

# Python



## Primeros Pasos

### Parte II

Algoritmos y Programación I - FIUBA  
Lic. Gustavo Bianchi

1

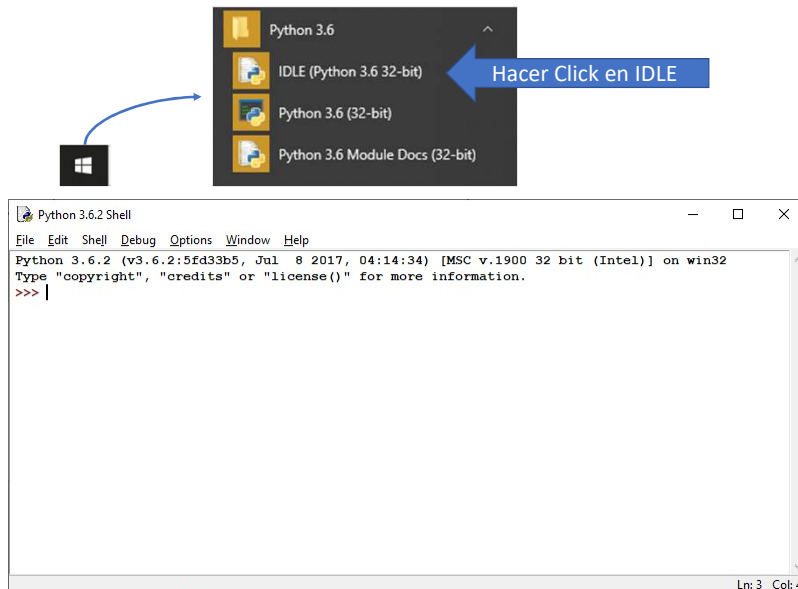
## Algunos elementos del lenguaje

- La Consola de Python
- Definición de Variables
- Mostrar el contenido de una variable
- Ingreso de un dato
- Operadores Aritméticos, de Comparación y Lógicos

Algoritmos y Programación I  
Lic. Gustavo Bianchi

2

## Abriendo la consola de Python



Algoritmos y Programación I  
Lic. Gustavo Bianchi

3

## Definiendo Variables

**Nombre de la Variable = Valor**

↑  
Operador de Asignación

**Vamos a declarar una variable x con un valor inicial 5**

```
>>> x = 5
```

```
>>> x
```

```
5
```

```
>>> type(x)
```

```
<class 'int'>
```

Python, a diferencia de otros lenguajes, no requiere que previamente declaremos el tipo de la variable. Al darle a x el valor 5, le asignará como tipo, **int** (entero)

La función **type**, nos permite corroborar el tipo que le fue asignado a la variable

Algoritmos y Programación I  
Lic. Gustavo Bianchi

4

## Definiendo Variables - CONTINUACION

Ahora a **x** le asignaremos el valor 5.2

```
>>> x = 5.2
```

el tipo de x es **float** (real)

Por último, a **x** le asignaremos la palabra "Python"

```
>>> x = "Python"
```

el tipo de x es **str** (cadena de caracteres)

Acabamos de comprobar la característica mencionada de **tipado dinámico**. Nuestra variable x, fue cambiando de tipo en función del valor que almacenaba.

Algoritmos y Programación I  
Lic. Gustavo Bianchi

5

## Más sobre Variables

Es una buena práctica dar a las **variables** un **nombre representativo de lo que almacenan**, por eso, limitar el uso de las letras a casos que lo justifiquen

```
>>> edad = 25
>>> nombre = "Juan"
```

Que sucederá si  
hacemos esto?

```
>>> nombre + edad
```

**ERROR**

Al ser **fuertemente tipado**, no es posible realizar operaciones con variables de distinto tipo, sin efectuar la conversión que corresponda.

Entonces, qué debemos hacer?

```
>>> nombre + str(edad)
```

**CORRECTO!**

Conversión del tipo **int** al tipo **str**

Algoritmos y Programación I  
Lic. Gustavo Bianchi

6

## Cómo muestro el valor de una variable?

```
>>> print(edad)
```

```
25
```

```
>>> print(nombre)
```

```
Juan
```

```
>>> print(nombre, edad)
```

```
Juan 25
```

```
>>> help(print) →
```

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
```

Algoritmos y Programación I  
Lic. Gustavo Bianchi

7

## Cómo muestro un mensaje?

```
>>> print("Hola Mundo")
```

```
Hola Mundo
```

La cadena va entre comillas  
dobles o simples

Y si quiero combinar mensajes con valores de variables?

```
>>> print("Hola", nombre, "tu tienes", edad, "años")
```

```
Hola Juan tu tienes 25 años
```

```
>>> print(
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Algoritmos y Programación I  
Lic. Gustavo Bianchi

8

## Cómo solicito el ingreso de un valor para una variable?

```
>>> edad = input("Edad: ")
Edad: _
```

**ATENCIÓN!** El dato ingresado, siempre será almacenado como string

```
>>> help(input) →
```

**input(prompt=None, /)**

Read a string from standard input. The trailing newline is stripped.

The prompt string, if given, is printed to standard output without a trailing newline before reading input.

If the user hits EOF (\*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError. On \*nix systems, readline is used if available.

Algoritmos y Programación I  
Lic. Gustavo Bianchi

9

## Cómo solicito el ingreso de un valor para una variable?

Dado que la función input siempre devuelve una cadena de caracteres, quizás queramos que el dato ingresado sea tratado en un futuro como un número, para ello, debemos convertirlo al tipo numérico que sea conveniente.

```
>>> edad = int( input("Edad: ") )
```

En este caso, antes de asignar, se **convierte a entero**

Algoritmos y Programación I  
Lic. Gustavo Bianchi

10

## Operadores Aritméticos

Operación	Operador	Ejemplos para x=5 y=4
Suma	+	>>> x + y 9
Resta	-	>>> x - y 1
Multiplicación	*	>>> x * y 20
División Real	/	>>> x / y 1.25
División Entera	//	>>> x // y 1
Resto de la División	%	>>> x % y 1
Potencia	**	>>> x ** y 625

Algoritmos y Programación I  
Lic. Gustavo Bianchi

11

## Operadores de Comparación

Comparaciones		Ejemplos para x=5 y=4
Mayor	>	>>> x > y True
Menor	<	>>> x < y False
Igual	==	>>> x == y False
Distinto	!=	>>> x != y True
Mayor Igual	>=	>>> x >= y True
Menor Igual	<=	>>> y <= x True
Varios		>>> 3 < y <= x < 10 True

Algoritmos y Programación I  
Lic. Gustavo Bianchi

12

## Operadores Lógicos

Conector Lógico		Ejemplos para a=True b=False
Disyunción	<b>or</b>	>>> a <b>or</b> b True
Conjunción	<b>and</b>	>>> a <b>and</b> b False
Negativa	<b>not</b>	>>> <b>not</b> a False

Algoritmos y Programación I  
Lic. Gustavo Bianchi