



# Reforzamiento Programación

Preparando la PEP 3  
PARTE I

# Links importantes que deben tener en cuenta!

1. Python Tutor: <http://pythontutor.com/visualize.html#mode=edit>
2. PEP 8 Online: <http://pep8online.com/>
3. Funciones nativas en Python:  
<https://docs.python.org/es/3/library/functions.html>

**En mi drive varios resúmenes y cosas que pueden ayudarles!**

**Link (Correo USACH):**

**[https://drive.google.com/drive/folders/1CifZ\\_2vfwwx8Qolk1NrXXsavAZ0FpjvF?usp=sharing](https://drive.google.com/drive/folders/1CifZ_2vfwwx8Qolk1NrXXsavAZ0FpjvF?usp=sharing)**

# *Estructura de un código*

# Bloque de Definición

# Importación de Funciones

# Definición de Constantes

# Definición de Funciones

""

Descripción de la función:

Entrada:

Salida:

""

# Bloque Principal

# Entrada

# Procesamiento

# Salida

# Tipos de datos

Int: Simplemente enteros

int()

Ej: 5

Float: Números decimales

float()

Ej: 5.0

Str: Strings, cadenas de texto

str()

Ej: 'Hola Mundo!'

List: Listas, las cosas que se definen con []

list()

Ej: [6.6, 7.0, 7.0]

Bool: Verdadero o Falso

bool()

Ej: True

Con type() podemos saber el tipo de dato de un elemento

# Operadores Matemáticos

Operación	Operador	<u>Aridad</u>	<u>Asociatividad</u>	Prioridad
Exponenciación	**	Binaria	Por la derecha	1
Identidad	+	Unaria	----	2
Cambio de Signo	-	Unaria	----	2
Multiplicación	*	Binaria	Por la izquierda	3
División	/	Binaria	Por la izquierda	3
División entera	//	Binaria	Por la izquierda	3
Módulo (resto)	%	Binaria	Por la izquierda	3
Suma	+	Binaria	Por la izquierda	4
Resta	-	Binaria	Por la izquierda	4

# Operadores lógicos

operador	comparación
==	es igual que
!=	es distinto de
<	es menor que
<=	es menor o igual que
>	es mayor que
>=	es mayor o igual que

# Operadores lógicos

Negación	not
.....	
Conjunción	and
.....	
Disyunción	or

# Buenas practicas (Variables)

Variables se escriben con minúscula separando por un guion bajo

Ej: nombre\_variable

Constantes siguen la misma regla, pero con mayúsculas!

Ej: PI



# Entrada y salida

`input()`: Todo lo ingresado por `input` es un `string`, a menos de que vaya algún método antes!

`eval()`: Si se coloca antes de un `input` se puede “controlar” el tipo de dato de la entrada. Muy útil cuando se quiere trabajar con listas ingresadas por entrada.

`print()`: Imprime lo que sea, es la salida del código.

# Condicional (if)

if (condicion):

elif (condicion):

else:

**OJO!: Cada if es “independiente”. Si se tienen muchos ifs se van a ejecutar todos**

# Ciclo While

While (condicion):

Contador: Usualmente se usa la letra i. Nos permite acabar el ciclo while.

**OJO con estas MALAS PRACTICAS:**

`while True:`

`break`

# Listas

Son mutables (se pueden modificar sus elementos)

Se utilizan [ ] y se separan los elementos con comas (,)

Ej: lista = [3,4,2,1]

lista [0] = lista [-4] = 3

lista [1] = lista [-3] = 4

# Métodos listas

Concatenar: se utiliza “+”, siempre que sean dos variables de tipo lista, ejemplo:

$a = [1,2,3]$

$b = [4,5,6]$

$a+b = [1,2,3,4,5,6]$

$[1,2]*2 = [1,2,1,2]$

# Métodos listas

**len()**: largo, ejemplo `len(lista) = 4`

**append()**: agregar element al final, `lista.append(5)`

**pop()**, eliminar, sin parámetro elimina el último elemento, `lista.pop()`. Con parámetro, elimina elemento posición dada. `lista.pop(2)`

**index()**, primera posición (habiendo uno o más) elemento dado, `lista.index(2)`

**remove()**, elimina elemento, a diferencia del pop, elimina el elemento no por la posición. `lista.remove(1)`

# Métodos listas

**Insert()**, recibe dos parámetros, primero, posición a insertar, 2do elemento a insertar. `lista.insert(2,9)`

**sort()**: Ordena una lista de menor a mayor, ejemplo: `lista.sort()`

**count()**: Cuenta la cantidad de ocurrencias de un elemento en una lista, ejemplo: `lista.count(1) = 1`

# Strings

Son inmutables!

**upper():** a mayúsculas, `string.upper()` = “HOLA”

**lower():** a minúsculas, `string.lower()` = “hola”

**strip()** = elimina espacios en blanco y saltos de línea iniciales y finales, ejemplo  
`string = “ hola mundo \n”, string.strip() = “hola mundo”`



# Algunos condicionales conocidos con strings

`isalpha()`: Alfabetico

`islower()`: Minusculas

`isupper()`: Mayusculas

`isdigit()`: Digitos

**PEP 1**