HOJA DE REFERENCIA DE PYTHON - NIVEL FUNDAMENTAL

EL GIMNASIO DE PYTHON

CÓDIGO PITÓN

Asignación de variables

```
x = 10      # enteros
y = 0.5     # decimales
z = 'hola'  # texto

tupla = (1, 2, 3)
lista = [4, 5.5, 'a6']
conjunto = {'a', 'b', 'c'}

diccionario = {10: 1.1, 20: 1.0, 30: 1.3}
diccionario[40] = 1.5

x, y, z = 1, 2, 3  # asignación múltiple

x, y = y, z  # intercambio de valores

x, y, z = (1, 2, 3)  # desempaquetado
```

Acceso a valor de variables

```
valor = lista[3]  # acceso a posición 3
valor = tupla[4]  # acceso a posición 4
valor = texto[5]  # acceso a carácter 5

valor = diccionario['clave']
# valor 'def' si 'clave' no existe
valor = diccionario.get('clave', 'def')

valor = Clase.atrib  # variable de clase
valor = instancia.atrib
```

Aritmética básica

```
3 + 5   # sumas
5 - 3   # restas
5 * 3   # multiplicaciones
3 ** 2  # potencias 3 ** 2 == 3² == 9
5 / 3   # división 5 / 3 == 1.6666667
5 // 3  # división entera 5 // 3 == 1
5 % 3   # módulo (resto o residuo) de
        # la división entera 5 % 3 == 2
```

Importación de módulos

```
import módulo
import módulo as m

from módulo import elemento
from módulo import elemento as e
```

Gestión de excepciones

```
try:          # para capturar excepciones
    código 1 # código a ejecutar
except:
    código 2 # se ejecuta si se produce
             # error en código 1
finally:      # opcional
    código 3 # se ejecuta siempre aunque
             # haya error en código 1


raise ValueError()  # generar un error

# errores comunes
SyntaxError  # sintaxis incorrecta
ValueError   # valor argumento no válido
ZeroDivisionError   # división por cero
NameError    # variable no definida
ModuleNotFoundError # modulo no existe
FileNotFoundError   # fichero no existe
IndexError   # índice no existe
KeyError     # clave no existe
```

Salida y formato de texto

```
>>> a = 10
>>> b = 20
>>> print(10)
10
>>> print('a: ' + str(a))
a: 10

>>> print('a: %i' % a)
a: 10

>>> print('a: %i, b: %i' % (a, b))
a: 10, b: 20

>>> print('a: {}, b: {}'.format(a, b))
a: 10, b: 10

>>> print(f'a: {a}, b: {b}')
a: 10, b: 10
```

Listas

```
lista = [5, 6, 7]

lista.append(8)      # [5, 6, 7, 8]
lista.insert(1, 8)   # [5, 8, 6, 7]
lista.remove(6)      # [5, 7]
del lista[0]         # [6, 7]
lista.clear()        # []
lista.extend([8, 9]) # [5, 6, 7, 8, 9]
lista.reverse()      # [7, 6, 5]
len(lista)           # 3
lista.index(7)       # 2
```

Condicionales

```
if condición:     # condicional simple
    instrucción

if condición:     # condicional if else
    instrucción 1
else:
    instrucción 2

if condición 1:   # ifs excluyentes
    instrucción 1
elif condición 2:
    instrucción 2
else:
    instrucción 3
```

No existe la sentencia switch

Más información de condicionales

Booleanos y comparaciones

```
True        # valor cierto
False       # valor false

a > b       # 3 > 2 es True
a >= b      # 3 >= 3 es True
a < b       # 3 < 2 es False
a <= b      # 3 <= 3 es True
a == b      # 3 == 3 es True
a != b      # diferente 3 != 3 es False
a in lista  # 3 in [2, 3, 4] es True
```

Más información

Comprensión de listas

```
>>> [ v * 2 for v in [1, 2, 3] ]
[2, 4, 6]
```

Comentarios

```
# comentario de una línea al comienzo

a = 5  # comentario de una línea al final

'''
Comentario
de varias líneas
con comillas simples
'''

"""
Otro comentario
de varias líneas
con comillas dobles
"""
```

Funciones

```
def suma(a, b):
    return a + b

r = suma(10, 15)   # r == 25

def opera(a, b, c):
    return suma(a, b), b - c

r, s = opera (10, 15, 3)  # r == 25
                          # s == 12
```

Bucles

```
while condición:
    código  # se ejecuta mientras
            # la condición sea cierta

for valor en secuencia: # para cada valor
    código con valor    # en la secuencia

continue  # sigue con la siguiente vuelta
break     # detiene el bucle
```

Notación de porciones

Para listas, tuplas, cadenas de texto y otras secuencias

```
# porción desde ini hasta fin
# tomando elementos de paso en paso
sec[ini:fin:paso]

sec = [10, 20, 30, 40, 50, 60]

sec[2:5] == [30, 40, 50]
sec[4:] == [50, 60]
sec[:2] == [10, 20]
sec[1:6:2] == [20, 40, 60]
sec[4:1:-1] == [50, 40, 30]
sec[:] == [10, 20, 30, 40, 50, 60]
sec[::-1] == [60, 50, 40, 30, 20, 10]
```

Entrada por teclado

```
>>> cadena = input('Dame un valor: ')
Dame un valor: hola
>>> print(cadena)
hola
```

ACADEMIA CÓDIGO PITÓN