# Tipos Básicos

## Enteros

In [1]:
```
2 + 4 # suma
```

Out[1]: 6

In [2]:
```
3 * 5 # multiplicación
```

Out[2]: 15

In [3]:
```
4 - 7 # resta de números enteros
```

Out[3]: -3

In [4]:
```
8 // 3 # división entera, cociente
```

Out[4]: 2

In [5]:
```
8 % 3 # división entera, resto
```

Out[5]: 2

In [6]:
```
2 + 4 * 3 # la multiplicación tiene prioridad
```

Out[6]: 14

In [7]:
```
(2 + 4) * 3 # los paréntesis cambian esa prioridad
```

Out[7]: 18

In [8]:
```
2 ** 3 # potencia
```

Out[8]: 8

In [9]:
```
a = 5 # utilizamos el nombre a para acceder a un valor
b = a * 3 # podemos utilizar los nombres en las expresiones
a = 2 * a # se evalúa la expresión y se modifica el valor almacenado en
a
a, b
```

Out[9]: (10, 15)

In [10]:
```
n = 7
(n * (n + 1)) // 2 # los paréntesis no hacen falta en esta expresión
```

Out[10]: 28

In [11]:
```
def poli(x):
    c2 = 3 * x**2
    c1 = -4 * x
    c0 = -6
    return c2 + c1 + c0

poli(5)
```

Out[11]: 49

In [12]:
```
def poli(x):
    s = 0
    pot = 1
    s = s + (-6 * pot)
    pot = pot * x
    s = s + (-4 * pot)
    pot = pot *x
    s = s + (3 * pot)
    return s

poli(5)
```

Out[12]: 49

## Reales, aritmética de coma flotante

In [13]:
```
a = 2.0 # El punto indica la coma decimal
b = 3.0
a + b # suma
```

Out[13]: 5.0

In [14]:
```
a * 7.0 # multiplicación
```

Out[14]: 14.0

In [15]:
```
a / 6.0 # división entre reales
```

Out[15]: 0.3333333333333333

In [16]:
```
a * 7 # los enteros se convierten en reales automáticamente
```

Out[16]: 14.0

In [17]:
```
def poli(x):
    s = 0.0
    pot = 1
    s = s + (-6 * pot)
    pot = pot * x
    s = s + (-4 * pot)
    pot = pot *x
    s = s + (3 * pot)
    return s

poli(5)
```

Out[17]: 49.0

In [18]:
```
2.0 ** 3.1 # potencia
```

Out[18]: 8.574187700290345

In [19]:
```
import math # podemos usar muchas funciones matemáticas
```

In [20]:
```
radio = 3
2 * radio * math.pi # math.pi es un nombre que contiene el número pi
```

Out[20]: 18.84955592153876

In [21]:
```
math.sin(math.pi/3) # math.sin, el seno del ángulo en radianes
```

Out[21]: 0.8660254037844386

In [22]:
```
math.cos(math.pi/3) # math.cos, el coseno. Obsérverse el error obtenido
```

Out[22]: 0.5000000000000001

In [23]:
```
math.sqrt(5) # la raíz cuadrada
```

Out[23]: 2.23606797749979

In [24]:
```
int(3.9) # construye un entero truncando la parte decimal
```

Out[24]: 3

In [25]:
```
int(-3.9)
```

Out[25]: -3

In [26]:
```
round(3.2) # redondea al entero más próximo. El resultado sigue siendo r
eal
```

Out[26]: 3

In [27]:
```
round(3.5)
```

Out[27]: 4

In [28]:
```
round(-3.5)
```

Out[28]: -4

In [29]:
```
round(-3.2)
```

Out[29]: -3

In [30]:
```
math.floor(3.7) # redondea al entero inferior. El resultado sigue siendo
real
```

Out[30]: 3

In [31]:
```
math.floor(-3.2)
```

Out[31]: -4

In [32]:
```
math.ceil(3.2)
```

Out[32]: 4

In [33]:
```
math.ceil(-3.2)
```

Out[33]: -3

## Complejos

In [34]:
```
z = complex(2, 4) #números complejos
z**2
```

Out[34]: (-12+16j)

In [35]:
```
# la unidad imaginaria es j
(2+3j)**2 # otra notación
```

Out[35]: (-5+12j)

In [36]:
```
import cmath
cmath.sqrt(-1) # en el módulo cmath se encuentran las versiones compleja
s  de las funciones de math.
```

Out[36]: 1j

## Booleanos

In [37]:
```
a = True # valor lógico de cierto
b = False # valor lógico de falso.
```

In [38]:
```
a and b # conjunción lógica
```

Out[38]: False

In [39]:
```
a or b # disyunción lógica
```

Out[39]: True

In [40]:
```
a or True
```

Out[40]: True

In [41]:
```
b and False
```

Out[41]: False

In [42]:
```
a = 2
b = 4
b == a * 2 # la comparación devuelve un valor lógico
```

Out[42]: True

In [43]:
```
a * 3 >= 7
```

Out[43]: False

In [44]:
```
b * 2 < 8
```

Out[44]: False

In [45]:
```
b * 2 <= 8
```

Out[45]: True

In [46]:
```
a>=2 and b<7
```

Out[46]: True

In [47]:
```
a>=2 or b<7
```

Out[47]: True

In [48]:
```
def is_in_circle(x, y ,r):
    return x**2 + y**2 <= r*r

is_in_circle(0.5, 0.5, 1)
```

Out[48]: True

In [49]:
```
is_in_circle(0.5, 0.9, 1)
```

Out[49]: False

## Cadenas de caracteres

In [50]:
```
a = "Hola"
b = 'Hola' # las cadenas de caracteres se pueden poner con comillas dobl
es o simples.
a, b
```

Out[50]: ('Hola', 'Hola')

In [51]:
```
a == b
```

Out[51]: True

In [52]:
```
nombre = "Juan"
saludo = "Hola " + nombre # + es la concatenación de cadenas de caracter
es
saludo
```

Out[52]: 'Hola Juan'

In [53]:
```
"H" * 10 # replicación
```

Out[53]: 'HHHHHHHHHH'

In [54]:
```
"Juan" < "Jual" # las cadenas de caracteres se comparan según el orden l
exicográfico
```

Out[54]: False

In [55]:
```
"Juan" < "Juap"
```

Out[55]: True

In [56]:
```
"j" == "J" # las mayúsculas y las minúsculas son diferentes
```

Out[56]: False

In [57]:
```
"J" < "j"
```

Out[57]: True

In [58]:
```
len("Hola") # la longitud de una longitud de cadena de caraceres
```

Out[58]: 4

In [59]:
```
len("Adiós") # Las letras con tilde ocupan 2 bytes.
```

Out[59]: 5

In [60]:
```
a = "Adiós"
a
```

Out[60]: 'Adiós'

In [61]:
```
print(a)
```

```
Adiós
```

### Cadenas de caracteres largas

In [62]:
```
quijote = """En un lugar de la Mancha, de cuyo nombre no quiero acordarm
e,
no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero,
adarga antigua, rocín flaco y galgo corredor.
Una olla de algo más vaca que carnero, salpicón las más noches,
duelos y quebrantos los sábados, lantejas los viernes, algún palomino de
añadidura los domingos,
consumían las tres partes de su hacienda.
El resto della concluían sayo de velarte, calzas de velludo para las fie
stas, con sus pantuflos de lo mesmo,
y los días de entresemana se honraba con su vellorí de lo más fino.
Tenía en su casa una ama que pasaba de los cuarenta, y una sobrina que n
o llegaba a los veinte,
y un mozo de campo y plaza, que así ensillaba el rocín como tomaba la po
dadera.
Frisaba la edad de nuestro hidalgo con los cincuenta años; era de comple
xión recia, seco de carnes,
enjuto de rostro, gran madrugador y amigo de la caza.
Quieren decir que tenía el sobrenombre de Quijada, o Quesada, que en est
o hay alguna diferencia en los autores que deste caso escriben;
aunque, por conjeturas verosímiles, se deja entender que se llamaba Quej
ana.
Pero esto importa poco a nuestro cuento; basta que en la narración dél n
o se salga un punto de la verdad."""
```

In [63]:
```
print(quijote)
```

```
En un lugar de la Mancha, de cuyo nombre no quiero acordarme,
no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero,
adarga antigua, rocín flaco y galgo corredor.
Una olla de algo más vaca que carnero, salpicón las más noches,
duelos y quebrantos los sábados, lantejas los viernes, algún palomino de
añadidura los domingos,
consumían las tres partes de su hacienda.
El resto della concluían sayo de velarte, calzas de velludo para las fies
tas, con sus pantuflos de lo mesmo,
y los días de entresemana se honraba con su vellorí de lo más fino.
Tenía en su casa una ama que pasaba de los cuarenta, y una sobrina que no
llegaba a los veinte,
y un mozo de campo y plaza, que así ensillaba el rocín como tomaba la pod
adera.
Frisaba la edad de nuestro hidalgo con los cincuenta años; era de complex
ión recia, seco de carnes,
enjuto de rostro, gran madrugador y amigo de la caza.
Quieren decir que tenía el sobrenombre de Quijada, o Quesada, que en esto
hay alguna diferencia en los autores que deste caso escriben;
aunque, por conjeturas verosímiles, se deja entender que se llamaba Queja
na.
Pero esto importa poco a nuestro cuento; basta que en la narración dél no
se salga un punto de la verdad.
```

### Caracteres especiales

In [64]:
```
especial1 = "Las \"comillas\" son caracteres especiales"
especial1
```

Out[64]:
```
'Las "comillas" son caracteres especiales'
```

In [65]:
```
especial2 = "Las \'comillas\' son caracteres especiales"
especial2
```

Out[65]:
```
"Las 'comillas' son caracteres especiales"
```

In [66]:
```
especial3 = "Por tanto la barra invertida también lo es \\"
especial3
```

Out[66]:
```
'Por tanto la barra invertida también lo es \\'
```

In [67]:
```
especial4 = "Hay otros caracteres especiales como el salto de línea.\nQu
e sirve para representar el fín de línea en ficheros.\nLo veremos más ad
elante"
print(especial4)
```

```
Hay otros caracteres especiales como el salto de línea.
Que sirve para representar el fín de línea en ficheros.
Lo veremos más adelante
```

## Errores comunes

In [68]:
```
def media (a, b):
    return (a + b) / 2 # es la división real aunque los argumentos sean
enteros.
media(1, 10)
```

Out[68]:
```
5.5
```

In [69]:
```
def media (a, b):
    return (a + b) // 2 # es la división entera.
media(1, 10)
```

Out[69]:
```
5
```

In [70]:
```
def media (a, b):
    return float(a + b) // 2 # la función float construye un real a part
ir de un entero. La división es entera pero como
media(1,10)                 # un argumento es real el resultado es un re
al.
```

Out[70]:
```
5.0
```

In [71]:
```
def paradoja(a):
    b = math.sqrt(a) # la raíz cuadrada de 2
    return a == b * b # esto debería ser cierto siempre, no lo es por el
error de la representación de los reales
paradoja(2)
```

Out[71]:
```
False
```

In [72]:
```
0.5 == math.cos(math.pi/3) # teóricamente esto también debería ser ciert
o.
```

Out[72]:
```
False
```

In [73]:
```
def fibonacci(n):
    phi = (1 + math.sqrt(5)) / 2
    return (phi ** n - (1 - phi) ** n)/math.sqrt(5)
fibonacci(4)
```

Out[73]:
```
3.0000000000000004
```

In [74]:
```
def fibonacci(n):
    phi = (1 + math.sqrt(5)) / 2
    return int(round((phi ** n - (1 - phi) ** n)/math.sqrt(5)))
fibonacci(4)
```

Out[74]: 3