

Python Científico

Unidad 01 - Introducción a Python

Sebastián Flores

<https://linktr.ee/sebastiandres>

¿Qué contenido aprenderemos?

- Asignaciones y expresiones
- Entrada y salida de datos
- Tipos de datos básicos

¿Porqué aprenderemos ese contenido?

- Interacción con programa requiere ingresar datos, realizar cálculos y regresar una solución.
- Para programar necesitamos conocer **extremadamente bien** los tipos de datos básicos.
- Con la clase de hoy ya podremos hacer programas **útiles** que calculan cosas.

Misión de la clase

Problema:

Escriba un programa que reciba una temperatura en grados Fahrenheit y entregue como resultado el equivalente en grados Celsius.

Caso de prueba

- Temperatura en Fahrenheit: 95
- Temperatura equivalente en Celsius: 35

Tipos de Datos

Enteros

..., -3, -2, -1, 0, 1, 2, 3, ...

Reales

..., -3.14, ..., 0.0, ..., 1E-8, ..., 1.0, ..., 1e8, ...

Lógicos

True, False

Texto

"abracadabra", 'Universidad Técnica Federico Santa Maria', "etcétera", ...

1-Asignación y Expresión

Bloques con los cuales se asignan valores a las variables

1.1 Asignación

- Correspondencia entre una variable y un valor.
- Nombre de variable siempre en la izquierda.
- Valor que toma siempre en la derecha. ### Ejemplos

In []:

```
x = 1
x = 1.0
x = True
x = "True"
```

1.2 Expresión

Combinación de valores y operaciones que al ser evaluados entregan un resultado.

Ejemplos

```
In [2]: (1.0 + 2.0)*3
```

```
Out[2]: 9.0
```

```
In [5]: (100>0) and (1./3<1./2)
```

```
Out[5]: True
```


Asignación

Típicamente la asignación de valores toma 2 formas:

- Inicialización:

```
In [9]: a = 3  
        b = 4
```

- Cálculo de expresión y luego asignación:

```
In [10]: c = (a**2 + b**2)**0.5
```

```
In [11]: peri = a + b + c
```

Asignación

¿Que valores tienen a, b y c al final del programa?

In [12]:

```
a = 1
b = a
c = a + b
a = c
print a, b, c
```

2 1 2

2- Input y Output

- Entrada de datos
- Salida de datos

2.1 Entrada de datos

- Utilizar la función **raw_input(texto)** para imprimir el **texto** en pantalla y esperar que el usuario ingrese un valor.
- La función **raw_input** siempre devuelve un string.
- Para poder utilizar el valor es necesario convertir al tipo de dato requerido.

2.1 Entrada de datos

- Utilizar la función **raw_input(texto)** para imprimir el **texto** en pantalla y esperar que el usuario ingrese un valor.
- La función **raw_input** siempre devuelve un string.
- Para poder utilizar el valor es necesario convertir al tipo de dato requerido.

In [13]:

```
nombre = raw_input("Ingrese su nombre: ")
edad = int(raw_input("Ingrese su edad [años]: "))
altura = float(raw_input("Ingrese su peso [kg]: "))
```

```
Ingrese su nombre: felipe kanzua
Ingrese su edad [años]: 18
Ingrese su peso [kg]: 1.8
```

2.2 Salida de datos

- Para imprimir valores en pantalla utilizamos la función **print**.
- La función **print** acepta cualquier tipo básico
- Es posible incluso encadenar datos conocidos utilizando la coma

In [14]:

```
print nombre
print edad
print altura
print nombre, edad, altura
print "Hola", nombre, "tienes", edad, "años y mides", altura , "metros"
```

```
felipe kanzua
```

```
18
```

```
1.8
```

```
felipe kanzua 18 1.8
```

```
Hola felipe kanzua tienes 18 años y mides 1.8 metros
```

Ejemplo con entrada y salida de datos

Preguntar el lado de un cuadrado y calcular el perímetro.

In [19]:

```
# Entrada de datos
a = float(raw_input("Ingrese el lado de un cuadrado: "))
# Cálculo de la respuesta
perimetro = a*4
# Salida de datos
print "El cuadrado de lado ", a, "tiene perimetro", perimetro
```

Ingrese el lado de un cuadrado: four

```
-----
-
ValueError                                Traceback (most recent call last)
<ipython-input-19-alf577144390> in <module>()
      1 # Entrada de datos
----> 2 a = float(raw_input("Ingrese el lado de un cuadrado: "))
      3 # Cálculo de la respuesta
      4 perimetro = a*4
      5 # Salida de datos

ValueError: could not convert string to float: four
```

3- Tipos de Datos

- Enteros
- Flotantes
- Booleanos
- Cadenas de Texto

3.1- Tipo de dato entero

- Viene del inglés **integer**
- Se abrevia como **int**
- Ejemplos

In [20]: 42

Out[20]: 42

```
In [27]: int(1.237237243E6)
```

Out[27]: 1237237

```
In [22]: 10000000000000000000000000000000000000000000000000000000
```

```
Out[22]: 10000000000000000000000000000000000000000000000000000L
```

3.1 Int

Operaciones en datos enteros

Operaciones aritméticas: **siempre regresan un entero**

- Suma: +
- Resta: -
- Multiplicación: *
- División: /
- Potenciación: **
- Módulo: %

In [28]:

```
print 3+7
print 1000000000 - 1000
print 10*10
print 3/2
print 2**2
print 19%2
```

```
10
999999000
100
1
4
1
```

3.2 Tipo de dato real

- Se llama real o flotante
- En inglés, **float**
- En python: **float**

```
In [29]: 231.45
```

```
Out[29]: 231.45
```

```
In [33]: float(3/2)
```

```
Out[33]: 1.0
```

3.2 Float

Operaciones en datos reales

Operaciones aritméticas: siempre regresan un real

- Suma: +
- Resta: -
- Multiplicación: *
- División: /
- Potenciación: **
- Módulo: %

In [34]:

```
print 10.0 + 1e2  
print 3.14 - 1e-1  
print 2.0*1e2  
print 1e-3/0.01  
print 1.1e0**2.0  
print 1.23e1%2.0
```

```
110.0  
3.04  
200.0  
0.1  
1.21  
0.3
```

OBSERVACION

Operaciones aritméticas entre int y float son posibles, pero el resultado siempre será un **float**.

In [35]:

```
print 1 + 1.0, 1.0 + 1
print 1 - 1.0, 1.0 - 1
print 2*3.0, 2.0*3
print 1./2, 1/2.
print 2**3.0, 2.0**3
print 15%2.0, 15.0%2
```

```
2.0 2.0
0.0 0.0
6.0 6.0
0.5 0.5
8.0 8.0
1.0 1.0
```

3-Tipos de datos lógico

- También se le llama booleano, del inglés **boolean**
- Se abrevia como **bool**
- Sólo hay 2 tipos: **True** y **False**

3- Bool

Operaciones para datos booleanos

- **Negación** : not
- **y**: and
- **o** : or

In [36]:

```
print True and True  
print True and False  
print False and True  
print False and False
```

```
True  
False  
False  
False
```

In [37]:

```
print True or True  
print True or False  
print False or True  
print False or False
```

```
True  
True  
True  
False
```

```
In [38]: print not True  
print not False
```

False

True

Operaciones en datos enteros

Comparaciones de reales o enteros: siempre regresan un booleano

- menor o igual : <=
- menor : <
- igual : =
- distinto : !=
- mayor : >
- mayor o igual : >=

In [47]:

```
a = 10.5  
print 5<=a<20  
print int(a)==int(10.9999999999999999999999999999999999)  
print a!=1E3  
print a>0 and a**2<200  
print a<0 or a>0
```

True
False
True
True
True

Tipo de dato texto

- Texto (también llamado cadena de texto)
- En inglés: **string**
- En python: **str**

```
In [ ]: "Hola clase"
```

```
In [ ]: 'Hola mundo'
```

```
In [ ]: '''Hmm'''
```

```
In [ ]: """Hey Jude"""
```

Operaciones en cadenas de texto

- Concatenación: `str_a + str_b`
 - Regresa un nuevo string concatenando las expresiones `str_a` y `str_b`, en ese orden.
- Auto-concatenación: `str * n`
 - Regresa un nuevo string, conteniendo `n` veces el string `str`.
- Número de caracteres: `len(str)`
 - Regresa un entero ≥ 0 , con el número de caracteres del string `str`.
- contención: `str_a in str_b`
 - Regresa un bool evaluando si el string `str_a` está completamente contenido en string `str_b`

In [52]:

```
print "hola" + ' ' + "mundo"  
print "waka"*2 + "eeo"  
print len('paralelepipedo')  
print "pollo" in "repollo"  
print "bcdo" in "abcdario"
```

```
hola mundo  
wakawakaeeo  
14  
True  
False
```

Funciones

- Valor absoluto: **abs**(x)
 - Regresa el valor absoluto de x (entero o flotante)
- Mínimo: **min**(x,y)
 - Regresa el menor valor entre x e y (enteros o flotantes).
- Máximo: **max**(x,y)
 - Regresa el mayor valor entre x e y (enteros o flotantes).
- Redondear: **round**(x)
 - Regresa el flotante correspondiente al entero más cercano a x (flotante o entero).
- type: **type**(x):
 - Regresa el tipo de dato de x (cualquier cosa).

In [53]:

```
# Valor absoluto, para int o float
print abs(-4-5), abs(-4.0-5)
print min(-1,20), min(3.0, 200), min(10,10.0)
print max(-1,20), max(1.0, 200), max(10,10.0)
print round(1.49), round(1.50), round(1.51)
print round(-1.49), round(-1.50), round(-1.51)
```

```
9 9.0
-1 3.0 10
20 200 10
1.0 2.0 2.0
-1.0 -2.0 -2.0
```

Preguntas

¿Que regresa cada una de las expresiones siguientes?

In [55]:

```
print type(1)
print type(1.0)
print type(True)
print type("True")
print type(float(1))
print type(type(1))
print type(float(int(bool(str(int(float(int(str(0))))))))))
print "Hola\nmundo\n\tcruel"
```

```
<type 'int'>
<type 'float'>
<type 'bool'>
<type 'str'>
<type 'float'>
<type 'type'>
<type 'float'>
Hola
mundo
    cruel
```

Problema de prueba

Problema:

Escriba un programa que reciba una temperatura en grados Fahrenheit y entregue como resultado el equivalente en grados Celsius.

Fórmula de conversión

Temperatura en Celsius = (Temperatura en Fahrenheit - 32) * 5 / 9

Caso de prueba

- Temp. en Fahrenheit: 95
- El equivalente en Celsius es: 35.0

Análisis del caso de prueba

- Dato de entrada o Input:
 - 95 (int o float)
- Dato de salida o Output:
 - String con el resultado de la conversión del input

Análisis del caso general

- Dato de entrada o Input: T_F
- Conversión a Celsius:

$$T_C = (5 / 9) * (T_F - 32)$$

O

$$T_C = (5.0 / 9.0) * (T_F - 32)$$

- Dato de salida o Output: String con el resultado de la conversión del input

Implementaciones Incorrectas

In [57]:

```
# Programa de Conversión de Temperatura INCORRECTO
f = float(raw_input('Ingrese temperatura en grados Fahrenheit: '))
c = 5 / 9 * (f - 32)
print 'El equivalente en grados Celsius es: ', c
```

```
Ingrese temperatura en grados Fahrenheit: 30
El equivalente en grados Celsius es: -0.0
```

In [58]:

```
# Programa de Conversión de Temperatura INCORRECTO
f = float(raw_input('Ingrese temperatura en grados Fahrenheit: '))
c = (f - 32) * ( 5 / 9 )
print 'El equivalente en grados Celsius es: ', c
```

```
Ingrese temperatura en grados Fahrenheit: 95
El equivalente en grados Celsius es: 0.0
```

Implementaciones Correctas

In [59]:

```
# Programa de Conversión de Temperatura INCORRECTO  
f = float(raw_input('Ingrese temperatura en grados Fahrenheit:'))  
c = (f - 32) * 5 / 9  
print 'El equivalente en grados Celsius es: ', c
```

```
Ingrese temperatura en grados Fahrenheit:95  
El equivalente en grados Celsius es: 35.0
```

In [60]:

```
# Programa de Conversión de Temperatura CORRECTO  
f = float(raw_input('Ingrese temperatura en grados Fahrenheit:'))  
c = (f - 32.0) * (5. / 9.)  
print 'El equivalente en grados Celsius es: ', c
```

```
Ingrese temperatura en grados Fahrenheit:95  
El equivalente en grados Celsius es: 35.0
```