

```
In [1]: import cmath, math, random, statistics
        from fractions import Fraction
```

Números:

Enteros en distintas bases

```
In [2]: entero = 100
        hexa = 0xFF
        entero, hexa
```

```
Out[2]: (100, 255)
```

```
In [3]: hex(entero), bin(hexa), oct(entero)
```

```
Out[3]: ('0x64', '0b11111111', '0o144')
```

Flotantes y decimales de precisión fija

```
In [4]: flotante = 10.45
        mantisas = float('1e-003')
        infinitos = float('-Infinity')
        no_definido = float("nan")

        flotante, mantisas, infinitos, no_definido
```

```
Out[4]: (10.45, 0.001, -inf, nan)
```

Fracciones

```
In [5]: frac = Fraction("4/5")
        frac, frac * Fraction("2/3"), frac * mantisas, frac * no_definido
```

```
Out[5]: (Fraction(4, 5), Fraction(8, 15), 0.0008, nan)
```

Imaginario

```
In [6]: imag = 4 + 5j
        imag * 2, imag, imag + 5 + 9j
```

```
Out[6]: ((8+10j), (4+5j), (9+14j))
```

```
In [7]: imag + no_definido
```

```
Out[7]: (nan+5j)
```

Operaciones

Sumar, restar, potencias, dividir, resto, división entera

```
In [8]: hexa + entero, hexa * entero, hexa ** 3
```

```
Out[8]: (355, 25500, 16581375)
```

```
In [9]: hexa / entero, hexa % entero, hexa // entero
```

```
Out[9]: (2.55, 55, 2)
```

Funciones matemáticas

```
In [10]: math.sqrt(hexa), math.log(hexa)
```

```
Out[10]: (15.968719422671311, 5.541263545158426)
```

```
In [11]: math.sin(60), math.cos(60), math.tan(60)
```

```
Out[11]: (-0.3048106211022167, -0.9524129804151563, 0.320040389379563)
```

```
In [12]: cmath.sin(imag), cmath.cos(imag)
```

```
Out[12]: ((-56.16227422023235-48.50245524177091j),  
          (-48.506859457844584+56.15717492513018j))
```

```
In [13]: cmath.polar(imag)
```

```
Out[13]: (6.4031242374328485, 0.8960553845713439)
```

Números aleatorios

```
In [14]: random.randint(1, 100)
```

```
Out[14]: 99
```

```
In [15]: random.gauss(mu=0, sigma=10.5)
```

```
Out[15]: -16.346364909850635
```

Listas

```
In [16]: mi_lista = [4, 13, 0, -2, 100]  
mi_lista
```

```
Out[16]: [4, 13, 0, -2, 100]
```

```
In [17]: mi_lista[0]
```

```
Out[17]: 4
```

```
In [18]: mi_lista[-1]
```

```
Out[18]: 100
```

```
In [19]: mi_lista[1:3]
```

```
Out[19]: [13, 0]
```

Procesado de listas por comprensión

```
In [20]: lista = [random.gauss(0, 0.6) for x in range(12)]  
lista
```

```
Out[20]: [0.013098860821720073,
          -0.41711290195712114,
          1.4030250544750604,
          -1.2315113097308397,
          0.4038148211318986,
          -0.6465557302067724,
          1.1809481370294428,
          -0.5030861212443873,
          -0.10340319416242585,
          0.6538796995454971,
          -0.03411292075196376,
          0.0025312568699263414]
```

```
In [21]: lista = [int(x * 1000) for x in lista]
         lista
```

```
Out[21]: [13, -417, 1403, -1231, 403, -646, 1180, -503, -103, 653, -34, 2]
```

```
In [22]: lista.sort()
         lista
```

```
Out[22]: [-1231, -646, -503, -417, -103, -34, 2, 13, 403, 653, 1180, 1403]
```

```
In [23]: random.shuffle(lista)
         lista
```

```
Out[23]: [1180, 13, 653, 2, -646, 403, -417, 1403, -34, -503, -103, -1231]
```

Estadísticas

```
In [24]: # help(statistics)
```

```
In [25]: statistics.stdev(lista)
```

```
Out[25]: 755.4613280518191
```

```
In [26]: statistics.mean(lista), statistics.median(lista)
```

```
Out[26]: (60, -16.0)
```

```
In [27]: lista.sort()
         statistics.quantiles(lista, n=5)
```

```
Out[27]: [-560.2, -89.2, 10.8, 863.8]
```

```
In [28]: random.shuffle(lista)
         statistics.quantiles(lista, n=5)
```

```
Out[28]: [-560.2, -89.2, 10.8, 863.8]
```

Caracteres

```
In [29]: nombre = "Ana"
         apellido = "Perez"
         apellido + ", " + nombre + ": alumna de la UNNE"
```

```
Out[29]: 'Perez, Ana: alumna de la UNNE'
```

```
In [30]: lista_año_22 = ["María", "Pedro", "Marta", "Alfonso"]
```

```
sorted(lista_año_22)
```

```
Out[30]: ['Alfonso', 'Marta', 'María', 'Pedro']
```

```
In [31]: todos = lista_año_22 + ["Inés", "Alfredo", "Martín"]
todos
```

```
Out[31]: ['María', 'Pedro', 'Marta', 'Alfonso', 'Inés', 'Alfredo', 'Martín']
```

```
In [32]: todos += ["Zulma", "Ulises"]
todos
```

```
Out[32]: ['María',
          'Pedro',
          'Marta',
          'Alfonso',
          'Inés',
          'Alfredo',
          'Martín',
          'Zulma',
          'Ulises']
```

```
In [33]: afuera = [todos.pop(), todos.pop()]
todos, afuera
```

```
Out[33]: (['María', 'Pedro', 'Marta', 'Alfonso', 'Inés', 'Alfredo', 'Martín'],
          ['Ulises', 'Zulma'])
```

Diccionarios

```
In [34]: a = {"Ana": 10, "Oscar": 50}
a["Ana"]
```

```
Out[34]: 10
```

```
In [ ]:
```