CHULETAS PYTHONERAS

LISTAS	
lista.append('a')	Añadimos el elemento 'a' **al final** de lista.
lista.extend(['a','b'])	Añadimos los elementos 'a' y 'b' **al final** de la lista
lista.insert(i,'c')	Insertamos el elemento 'c' en la posición i-1. Ojo! que la posición exista.
lista.remove('a')	Elimina la **primera** aparición del elemento 'a'.
lista.pop([i])	Devuelve y elimina el elemento en la **posición i**. Si no indicamos i, lo hará con el **último elemento**.
lista.index('a')	Devuelve el **índice** del **primer** elemento que coincide con 'a'. Ojo 'a' debe existir!
lista.count('a')	Devuelve el **número de veces** que 'a' aparece en la lista.
lista.sort()	**Ordena** la lista.
lista.reverse()	**Invierte** el orden la lista.

CADENAS	
cadena.capitalize()	Copia de la cadena con la **primera posición en mayúsculas**.
cadena.count('ab')	**Número de veces** que la sub cadena 'ab' aparece.
cadena.find('ab')	**Índice** de la **primera posición** donde aparece la sub cadena 'ab'
cadena.replace('old','new')	**Reemplaza** la sub cadena 'old' dentro de cadena por 'new'.
cadena.split('del')	**Divide** la cadena según el delimitador. **Ojo devuelve una lista**.
cadena.joint(['cad1', 'cad2'])	**Concatena** cad1 y cad2 en los **extremos** de la cadena.
cadena.strip()	**Elimina** los **espacios en blanco** al **principio** y al **final** de la cadena.

FUNCIONES NATIVAS	
print("hola") print var1 print "variable 1 %d" %var1 print "variable1: ", var1	**Imprimir** por la salida estándar
type(var1)	Devuelve el **tipo** del objeto (entero, flotante, etc.)
len(lista1)	Longitud de una secuencia (listas y cadenas)
range(0, 10)	[0, 1, 2, 3, 4, 5, 6, 7, 8, **9**]
xrange(0, 10)	Igual que range pero más rápido.
id(var)	**Identificador** en memoria del objeto.
int(), float(), list(), dict(), etc.	**Conversión** de tipo.
max(lista1), min(lista1)	Devuelve el **máximo** o **mínimo** de la lista.
zip([10, 11, 12], [20, 21, 22])	Devuelve [(10, 20), (11, 21), (12, 22)]
map(func, lista1)	Llama a la **función func()** pasando como **entrada los elementos de lista1**.
reduce(func(a, b), lista1)	Llama a la función func(a, b), siendo **'a' el valor acumulado** y **'b' el elemento actual de la lista**.
filter(func(a), lista1)	**Incluye los elementos para los func(a) devuelve True**.
for i, j in enumerate([10, 11, 12]):	i, j = 0, 10; 1, 11; 2, 12
sorted(lista1)	Devuelve un **objeto nuevo** que corresponde a lista1 **ordenada**.
input()	Paso de **valores por teclado**.
raw_input()	Paso de **valores por teclado**. Ojo todo se considera una cadena.
dir(objeto)	Devuelve la lista con los **métodos del objeto**.

TUPLAS	
tupla.count('a')	Cuenta el **número de veces** que 'a' aparece en la tupla.
tupla.index('a')	Devuelve el índice de **la primera aparición** de 'a'. Ojo la tupla debe contener un elemento 'a'.

Realizado por Daniel Gutiérrez d.gutierrez.reina@gmail.com

CHULETAS PYTHONERAS

DICCIONARIOS	
dict.items()	Devuelve una **lista** con el formato [(key1, value1), (key2, value2),…]
dict.keys()	Devuelve la **lista** de las **keys**.
dict.values()	Devuelve la **lista** de **values**.
dict.copy()	**Copia** de forma **superficial**, **no apuntan** a la misma dirección de memoria.
dict.pop(key1)	**Elimina la pareja key1-value1** del diccionario. **Ojo** que la key1 exista!
dict.clear()	**Elimina todos** los componentes del diccionario.

BUCLES

```
for  i in secuencia:
    # cuerpo del bucle!
    # i toma los elementos de la secuencia!
    # no olvidar tabular!
    # no olvidar dos puntos!
    # Secuencias son listas y cadenas!
# Aquí acaba el bucle
```

```
while condicion:
    # cuerpo del bucle!
    # no olvidar tabular y los dos puntos!
# Aquí acaba el bucle
```

break	Salimos del bucle
continue	Saltamos a la siguiente iteración
pass	No hace nada, TODO

List comprenhension:

```
[x*x for i in [1, 2, 3, 4]]
```

Resultado:

```
[1, 4, 9, 16]
```

CONTROL DE FLUJO

```
If condicion:
        # si se cumple la condición, Ojo tabular!
else:
        # si no se cumple la condición, Ojo tabular!
```

IMPORTAR LIBRERÍAS/MÓDULOS

import random	random.uniform(0, 10)
import random as rd	rd.uniform(0, 10)
from random import uniform	uniform(0, 10) # Ojo sólo uniform **el resto no está disponible**.
from random import*	Uniform(0, 10) # **Todos** los métodos y objetos disponibles.

FUNCIONES

```
Def nombre_func(var1, var2):
    """ Docstring
    """
    # cuerpo de la función!
    # no olvidar tabulación!
    # no olviadar dos puntos!
    return  resultado # opcional
```

```
lambda x: x * x
```

MUTABILIDAD

Enteros	Inmutable
Flotantes	Inmutable
Booleanos	Inmutable
Listas	mutables
Tuplas	Inmutables
Diccionarios	mutables

Realizado por Daniel Gutiérrez d.gutierrez.reina@gmail.com

CHULETAS PYTHONERAS

MANEJO DE ARCHIVOS	
Archivo = open("myfile.txt", r)	Abrir archivo myfile.txt en **modo lectura.**
Archivo = open("myfile.txt", w)	Abrir archive myfile.txt en **modo escritura.**
Archivo.close()	**Cerrar archivo** tanto en lectura como escritura.
Archivo.write(str1)	**Escribe str1** en el archivo.
Archivo.readlines()	Devuelve una **lista** que contiene todas las **líneas del archivo.**
for linea in Archivo: # trabajar con linea	**Recorrer todas las líneas** del archivo.

FUNCIONES MÁGICAS	
%clear	**Limpia** el terminal (los objetos no)
%exit	**Elimina** los objetos
%run archivo.py	**Ejecuta** el script
%history	**Historial** de comandos
%cd "carpeta"	**Cambia al directorio** carpeta
%"comando"?	**Ayuda**
PIP	
!pip install modulo	**Instala** módulo/librería
!pip upgrade modulo	**Actualiza** módulo/librería
!pip show version	Ver **Versión** módulo/librería

CLASES	

class myclase (object):
"""Docstring
"""
def __init__(self,atr1, atr2):
# aquí empieza el constructor 	
self.atr1 = atr1
self.atr2 = atr2
def metodo1(self):
# cuerpo del método 1 	
return (self.atr1 + 2)
def metodo2(self):
# cuerpo del método 2 		

Obj1 = myclase(10, 20)	**Definir e inicializar** el objeto
Obj1.atr1	**Acceso** a los **atributos**
Ob1.metodo1()	**Acceso** a los **métodos**

SLICING	
secuencia[0]	**Primer** elemento
secuencia[-1]	**Último** elemento
secuencia[:]	**Todos** los elementos
secuencia[a:b]	Elementos en el **intervalo [a, b)**
secuencia[a:b:p]	Elementos en el **intervalo [a, b)** con un **paso p**

Realizado por Daniel Gutiérrez d.gutierrez.reina@gmail.com