



Python Basics

Funciones internas

No modifican los objetos

```
lista = [1,2,3]

len(lista)
max(lista)
type(lista)
print(lista)
str(lista)
```

Funciones externas

Importando otros módulos

```
import math
math.sqrt(25)
math.cos(0)
```

Métodos

Pueden modificar los objetos

```
texto = "Hola a todos"

texto.lower()
texto.upper()
texto.replace('todos', 'todas')
texto.split(" ")
```

Listas

```
notas = ['A', 'B', 'D']

notas[0] # 'A'
notas[-1] # 'D'
notas.append('Z') # Método
print(notas) # Función
```

A	B	A and B	A or B	not A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

U Flujos de control

if elif else

```
if nota > 0 and nota < 5:
    print('Suspenso')
elif nota > 5 and nota < 10:
    print('Aprobado')
elif nota == 10:
    print('Matrícula')
    print('Enhorabuena!')
else:
    print('Nota no válida')
```

elif y else son opcionales

for

```
dias = [1,5,10,15,30]

for dia in dias:
    print(dia)

for k in range(len(dias)):
    print(k, dias[k])

for ind, val in enumerate(dias):
    print(ind, val)
```

break continue

```
for el in [1,2,0,-2,3]:
    if el < 0:
        break
    elif el == 0:
        continue
    print(10/el)
```

try except

```
for el in [1,2,3,"cuatro"]:
    try:
        print(el+2)
    except:
        print(el, 'no es un número')
```

while

```
i = 0

while(i < 5):
    print(i)
    i = i + 1
```

U Colecciones

Comprensión de listas: List = [**expression** for **item** in **iterable** (if conditional)]

Colección	Sintaxis	Acceso	Mutable	Ordenada
LISTA	L = [1, 2, 'Z']	L[0]	SÍ	SÍ
TUPLA	T = (1, 2, 'Z')	T[1]	NO	SÍ
DICCIONARIO	D = {'A':1, 'B':2, 'C':'Z'}	D['A']	SÍ	NO
SET	S = {1, 2, 'Z'}	NO	SÍ	NO (elementos únicos)

```
pelis = {'Toy Story':['Aventura','Comedia','Fantasia'],  
        'Jumanji':['Aventura','Fantasia'],  
        'Sabrina':['Comedia','Drama','Romance'],  
        'Casino':['Crimen','Drama']  
}
```

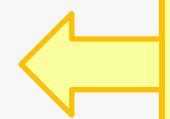


Diccionario de listas

```
pelis['Casino'] # ['Crimen', 'Drama']
```

```
generos = [] # lista de géneros (inicialmente vacía)
```

```
for clave, valor in pelis.items():  
    for gen in valor:  
        generos.append(gen)
```



Bucle en diccionarios

```
generos # ['Aventura','Comedia','Fantasia','Aventura','Fantasia','Comedia','Drama','Romance','Crimen','Drama']
```

```
# Valores únicos
```

```
set(generos) # {'Aventura', 'Comedia', 'Crimen', 'Drama', 'Fantasia', 'Romance'}
```

U Colecciones

Ejemplos de colecciones reales

```
cutoas = {
  "group_name_2": "FBSPA Segunda División > Regular Season-19",
  "offer_lastupdate": 1567116039,
  "ateam_id": 8694,
  "hteam_id": 1010,
  "source": "ahc",
  "match": "Almeria v Huesca",
  "match_lastupdate": 1567084019,
  "group_name": "FBSPA",
  "offer_inrunning": 0,
  "ts": 1567116037,
  "bookmaker_name": "victor",
  "hteam_name_2": "Almeria",
  "sport_id": 1,
  "match_live": "true",
  "price_oh": 2.1100000000000003,
  "competition_id": 12,
  "match_id": 1676695,
  "hteam_name": "Almeria",
  "competition_name": "Segunda División",
  "bookmaker_id": 22,
  "offer_isturn": 0,
  "ateam_name": "Huesca",
  "match_txid": 5661535,
  "offer_ot": "FT - Asian Handicap",
  "offer_lineid": 1,
  "offer_bmoid": 0,
  "offer_ts": 1567116,
  "offer_id": 16766950220331,
  "score_count": 0,
  "league_id": 85189,
  "event_type": "compact_price_update",
  "price_oa": 1.8,
  "ateam_name_2": "Huesca",
  "offer_otid": 33,
  "_query": "simple",
  "match_xsid": 1167403,
  "price_type": "normal",
  "seq": -1154899510,
  "group_name_ch": "\u005cu897f\u005cu73ed\u005cu7259\u005cu4e59\u005cu7d44\u005cu806f\u005cu8cfd",
  "status": "real-time",
  "match_time": 1567274400,
  "league_name": "SPA Segunda Division",
  "offer_flags": 1
}
```

```
airbnb =
{ '_id': '10009999',
  'listing_url': 'https://www.airbnb.com/rooms/10009999',
  'name': 'Horto flat with small garden',
  'summary': 'One bedroom + sofa-bed in quiet and bucolic neighbourhood ri',
  'space': 'Lovely one bedroom + sofa-bed in the living room, perfect for',
  'description': 'One bedroom + sofa-bed in quiet and bucolic neighbourhood',
  'neighborhood_overview': 'This charming ground floor flat is located in',
  'notes': 'There´s a table in the living room now, that does not show in',
  'transit': 'Easy access to transport (bus, taxi, car) and easy free park',
  'property_type': 'Apartment',
  'room_type': 'Entire home/apt',
  'bed_type': 'Real Bed',
  'minimum_nights': '2',
  'maximum_nights': '1125',
  'cancellation_policy': 'flexible',
  'accommodates': 4,
  'bedrooms': 1,
  'beds': 2,
  'number_of_reviews': 0,
  'bathrooms': 1,
  'amenities': ['Wifi',
    'Wheelchair accessible',
    'Kitchen',
    'Free parking on premises',
    'Smoking allowed',
    'Hot tub',
    'Buzzer/wireless intercom',
    'Family/kid friendly',
    'Washer',
    'First aid kit',
    'Essentials',
    'Hangers',
    'Hair dryer',
    'Iron',
    'Laptop friendly workspace'],
  'price': '317.00',
  'host_verifications': ['email', 'phone', 'facebook'],
  'address': {'street': 'Rio de Janeiro, Rio de Janeiro, Brazil',
    'suburb': 'Jardim Botânico',
    'government_area': 'Jardim Botânico',
    'market': 'Rio De Janeiro',
    'country': 'Brazil',
    'country_code': 'BR'}
}
```

```
crypto = [
  {
    "price_close": 131.62,
    "price_high": 131.91,
    "price_low": 130.48,
    "price_open": 130.66,
    "time_close": "2019-01-01 00:59:57.900",
    "time_open": "2019-01-01 00:00:00.375",
    "time_period_end": "2019-01-01 01:00:00.000",
    "time_period_start": "2019-01-01 00:00:00.000",
    "trades_count": 1501,
    "volume_traded": 8127.92426351
  },
  {
    "price_close": 130.79,
    "price_high": 131.62,
    "price_low": 130.77,
    "price_open": 131.58,
    "time_close": "2019-01-01 01:59:37.975",
    "time_open": "2019-01-01 01:00:02.315",
    "time_period_end": "2019-01-01 02:00:00.000",
    "time_period_start": "2019-01-01 01:00:00.000",
    "trades_count": 1323,
    "volume_traded": 3346.56927114
  },
  {
    "price_close": 130.06,
    "price_high": 130.88,
    "price_low": 129.55,
    "price_open": 130.8,
    "time_close": "2019-01-01 02:59:46.621",
    "time_open": "2019-01-01 02:00:06.532",
    "time_period_end": "2019-01-01 03:00:00.000",
    "time_period_start": "2019-01-01 02:00:00.000",
    "trades_count": 1150,
    "volume_traded": 6482.35342142
  }
]
```


U Funciones

Definición

```
def f1(a, b, c):  
    return (a + b) / c  
  
def f2():  
    return 3.1415926  
  
def f3():  
    print(3.1415926)  
  
def f4(a, b = 10):  
    return (a * b - a)  
  
def f5(*args):  
    return max(args)/min(args)
```

Uso

```
s1 = f1(1, 2, 3) # s1 vale 1 (hay que usar 3 argumentos sí o sí)  
s1_2 = f1(c = 3, b = 2, a = 1) # s1_2 vale 1  
s2 = f2() # s2 vale 3.1415926 (puedo tener funciones sin argumentos)  
s3 = f3() # s3 vale None. La función imprime 3.1415926 (no hay return)  
s4 = f4(5) # s4 vale 45 (el argumento b es igual a 10 por defecto)  
s4_2 = f4(5, 4) # s4_2 vale 15 (he usado los dos argumentos)  
s5 = f5(2, 4, 6) # s5 vale 3 (función con argumentos variables)
```