

Taller de Python



Profesora Teresa Tapia Soto



Contenidos

- ❖ Python
- ❖ Primer Programa
- ❖ Tipos de Datos y Operadores
- ❖ Variables
- ❖ Funciones de Input/Output

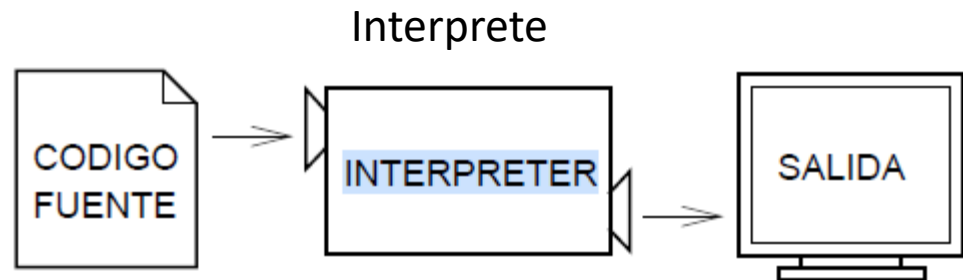


Phyton

Definición: *Python* es un “lenguaje de programación de uso extendido, alto nivel y con múltiples fines”.

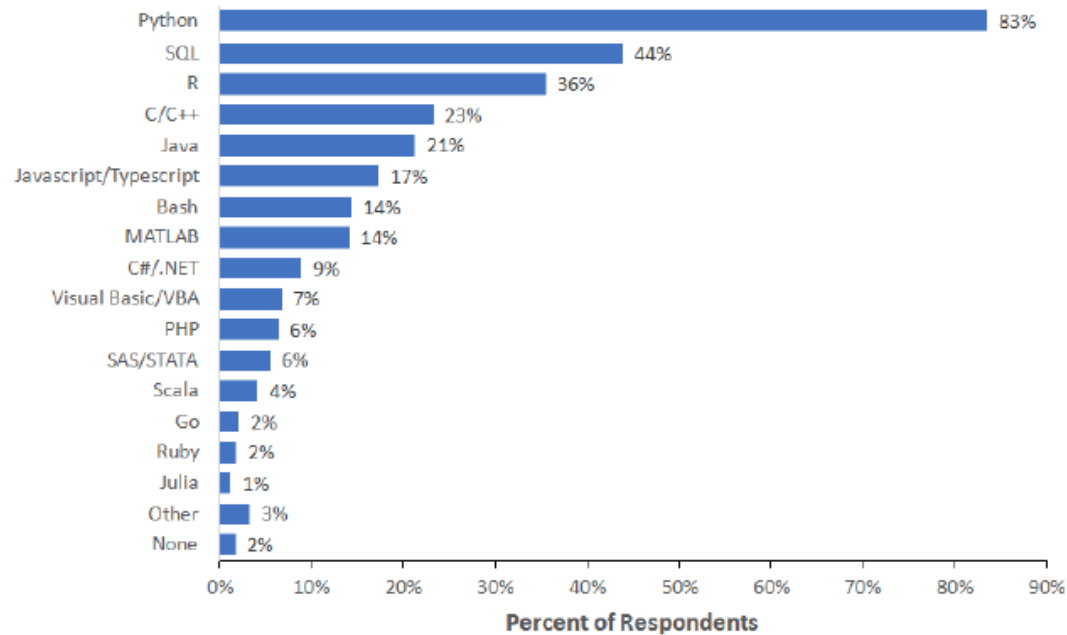
Actualmente ya vamos Python 3.0

- Sintaxis fácil de entender
- Código más organizado
- Hacer uso de la indentación
- Es de código abierto
- Multiplataforma
- Comunidad extensa



Phyton

What programming language do you use on a regular basis?



Note: Data are from the 2018 Kaggle Machine Learning and Data Science Survey. You can learn more about the study here: <http://www.kaggle.com/kaggle/kaggle-survey-2018>. A total of 18827 respondents answered the question.



Copyright 2019 Business Over Broadway



Phyton

Language Types (click to hide)



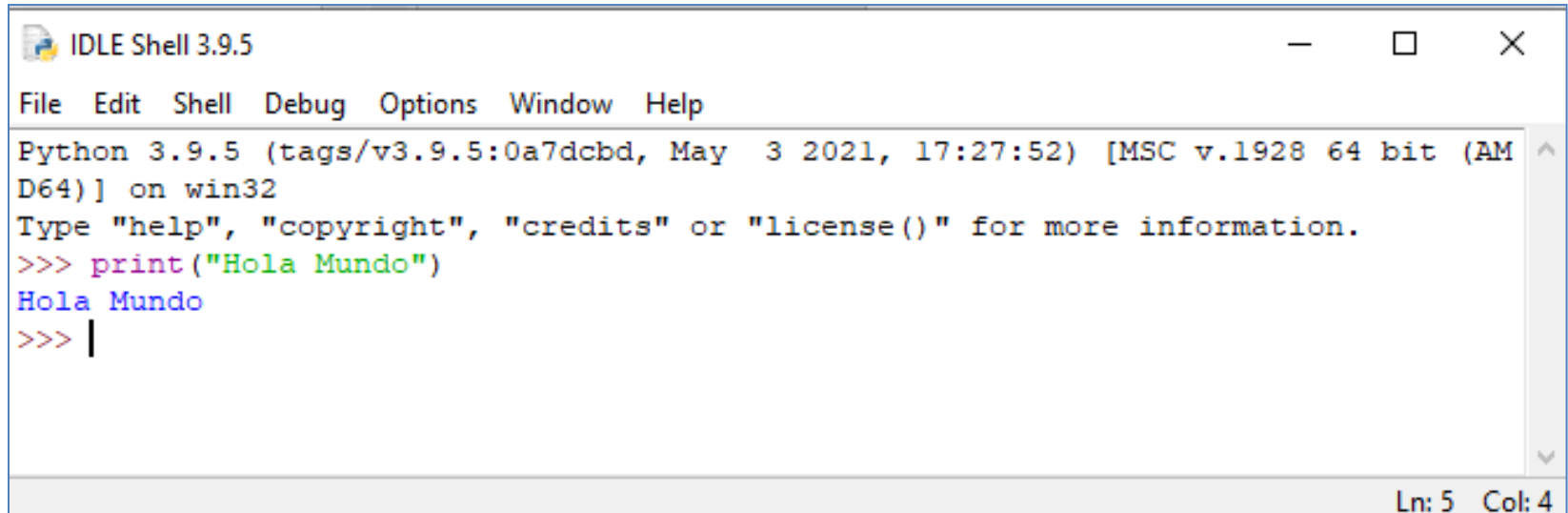
Language Rank	Types	Spectrum Ranking
1. Python	Web, Enterprise	100.0
2. C	Mobile, Enterprise, Embedded	99.7
3. Java	Web, Mobile, Enterprise	99.4
4. C++	Mobile, Enterprise, Embedded	97.2
5. C#	Web, Mobile, Enterprise	88.6
6. R	Enterprise	88.1
7. JavaScript	Web, Mobile	85.5
8. PHP	Web	81.4
9. Go	Web, Enterprise	76.1
10. Swift	Mobile, Enterprise	75.3
11. Arduino	Embedded	73.0
12. Ruby	Web, Enterprise	72.4
13. Assembly	Embedded	72.1
14. Scala	Web, Mobile	68.3
15. Matlab	Enterprise	68.0

<https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>

Nuestro primer Programa!!!

Primer Programa

- ❖ Usaremos IDLE y programaremos Hola Mundo!!!



```
Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hola Mundo")
Hola Mundo
>>> |
```

Ln: 5 Col: 4

- ❖ Agregaremos elementos conocidos:
 - ❖ Comentarios → #

Tipos de Datos



Tipos de Datos

- ❖ ¿Cómo sé que tipo de datos estoy manejando? → `type()`,
- ❖ Los tipos básicos de Python son: **Numéricos**, **Cadenas**, **Booleanos**
 - **Números** – *int, float, complex*
Ejemplos:
 - **int** (entero): `3`
 - **float** (reales; de punto flotante): `15.57`
 - **complex** (complejo*): `7+5j`
 - **Cadenas de texto (strings)** – *str*
Ejemplo:
 - **str**: `"Hola mundo!"`
 - **Valores booleanos** – *bool*
 - **bool**: `True` (verdadero)
 - **bool**: `False` (falso)



Operadores Aritméticos

- ❖ Son aquellos que me permiten realizar operaciones con números o cadenas de texto.

Operador	Descripción	Ejemplo	Resultado
+	Suma	3 + 2	5
-	Resta	4 - 7	-3
-	Inverso aditivo	-7	-7
*	Multiplicación	2 * 6	12
**	Potencia	2 ** 6	64
/	División	3.5 / 2	1.75
//	División entera	3.5 // 2	1.0
%	Módulo	7 % 2	1



Precedencia de Operadores Aritméticos

❖ Precedencia de Operadores.

Recordar el acrónimo **PEMDAS** como regla para recordar el orden de las ejecución de operaciones (siempre de izquierda a derecha):

P → Paréntesis

E → Exponenciación

M, D → Multiplicación y División tienen la misma

A, S → Adición y sustracción

$$\bullet 3 + 5 + 6 = ((3 + 5) + 6)$$

$$\bullet 2 - 5 + 6 = ((2 - 5) + 6)$$

$$\bullet 2 - (5 + 6) = (2 - (5 + 6))$$



Operadores Booleanos

- Los objetos de tipo *bool* son variables que pueden tomar sólo dos valores: **True** o **False**.
- Operadores booleanos

`==, >=, <=, >, <, !=`

- `5 > 2`
- `5 < 2`
- `5 == 5`
- `5 == 6`
- `5 >= 5`
- `5 >= 4`
- `5 <= 5`
- `5 >= 4`
- `5 != 6`
- `5 != 5`
- `5.5 > 5.4`
- `5.1 < 5`

- `True and True`
- `True and False`
- `False and True`
- `False and False`

True
False
False
False

- `True or True`
- `True or False`
- `False or True`
- `False or False`

True
True
True
False

- `not True`
- `not False`

False
True

- `5 > 4 or 6 != 5`
- `5 == 5 and 6 == 7`
- `not (5 != 4)`

True
False
False



Precedencia Operadores Booleanos

- Entre los operadores del tipo **bool**, las precedencias son las siguientes:

Mayor precedencia

- `<, <=, >, >=, !=, ==`
- `not`
- `and`
- `or`

Menor precedencia

- Ejemplo:

- `expr1 and not expr2 or expr3` es equivalente a
- `(expr1 and (not expr2)) or expr3`

Más sobre precedencia

<https://docs.python.org/3/reference/expressions.html#operator-precedence>



Operadores Sobre Cadenas

- Se pueden usar operadores aritméticos

```
>>> "San" + "tiago"
'Santiago'
>>> "Chi"*3
'ChiChiChi'
>>> "Le"*3
'LeLeLe'
>>> "Viva" + " Chile!"
'Viva Chile!'
```

- Se pueden usar operadores de comparación

```
>>> "santiago" == "santiago"
True
>>> "Santiago" == "santiago"
False
```



Variables



Variables

Definición: La *asignación* es el acto de dar valor a una variable.

- El valor de la expresión que está a la derecha del signo =, se asigna a la variable de la izquierda.
- `nombre_variable= expresión`

`a = 3` no es lo mismo que `3 = a`



```
>>> pi = 3.14159265359
>>> r = 1.298373
>>> perimetro = 2 * pi * r
>>> area = pi * r ** 2
>>> area
5.296010335524904
>>> print(area)
5.296010335524904
>>> perimetro
8.157918156839218
>>> print(perimetro)
8.157918156839218
```



Variables

Asignaciones con operadores

- Fijarse que la sentencia `i=i+1` aplica un incremento unitario al contenido de la variable `i`
- En Python se puede resumir de la siguiente forma

```
>>> i = 0
>>> i += 1
>>> i
1
```

Notar que entre el signo `+` y el signo `=` no debe haber espacios

```
Z = 5
Z += 2
Z *= 2
Z /= 2
Z -= 2
Z %= 2
Z **= 2
```

- ¡No nos olvidemos de crear la variable antes de usarla!

```
>>> i += 1
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    i += 1
NameError: name 'i' is not defined
```

Error, ya que no estaba declarada la variable `i`.

Primero debemos declarar la variable y asignarle un valor

Variables

Hay algunas *palabras reservadas* en Python que no pueden ser utilizadas como nombres de variables

```
and, as, assert, break, class, continue, def, del, elif, else,
except, exec, finally, for, from, global, if, import, in,
is, lambda, not, or, pass, print, raise, return, try,
while, yield
```



I/O –Input/Output

- ❖ Ingreso de valores por la entrada estándar.

```
>>>  
>>> a=input("Diga un nombre: ")  
Diga un nombre: Teresa  
>>> |
```

- ❖ Funciones de interés: int(), float(), bool().



Taller de Python



Profesora Teresa Tapia Soto



