

# Python - Iniciación

CS1100 - Introducción a Ciencia de la Computación  
UTEC

# ¿Por qué programar en Python?

To Display "Hello World"

"Hello World!" Program in Python

```
print("Hello World!")
```

"Hello World!" Program in C

```
#include <stdio.h>
int main()
{
    printf("Hello World !");
    return 0 ;
}
```

"Hello World!" Program in C++

```
#include <iostream>
using namespace std ;
int main()
{
    cout <<"Hello World !";
    return 0 ;
}
```

"Hello World!" Program in Java

```
public class HelloWorld {
    public static void main(Strings[] args) {
        System.out.println("Hello World !");
    }
}
```

Expresivo



Productivo

# Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Conocer el entorno de python

## Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Conocer el entero de python
- Conocer los tipos de datos en python

## Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Conocer el entero de python
- Conocer los tipos de datos en python
- Conocer la sintaxis de python

# Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Conocer el entero de python
- Conocer los tipos de datos en python
- Conocer la sintaxis de python
- Realizar operaciones fundamentales en python

# Hola Mundo!!!

```
1 >>> print("Hola Mundo !!!")
2 Hola Mundo !!!
```

# Ingreso y salida de datos

## input

- Permite ingresar datos desde teclado

```
1  >>> nombre = input("Ingresa tu nombre:")
2  Ingresa tu nombre: Marvin
3  >>> print("Hola " + nombre)
4  Hola Marvin
```

# Ingreso y salida de datos

## input

- El comando "input" retorna una cadena
- Podemos hacer operaciones con la variable que creamos
- Convertir de cadena a entero usando el constructor int

```
1  >>> t = input("Ingresa algo: ")  
2  Ingresa algo: Hola  
3  >>> 5*t  
4  'HolaHolaHolaHolaHola'  
5  >>> print(5*t)  
6  HolaHolaHolaHolaHola
```

```
1  >>> n = int(input("Ingrese un numero: "))  
2  Ingrese un numero: 7  
3  >>> print(5*7)  
4  35
```

# Concatenación y multilinea

```
1 >>> "primero" "segundo"  
2 'primerosegundo'
```

```
1 >>> """Utec es  
2 ... mi Universidad  
3 ... querida"""  
4 'Utec es\nmi Universidad\nquerida'  
5 >>> '''Yo  
6 ... vivo en  
7 ... Barranco'''  
8 'Yo\nvivo en\nBarranco'  
9 >>> print('Utec es\nmi Universidad\nquerida')  
10 Utec es  
11 mi Universidad  
12 querida  
13 >>> print('Yo\nvivo en\nBarranco')  
14 Yo  
15 vivo en  
16 Barranco
```

# Caracteres especiales

## Backslash

- Para usar comillas simples o dobles dentro de una cadena
- Para caracteres especiales es necesario mandar a imprimir para verlo

```
1  >>> "Esto es un \" en una cadena"
2  'Esto es un " en una cadena'
3  >>> 'Esto es un \' en una cadena'
4  "Esto es un ' en una cadena"
5  >>> s = 'Un \\ en una cadena'
6  >>> s
7  'Un \\ en una cadena'
8  >>> print(s)
9  Un \ en una cadena
```

# Tipos de datos

- Números enteros (int)
- Números de punto flotante (float)
- Números complejos (complex)
- Valores booleanos (bool)
- Cadenas de caracteres (str)

# Tipos de datos

## int

- Es de precisión ilimitada
- Decimal, Binario, Octal, Hexadecimal

```
1      >>> 45
2      45
3      >>> 0b10
4      2
5      >>> 0o10
6      8
7      >>> 0x10
8      16
```

# Tipos de datos

## float

- Números literales, Notación Científica, constructor float

```
1      >>> 3.1415
2      3.1415
3      >>> 15e4
4      150000.0
5      >>> 1.616e-35
6      1.616e-35
7      >>> float(7)
8      7.0
9      >>> float("3.1415")
10     3.1415
```

# Tipos de datos

## complex

- Para hacer operaciones polares

```
1  >>> z = complex(2,-3)
2  >>> print(z)
3  (2-3j)
4  >>> z = complex(1)
5  >>> print(z)
6  (1+0j)
7  >>> z = complex()
8  >>> z
9  0j
10 >>> z = complex("5-9j")
11 >>> print(z)
12 (5-9j)
```

# Tipos de datos

## bool

- Representación lógica
- Muy importante para las estructuras de control condicional y repetitiva

```
1  >>> True
2  True
3  >>> False
4  False
5  >>> bool(0)
6  False
7  >>> bool([])
8  False
9  >>> bool("")
10 False
11 >>> bool(1)
12 True
13 >>> bool(0.123)
14 True
15 >>> bool("hola")
16 True
```

# Tipos de datos

## str

- Es inmutable
- Secuencia de unicode code points

```
1  >>> "Estas son las 'comillas simples' dentro de comillas dobles"
2  "Estas son las 'comillas simples' dentro de comillas dobles"
3  >>> 'Estas son las "comillas dobles" dentro de comillas simples'
4  'Estas son las "comillas simples" dentro de comillas simples'
5  >>> type(str)
6  <class 'type'>
7  >>> help(str)
```

# Ejemplos

## type

- Nos indica el tipo de datos de cada variable

```
1  >>> type(5)
2  <class 'int'>
3  >>> type(3.5)
4  <class 'float'>
5  >>> type(True)
6  <class 'bool'>
7  >>> type([])
8  <class 'list'>
9  >>> type(complex())
10 <class 'complex'>
11 >>> type(None)
12 <class 'NoneType'>
```

# Operadores aritméticos

- Si ambos operandos son enteros, el resultado es entero
- Si un operando es float, el resultado es FLOAT

```
1  >>> 5+2
2  7
3  >>> 5+7.5
4  12.5
5  >>> 5-2
6  3
7  >>> 5.0-2
8  3.0
9  >>> 5.2*10
10 52.0
11 >>> 5*2
12 10
```

# Operaciones con cadenas

- Concatenar
- Obtener el Unicode code point (caracter, letra) de una cadena de texto

```
1  >>> a = "UTEC"
2  >>> b = "life"
3  >>> c = "Hackers"
4  >>> logo = a + " " + b + c
5  >>> print(logo)
6  UTEC lifeHackers
7  >>> logo[0]
8  'U'
9  >>> logo[1]
10 'T'
```

# Operadores aritméticos

## ■ División real, División entera, Residuo, Potencia

```
1      >>> 5/2
2      2.5
3      >>> 5.0/2
4      2.5
5      >>> 5//2
6      2
7      >>> 5%2
8      1
9      >>> 5.0%2
10     1.0
11     >>> 2**5
12     32
13     >>> 2.0**5
14     32.0
```

# Indentación

## Definición

- En Python, la indentación tiene un significado semántico.
- Esto significa que cambiar la indentación de un programa puede cambiar su significado

```
1  >>> nota = 11.1
2  >>> if nota > 10:
3      ...     print("El alumno APRUEBA")
4  ... else:
5      ...     print("El alumno REPRUEBA")
6  ...
7  El alumno APRUEBA
```

# Precedencia de Operadores:

```
1 >>> 3*5 + 1  
2 16  
3 >>> 3*(5+1)  
4 18
```

Prioridad	Operador
1	Exponente: **
2	Mult, Div, Div entera, Mód: *, /, //, %
3	Suma, Resta: +, -

Table: Precedencia de operadores.

## Ejercicio 1

### Enunciado

**Escribir un programa en Python, que permita hallar el área y el volumen de una esfera. Tenga en cuenta que para una esfera:**

- el área es  $A = 4\pi r^2$
- y el volumen es  $V = \frac{4\pi r^3}{3}$

## Ejercicio 2

### Enunciado

**Escribir un programa en Python, que permita ingresar un monto el SOLES y debe mostrar su equivalente el DOLARES y EUROS:**

- dolares=?
- euros=?

## Ejercicio 3

### Enunciado

**Desarrollar un programa en python para cada una de las formulas mostradas en la siguiente imagen**

### Formulas (Movimiento Rectilíneo Uniformemente Acelerado)

$$a = \frac{v_f - v_i}{t}$$

$$t = \frac{v_f - v_i}{a}$$

$$x = \left( \frac{v_f + v_i}{2} \right) * t$$

$$v_f = v_i + (a * t)$$

$$x = (v_i * t) + \frac{a*t^2}{2}$$

$$2 * a * x = v * f^2 - v * t^2$$

Donde:

$a$  = Aceleración ( $\frac{m}{s^2}$ )

$v_f$  = Velocidad final ( $\frac{m}{s}$ )

$v_i$  = Velocidad inicial ( $\frac{m}{s}$ )

$t$  = Tiempo (s)

$x$  = Distancia (m)

# Cierre

En esta sesión aprendiste:

- A conocer los tipos de datos str, int, float, complex, bool

# Cierre

En esta sesión aprendiste:

- A conocer los tipos de datos str, int, float, complex, bool
- Conocer el tipo de datos de una variable usando (type)

# Cierre

En esta sesión aprendiste:

- A conocer los tipos de datos str, int, float, complex, bool
- Conocer el tipo de datos de una variable usando (type)
- Indentación y sintaxis de python

# Cierre

En esta sesión aprendiste:

- A conocer los tipos de datos str, int, float, complex, bool
- Conocer el tipo de datos de una variable usando (type)
- Indentación y sintaxis de python
- Realizar operaciones con los tipos de datos