

# CHEATSEET PYTHON3 hoja 1

## VARIABLES

Crear variable entera: edad = 25

Crear variable coma flotante: coeficiente = 2.28

Crear variable de texto: objetivo = "Aprender a programar"

## COMENTARIOS

El símbolo # indica que lo que sigue es un comentario. Este comentario no se tomará en cuenta al ejecutar el código.

## OPERADORES ARITMÉTICOS

Básicos: 5 + 2 - (4 \* 3) # el resultado es -5

División normal: 5 / 2 # el resultado es 2.5

División entera: 5 // 2 # el resultado es 2

Módulo: 5 % 2 # el resultado es 1 (resto de la división entera)

Potencia: 5 \*\* 2 # el resultado es 25

## OPERADORES DE COMPARACION

== # igual que

!= # distinto que

> # mayor que

< # menor que

>= # mayor o igual que

<= # menor o igual que

## OPERADORES BOOLEANOS

True # es verdadero

False # es falso

and # y

or # o

not # no

## MANEJO DE CONSOLA

Imprimir texto a consola: print("Adimra")

Imprimir variable a consola:

variable = "Adimra"

print(variable) o print("Curso " + variable)

Leer variable de texto desde consola:

pais = input("Indicar país de origen: ")

Leer variable entera desde consola:

edad = int(input("Indicar edad: "))

Leer variable flotante desde consola:

coeficiente = float(input("Indicar coeficiente: "))

## CICLO FOR

suma = 0

for ciclo in range(5):

    suma = suma + ciclo

print(suma)

# el resultado final de suma será 10, pues la variable de ciclo

# siempre inicia en 0, => 0 + 1 + 2 + 3 + 4

# CHEATSEET PYTHON3 hoja 2

## CONDICIONALES

Condicional si:

```
if (x > 5):  
    print("x es mayor a 5")
```

Condicional si / sino:

```
if (edad < 18):  
    print("Es menor de edad")  
else:  
    print("Es mayor de edad")
```

Condicional si / sino si / sino:

```
if (nota >= 8):  
    print("Excelente")  
elif (nota >= 6):  
    print("Está bien")  
elif (nota >= 4):  
    print("Falta poco")  
else:  
    print("Falta bastante")
```

## MANEJO DE CADENAS

Crear cadena con comillas simples o dobles:

```
cadena1 = "Curso"  
cadena2 = ' Adminra'  
print(cadena1 + cadena2) # imprime Curso Adimra
```

Concatenación:

```
tipo = "Introdutorio"  
mensaje = "Curso " + tipo + " Adminra"  
print(mensaje)  
# Imprime Curso Introdutorio Adminra
```

Convertir tipo para concatenación:

```
edad = 23 # es un entero  
nombre = "Juan Perez"  
mensaje = nombre + " tiene " + str(edad)
```

Formateo de cadena (para evitar conversiones):

```
nombre = "Alicia"  
edad = 20  
mensaje = f"{nombre} tiene {edad} años"  
# solo funciona desde Python 3.6 en adelante
```

Detectar si una cadena contiene otra cadena:

```
"man" in "manzana" # True, es verdadero  
"lim" in "manzana" # False, es falso
```

Variable cadena a minúsculas o mayúsculas:

```
cadena.lower() # minúsculas  
cadena.upper() # mayúsculas
```

# CHEATSEET PYTHON3 hoja 3

## CICLO WHILE

Infinito:

```
while(True):  
    print("Este mensaje se imprime constantemente")
```

Con condición:

```
cantidad = 0  
while(cantidad < 5):  
    print(cantidad)  
    cantidad = cantidad + 1  
# Este while se ejecutará 5 veces  
# Para estos casos se puede utilizar directamente un for()
```

Indefinido:

```
ingreso = -1  
while(True):  
    ingreso = int(input("Ingresar número: "))  
    if (ingreso == 0):  
        break  
    print(ingreso)  
# Se ejecutará de forma indefinida hasta que se ingrese 0  
# El mismo comando break se puede utilizar en el for()
```

## LISTAS

Crear:

```
lista1 = [22, 15, 0, -3, 10]  
lista2 = [22, 15, "Carlos", -3, True]  
# En Python las listas pueden contener distintos tipos de datos  
# aunque mayormente utilizaremos el primer caso (lista1)
```

Iterar:

```
for valor in lista1: o bien for indice, valor in enumerate(lista1)
```

Agregar:

```
lista1.append(16) # se agrega el item 16 al final de la lista
```

Quitar: `lista1.remove(-3)` # se quita el item -3 de la lista

Ordenar:

```
lista1.sort() # ascendente: -3, 0, 10, 15, 22  
lista1.sort(reverse=True) # descendente: 22, 15, 10, 0, -3  
lista1.reverse() # lista1 queda ordenada como: 10, -3, 0, 15, 22
```

Verificar si contiene un item (devuelve True o False):

```
if (-3 in lista1):  
    print("Lista1 contiene un -3")
```

Obtener índice en la lista: `indice = lista1.index(15)` # indice es 1

# CHEATSEET PYTHON3 hoja 4

## LISTAS vs TUPLAS vs DICCIONARIOS

Listas:

```
listaVariables = [23, 15, 1000, 0, -2]
```

- Se definen con corchetes [ ]
- Son mutables (podemos agregar, quitar o modificar items), utilizando `append()`, `remove()` o `lista[indice] = nuevoValor`.
- Pueden contener distintos tipos de datos, aunque usaremos habitualmente un solo tipo.
- Son objetos, que disponen toda una serie de propiedades.

Tuplas:

```
tuplaDias = ("Lun", "Mar", "Mie", "Jue", "Vie", "Sab", "Dom")
```

- Se definen con paréntesis ( )
- Son inmutables (no podemos agregar, quitar o modificar)

Diccionarios:

```
datosPersonales = { "codigo": 1344, "apellido": "Perez",  
"nombre": "Juan", "saldo": 34620 }
```

- Se definen con llaves { }
- Cada item está formado por un par clave:valor.
- Son parcialmente mutables (podremos agregar, quitar

o modificar valores de claves, pero no nombres de claves.

Agregar item:

```
datosPersonales["tipoCuenta"] = 3
```

# agrega la clave tipoCuenta con el valor 3, al final del diccionario.

Modificar valor clave:

```
datosPersonales["codigo"] = 1345
```

# cambia el valor de la clave codigo a 1345

Quitar item:

```
datosPersonales.pop("codigo", None)
```

# borrar la clave codigo. El parámetro None evita que el intérprete genere un error si la clave no existe en el diccionario.

## DATOS DESDE ARCHIVO

Abrir archivo para lectura:

```
archivoDatos = open("datos.txt", "r") # r = read, lectura
```

```
for linea in archivoDatos:
```

```
    print(linea)
```

# CHEATSEET PYTHON3 hoja 5

## FUNCIONES

Definición:

```
def nombreFuncion():
```

Llamado:

```
nombreFuncion()
```

Con argumentos:

```
def calculo(ancho, alto):  
    print("Superficie:", ancho * alto)
```

```
calculo(30, 5.5)
```

Con retorno de valor:

```
def calculo(ancho, alto):  
    return ancho * alto
```

```
superficie = calculo(30, 5.5)
```

## FUNCIONES

Python puede retornar más de un valor desde una función:

```
def calculos(ancho, alto):  
    superficie = ancho * alto  
    perimetro = ancho * 2 + alto * 2  
    return superficie, perimetro
```

```
valorSuperficie, valorPerimetro = calculos(30, 5.5)
```