

In [1]:

```
import os
```

In [1]:

```
#Entero - Integer - int (Número entero)
x = 2

#Cadena - String (Lista de caracteres) (el espacio es un caracter)
x = "lorena"

#Print muestra lo que contenga dentro del paréntesis
print (4) #Muestra 4
print (x) #Muestra lorena porque muestra la última variable de x
```

4  
lorena

In [2]:

```
#Entero - Integer - int (Número entero)
x = 7

#Cadena - String (Lista de caracteres)
x = "lorena"
print (x) #Muestra lorena

x = 7
print (x) #Muestra 7
```

lorena  
7

In [3]:

```
x = 5
y = 7

z= x + y
print (z) #Muestra 12 porque suma las variables
```

12

In [ ]:

```
h = 3
```

In [4]:

```
print (h) #Da error porque h no está ejecutado (botón play)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-4-176fc882a115> in <module>
----> 1 print (h) #Da error porque h no está ejecutado (botón play)
```

NameError: name 'h' is not defined

In [15]:

```
x = "lorena"
y = '"silvia"' #Tambien se puede poner con una sola comilla. Además, da la posibilidad de meter doble comilla y que salga en el print.

g = x + y
print (g) #Las cadenas se concatenan
```

lorena"silvia"

In [11]:

```
print (g + 2) #Da error porque no se pueden sumar string + enteros
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-11-dae7f1e83bd1> in <module>  
----> 1 print (g + 2) #Da error porque no se pueden sumar string + enteros  
  
TypeError: can only concatenate str (not "int") to str
```

In [13]:

```
type(g) #Te dice de que tipo es la variable que tiene en el interior
```

Out[13]:

str

In [18]:

```
print(g, z) #Muestra varios elementos mediante la coma y añade un espacio entre ellos.
```

lorena"silvia" 12

In [19]:

```
u = "g"  
silvia = "silvia tiene "  
anos = " años"  
  
suma = silvia + u + anos  
print(suma)
```

silvia tiene g años

In [20]:

```
n = 2  
m = "3"  
  
print(n+m) #Da error porque no se puede sumar int + str
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-20-1ef510070a8d> in <module>  
      2 m = "3"  
      3  
----> 4 print(n+m) #Da error porque no se puede sumar int + str  
  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [26]:

```
#Cambiar de int a str  
j = 2  
print(j)  
print(type(j))  
  
j = str(j)  
print(j)  
print(type(j))  
  
j = str(j) + " " + silvia #Hemos incluido un espacio entre j y la variable silvia(que es silvia tiene)  
print(j)  
print(type(j))
```

2  
<class 'int'>  
2

```
<class 'str'>
2 silvia tiene
<class 'str'>
```

In [ ]:

```
type
str
int
print
#Son palabras reservadas que no se debe poner igual, es decir, no se puede hacer type = 2
porque type sirve para saber el tipo de variable y por tanto no se le puede dar un valor
```

In [31]:

```
#Cambiar de str a int
lor = "lorena"
lor = int(lor) #En este caso da error porque no se puede pasar de una cadena escrita a en
tero
print(lor)
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-31-408092de65dd> in <module>
      1 #Cambiar de str a int
      2 lor = "lorena"
----> 3 lor = int(lor) #En este caso da error porque no se puede pasar de una cadena escr
ita a entero
      4 print(lor)
```

**ValueError:** invalid literal for int() with base 10: 'lorena'

In [32]:

```
#Cambiar de str a int
lor = "98"
lor = int(lor) #En este caso sí se puede ya que es un número.
print(lor)
```

98

In [33]:

```
#Para ver la longitud de una lista de caracteres (lista)
#Lenght

print(len(lor)) #Da error porque no puede traducir un entero (lo habíamos transformado de
str a int)
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-33-35ae1eac01db> in <module>
      2 #Lenght
      3
----> 4 print(len(lor))
```

**TypeError:** object of type 'int' has no len()

In [34]:

```
#Para ver la longitud de una lista de caracteres (lista)
#Lenght

mn = "lista de caracteres"
print(len(mn)) #Cuenta el espacio incluso
```

19

In [ ]:

```
x = 4 #int porque es entero
```

```
y = 4.2 #Float: con decimales
```

In [ ]:

```
num1 = 12
num2 = 3

suma = num1 + num2
resta = num1 - num2
multiplicacion = num1 * num2
division = num1 / num2 #Te hace la división en float
division_absoluta = num1 // num2 #No tiene en cuenta los decimales

print("suma:", suma)
print("resta", resta)
print("division", division)
print("multiplicacion", multiplicacion)
print("division absoluta", division_absoluta)
```

In [38]:

```
j = 12.5
k = 4
print(j / k)
print(j // k)
print(type(j / k))
print(type(j // k)) #Es float porque hay uno que es float (12.5) pero no tiene en cuenta los decimales

j = 13
k = 4
print(j / k)
print(j // k)
print(type(j / k))
print(type(j // k)) #No es float a pesar de que tenga decimales la división ya que los dos numeros son int

3.125
3.0
<class 'float'>
<class 'float'>
3.25
3
<class 'float'>
<class 'int'>
```

In [46]:

```
j = "2"
k = 3
y = 5

#No hace falta poner print en Jupyter Notebook pero solo imprime la ultima línea (y)
j #Si fuese la ultima, al ser cadena lo pone con comilla individual '2'
k
y
```

Out[46]:

5

In [47]:

```
#Para imprimir todas hay que poner print
print(j)
print(k)
y
```

2  
3

Out[47]:

5

In [48]:

```
#Otro ejemplo en
g = int(5.6/2)
g #Al ser int te quita los decimales
```

Out[48]:

2

In [51]:

```
#Declarar 3 variables:
    #Una con el nombre edad con el valor de nuestra edad
    #Otra "edad_companero_der" que contenga la edad de tipo entero de vuestro compañero d
e la derecha
    #Otra "suma anterior" que contenga la suma de las dos variables anteriormente declara
das

edad= 26
edad_companero_der= 22
suma_anterior = edad + edad_companero_der
suma_anterior
```

Out[51]:

48

In [52]:

```
i = "hola"
i.upper() #Para llamar a la función hay que poner los paréntesis
```

Out[52]:

'HOLA'

In [53]:

```
o = i.upper()
o.lower()
```

Out[53]:

'hola'

In [54]:

```
x = 2
x = x + 1
x #Actualiza el valor de x
```

Out[54]:

3

In [58]:

```
x = 2
x += 1 #Es lo mismo que (x + 1)
x
```

Out[58]:

3

In [ ]: