Python 3 Español Cheat Sheet

by Ismael Mercado (torerohk) via cheatography.com/23626/cs/5397/

print()

```
print()
```

print al ser funcion siempre se utiliza con parentesis

Variable String

```
y = "a"
z = 'Hola'
multilinea = " " " cadena de texto
con mas de una linea"""
```

variables de tipo string van entre comillas dobles o sencillas
para cadenas de texto de multiples lineas se utiliza """ texto """

Operadores Matematicos

suma	a+b
resta	a-b
multiplicacion	a*b
division_real	a/b
division_entera	a//b
resto	a%b
potencia	a**b

los operadores matemáticos principales pueden utilizarse combinados respetando la jerarquía al resolverlas
1. Resolver ( ) [ ] { }
2. Resolver exponentes.
3. Resolver * y / de izquierda a derecha
4. Resolver + y - de izquierda a derecha

Operadores Logicos

Igual a	==
Diferente a	!=
Menor que	<
Menor o igual que	<=
Mayor que	>
Mayor o igual que	>=

Devolverán un valor boleano

Metodos para Strings

len() retorna longitud de caracteres en string:

```
len(st ring)
```

lower() retorna string en minúsculas:

```
string.lo wer()
```

upper() retorna string en mayúsculas:

```
string.up per()
```

capitalize() retorna primer carácter de string en mayúsculas

```
string.ca pit alize()
```

str() retorna conversión explícita de strings:

```
str(st ring)
```

Literales

```
variab le.l ower()
variab le.u pper()
```

No Literales

```
len(va riable)
str(va riable)
```

String Inmutable a Flexible

```
print("%s" % (variable)) o print("%s" %
("string"))
nom= " Ism ael "
ape = " Mer cad o"
# variables
print ("mi nombre %s. mi apellido %s ." % (nom,
ape))
# strings
print ("mi nombre %s. mi apellido %s ." % ("Is -
mae l", " Mer cad o"))
```

Comparadores guia

	AND	
True	True	True
True	False	False
False	True	False
False	False	False
	OR	
True	True	True
True	False	True
False	True	True
False	False	False

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Comparadores guia (cont)

NOT	
True	False
False	True

Diccionarios

Estructura de datos que almacena valores utilizando otros como referencia para su acceso y almacenamiento, es iterable, mutable y puede contener elementos de diferente tipo; se declara entre llaves *{clave:valor}*

```
diccio nar io= {'a':1, 'b':2, 'c':3}
```

Podemos utilizar la funcion **dict()**

```
diccio nar io= dic t(a=1, b=2, c=3)
```

Acceder a un elemento utilizamos el indice

```
diccio nar io['c']
```

Modificar un valor

```
diccio nar io[ 'b']=28
```

Nuevos elementos añadimos una clave no existente

```
diccio nar io[ 'd']=4
```

Iterar con un diccionario

items() *Acceso a claves y valores*

```
diccio nar io.i tems()
```

values() *Acceso a valores*

```
diccio nar io.v al ues()
```

keys() *Acceso a claves*

```
diccio nar io.k eys()
```

Ordenar un diccionario

```
sorted (di cci onario)
```

Ordenar un diccionario en inverso

```
sorted (di cci onario, revers e=True)
```

Matrices

Anidando listas construimos matrices de elementos

```
matriz =[[ 1,2 ,3] ,[4 ,5,6]]
```

para acceder a los elementos utilizamos

```
matriz [0][1]
```

sustituir un elemento

```
matriz [1] [0]=33
```

crear, modificar y leer archivos en disco

Función para crear un archivo

```
def crearA rch ivo():
   arc hiv o=o pen ('d ato s.txt', 'w')
    ar chi vo.c lose()
```

Función para escribir en un archivo

```
def escrib irA rch ivo():
    ar chi vo= ope n(' dat os.t xt', 'a')
    ar chi vo.w ri te( 'prueba de texto\n')
    ar chi vo.c lose
```

Función para leer un archivo

```
def leerAr chi vo():
    ar chi vo= ope n(' dat os.t xt', 'r')
    linea = archiv o.r ead line()
    while linea! ="":
         p rin t(l inea)
         l ine a=a rch ivo.re adl ine()
    ar chi vo.c lose()
```

Modos de apertura de archivos

Indicador	Modo de apertura	Ubicación del puntero
`r	Solo lectura	Al inicio del archivo
rb	Solo lectura en modo binario	Al inicio del archivo
r+	Lectura y escritura	Al inicio del archivo

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Modos de apertura de archivos (cont)

rb+	Lectura y escritura en modo binario	Al inicio del archivo
w	Solo escritura. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
wb	Solo escritura en modo binario. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
w+	Escritura y lectura. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
wb+	Escritura y lectura en modo binario. Sobreescribe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
a	Añadido (agregar contenido). Crea el archivo si éste no existe	Si archivo existe, al final. Si no, al comienzo
ab	Añadido en modo binario (agregar contenido). Crea el archivo si éste no existe	Si archivo existe, al final. Si no, al comienzo
a+	Añadido (agregar contenido) y lectura. Crea el archivo si éste no existe.	Si archivo existe, al final. Si no, al comienzo

Modos de apertura de archivos (cont)

ab+	Añadido (agregar contenido) y lectura en modo binario. Crea el archivo si éste no existe	Si archivo existe, al final. Si no, al comienzo

indicado a la función open() como una string en su segundo parámetro.

Funciones integradas

__impo rt__()	abs()	all()
any()	ascii()	bin()
bool()	bytear ray()	bytes()
callable()	chr()	classm ethod()
compile()	complex()	delattr()
dict()	dir()	divmod()
enumer ate()	eval()	exec()
filter()	float()	format()
frozen set()	getattr()	globals()
hasattr()	hash()	help()
hex()	id()	input()
int()	isinst ance()	issubc lass()
iter()	len()	list()
locals()	map()	max()
memory view()	min()	next()
object()	oct()	open()
ord()	pow()	print()
property()	range()	repr()
reversed()	round()	set()
setattr()	slice()	sorted()
static met hod()	str()	sum()
super()	tuple()	type()
vars()	zip()	

Python incluye las siguientes funciones y siempre están disponibles

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

type()

```
x = 3.1415
print( typ e(x))
>>< class 'float '>
```

La función type permite comprobar el tipo de variable

Variables Numericas

```
num_entero = 5
num_ne gativo = -7
num_real = 3.14
num_co mplejo = 3.2 + 7j
num_bi nario = 0b111
num_octal = 0o10
num_hex = 0xff
```

puedes crear variables del tipo Enteros, Reales, Complejos y los puedes representar en Decimal, Binario, Octal y Hexadecimal

Conjunto Matematico funcion set()

```
conjunto = set('246')
conjunto2 = {2, 4, 6}
```

se pueden utilizar los métodos *add()* y *remove()* para añadir o eliminar elementos.
si se crea un conjunto con valores repetidos, estos se eliminan automáticamente.

Operadores Comparadores

and	compara 2 elementos y devuelve True si ambos son verdaderos
or	compara 2 elementos y devuelve True si uno de ellos es verdadero
not	devuelve el valor opuesto de un boleano

primero se calcula *not*
después se calcula *and*
por último se calcula *or*

Definiciones

Iteración	Término general para la toma de cada elemento de algo, una después de la otra. Usar un bucle, explícita o implícita, al pasar sobre un grupo de elementos

Metodos Especiales para Strings

find() Retorna el indice del primer carácter que coincide con el buscado

```
cad = " ABC "
cad.fi nd( " B")
>>1
```

replace() reemplaza un carácter por otro

```
cad.re pla ce( " B", " Z")
>>AZC
```

split() divide una cadena basado en un caracter y retorna una lista

```
cad.sp lit (";")
```

join() retorna una cadena donde los valores son separados por un caracter

```
lista = ["Ho la", " Mun do"]
print ("+".j oi n(l ista,))
lista2 = " Hol a"
print ("-".j oi n(l ista2))
```

strip(), **lstrip()**, **rstrip()** eliminan los espacios en blanco, a la izquierda y a la derecha respectivamente

```
cad.st rip()
cad.ls trip()
cad.rs trip()
```

Tabla Basica

Tupla	( )	Inmutable
Lista	[ ]	Mutable
Diccionario	{ }	Mutable

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Tupla

Arreglo de objetos definido entre paréntesis es inmutable puede contener diferentes tipos de objetos.

```
tupla = (1, 'a', 3.5)
```

Se puede anidar una tupla dentro de otra

```
tupla2 = (1, (4, 'B'), 3.5)
```

Se puede acceder a los valores a través del indice.

```
tupla[1]
```

Lista

Arreglo de objetos definido entre corchetes es mutable puede contener diferentes tipos de objetos.

```
lista = [2, 'B', 4.5]
```

Se puede acceder a los valores a través del indice y reemplazarlos.

```
lista[1] = 'A'
```

Podemos comprobar si un valor existe en una lista usando **in**.

```
'B' in lista
```

se insertan valores al final de la lista con **.append()**

```
lista.a pp end ('n uevo')
```

insertar en una posición definida se utiliza el indice y **.insert()**

```
lista.i ns ert(2, 'C')
```

borrar un elemento usamos **del()**

```
del(li sta[1])
```

ordenar sin alterar **sorted()** y para orden inverso argumento **reverse**

```
sorted (lista)
sorted (lista, revers e=True)
```

ordenar con criterio como argumento

```
sorted (lista, key=st r.l ower)
```

ordenar alterando usamos **sort()**

```
lista.s ort()
```

Comprensión de Listas y Diccionarios

Compresión Lista

```
lista= [x for x in (1,2,3)]
```

Compresión Diccionario

```
diccio nario= {k: k+1 for k in (1,2,3)}
```

La comprensión es una construcción sintáctica de python, permite declarar una lista o diccionario a través de la creación de otra.

For y While

El bucle **while** (mientras) ejecuta un fragmento de código mientras se cumpla una condición.

```
edad = 0
while edad < 18:
    edad = edad + 1
    print "Felic idades, tienes " + str(edad)
```

Permiten ejecutar un mismo fragmento de código un cierto número de veces, mientras se cumpla una determinada condición.

If, Else y Elif

Evalúan la condición indicada y ejecutan una instrucción u otra

```
if condic ion1:
```

si condicion1 es True realiza esto

```
elif condic ion2:
```

si condicion2 es True realiza esto

```
else:
```

si ambas condiciones son False realiza esto

se pueden anidar

```
if condic ion1:
```

si condicion1 es True realiza esto

```
    if condic ion3:
```

si condicion3 es True realiza esto

```
    else:
```

si es False realiza esto

```
else:
```

en caso contrario realiza esto

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Clases, Objetos, Propiedades y Metodos

*Clases
*Objetos
*Propiedades
*Métodos

```
class Clase(): # La clase
    va rCl ase=0 # Variables de Clase
    def __init __( self): # Método de Instancia (constructor)
        s elf.va rIn sta ncia=0 # Variable de Instancia
objeto =Cl ase()
objeto.me tod oin sta ncia()
@class method # Decorador Metodo de Clase
    def clsmet (cls): # Obligatorio (cls)
Clase.c lsmet
```

Self hace referencia a si mismo
__init__ constructor para inicializar los objetos a un valor
al colocar *(clase)* se habilita la hereda los objetos de la clase Persona

Palabras reservadas

and	as	assert
break	class	continue
def	del	elif
else	except	False
finally	for	from
global	if	import
in	is	lambda
None	nonlocal	not
or	pass	raise
return	True	try
while	with	yield

Estas palabras no pueden utilizarse para nombrar variables.

Patrones caracteres

\n	Nueva Linea
\r	Retorno de carro
\t	Tabulador Horizontal
\w	Caracter minuscula
\W	Caracter Mayuscula
\s	Engloba minusculas y mayusculas
\S	cualquier caracter que no es espacio en blanco
\d	numero entre 0 - 9
\D	cualquier carácter que no es un numero
^	Inicio de cadena
$	Fin de cadena
``	Escape caracter especial
[]	rango de caracteres dentro de corchetes
^[]	cualquier caracter fuera de corchetes
\b	separacion entre numero y/o letra
{{Metacaracter}}	repeticiones
+	una o mas veces
*	cero o mas veces
?	cero o una vez
{n}	n numero de veces

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com