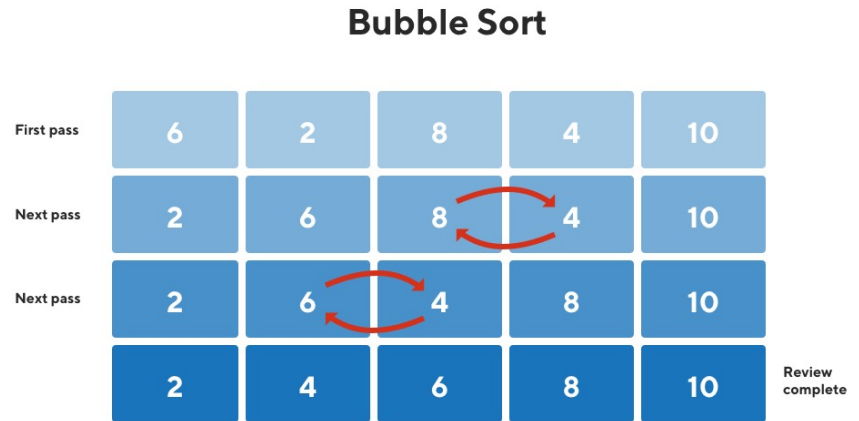
# Ejercicio utilizando While

- Crear un programa que imprima los números primos

0									
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

# Ejercicio utilizando For

- Crear un programa que ordene una lista



# Funciones en Python

Una función, es la forma de agrupar expresiones y sentencias (algoritmos) que realicen determinadas acciones, pero que éstas, solo se ejecuten cuando son llamadas.

Nombre de la función

Argumentos

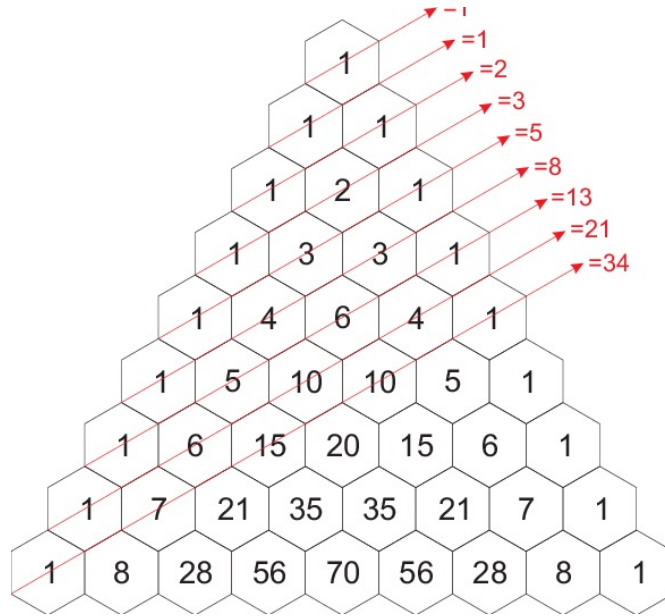
```
def suma(a,b):  
    resultado = a+b  
    return resultado  
  
print(suma(5,6))
```

Invocación

Valor de retorno

# Ejercicio

- Crear una función que retorne un arreglo con los números de Fibonacci



# Python import module

- A module is a file containing Python definitions and statements. The file name is the module name with the suffix .py appended
- Executing modules as scripts: `python fibo.py <arguments>`

**import sys**

```
if __name__ == "__main__":  
    print(sys.argv[1])
```

<https://docs.python.org/3/tutorial/modules.html>



# Python packages

Packages are a way of structuring Python's module namespace by using “dotted module names”.



```
sound/  
    __init__.py  
    formats/  
        __init__.py  
        wavread.py  
        wavwrite.py  
        aiffread.py  
        aiffwrite.py  
        auread.py  
        auwrite.py  
        ...  
    effects/  
        __init__.py  
        echo.py  
        surround.py  
        reverse.py  
        ...  
    filters/  
        __init__.py  
        equalizer.py  
        vocoder.py  
        karaoke.py  
        ...
```

Top-level **package**  
Initialize the sound **package**  
Sub**package** for file format conversion

Sub**package** for sound effects

Sub**package** for filters

# Leer y escribir archivos usando Python

- Write Operations

- Write()
- Writelines()



## Modes

- r = read
- w = write
- a = access

Seek(), close(), with:

- Read Operations

- Read()
- Readline()
- Readlines()



# Formato JSON



**{JSON}**  
JavaScript Object Notation

```
[
  {
    "description": "quarter",
    "mode": "REQUIRED",
    "name": "qtr",
    "type": "STRING"
  },
  {
    "description": "sales representative",
    "mode": "NULLABLE",
    "name": "rep",
    "type": "STRING"
  },
  {
    "description": "total sales",
    "mode": "NULLABLE",
    "name": "sales",
    "type": "INTEGER"
  }
]
```

# Manejo de errores y excepciones

- Syntax Errors

```
>>> while True print('Hello world')
File "<stdin>", line 1
    while True print('Hello world')
                ^
SyntaxError: invalid syntax
```

- Exceptions

```
>>> 10 * (1/0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
>>> 4 + spam*3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'spam' is not defined
>>> '2' + 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
```

- Handling Exceptions

```
>>> while True:
...     try:
...         x = int(input("Please enter a number: "))
...         break
...     except ValueError:
...         print("Oops! That was no valid number. Try again...")
... 
```



[https://github.com/alejoortiz/tutorial\\_python](https://github.com/alejoortiz/tutorial_python)





*Intelligence is the ability to avoid doing work, yet getting the work done.*  
*Linus Torvalds*