Cheatography

Starting with Python Cheat Sheet

by chembembere via cheatography.com/76595/cs/18871/

Comentarios

Comentar código	# <texto comentado>

Listas básico

Acción	Código
Crear una lista vacia	miLista = []
Crear una lista con valores	miLista=[<valor1>, <valor2>,...]
Añadir un valor a una lista	miLista.append(-<valor>)
Crear una lista de listas en base a 2 listas	list(zip(lista1, lista2))
Añadir una lista a otra con +	lista1+['valor1',...,'valorN']
Crear una lista consecutiva de números	range (<número>) (de 0 a número -1)

Operaciones con listas

Operación	Código
Tamaño de una lista	len(miLista)
Seleccionar un elemento	miLista[index]
Seleccionar el último elemento	miLista[-1]
Obtener sublista	miLista [comienzo:final] (ojo, final no se incluye)
Obtener los N primeros elementos de una lista	miLista[0:N] o miLista[:3]

Operaciones con listas (cont)

Obtener los los elementos desde la posición N hasta el final	miLista[N:]
Obtener los N últimos elementos de una lista	miLista[-3:]
Contar la aparición de un elemento en una lista	miLista.count('<elemento>')
Ordenar los elementos de una lista	miLista.sort() (no genera lista, actua sobre la lista original)
	miLista2=sorted(miLista)
Obtener el índice de un elemento en una lista	miLista.index(<elemento>) (Devuelve el índice de la primera aparición del elemento)
Ver si un elemento está en una lista	If element in miLista (True si está)

Loops para recorrer listas

Bucle for

```
for <variable> in <lista>:
  <código>
```

Bucle for usando índice en vez de valor

```
for i in range (len(<lista>)):
  código
```

Funciones para trabajar con listas

Función	Explicación
range()	range(10): Crea una lista de 0 a 9
	range (2,9): Crea una lista de 2 a 8
	range (2,9,2): Crea una lista de 2 a 9, pero saltando 2 posiciones
index()	milista.index(<elemento>) Devuelve el primer índice en el que se encuentra el elemento indicado, si no encuentra nada devuelve un ValueError

List Comprehensions

Para crear una list comprehension se usa:

```
usernames = [word for word in words if word[0] == '@']
```

Explicación:
Toma el elemento de la lista words en word.
Si el primer elemento de la variable es @ entonces lo añade a la lista
Si no, no hace nada.

Duplicar una lista

```
usernames = [word for word in words]
```

Explicación:
Al no haber condición, duplica toda la lista

Funciones

Para definir una función:

```
def <nombre>(<parámetro1>, <parámetro 2>...<parámentro n>):
  <código>
```


By **chembembere**

cheatography.com/chembembere/

Not published yet.
Last updated 12th May, 2019.

Funciones (cont)

```
  return <salida>
```

Argumentos

Positional arguments: their assignments depend on their positions in the function call.
keyword arguments: where we explicitly refer to what each argument is assigned to in the function call.

```
def create_spreadsheet(title, row_count=1000):
```

Una vez que se usan argumentos con keyword, detras de este argumento tienen que ser keyword.
El caso de que tenga argumentos con keyword, puedo llamar la función sin especificar esos parámetros que usan el valor por defecto.

Devolver valores

```
return (<valor>)
```

Si quiero devolver un texto más un valor, debo convertir primero el valor en string

```
return ("EL valor es"+ str(<valor>))
```

Multiples valores:

```
def square_point(x_value, y_value):
  x_2 = x_value * x_value
  y_2 = y_value * y_value
  return x_2, y_2
x_squared, y_squared = square_point(1, 3)
print(x_squared)
print(y_squared)
```

Impresión de resultados

Función	Efecto
print()	Imprime el contenido entre paréntesis

Funciones de transformación

Función	Efecto
str()	Convierte un tipo numérico en string. Muy útil cuando se quieren imprimir números junto con strings mediante el print()

Comandos de linea de comandos

Comando	Efecto
ls	Listar un directorio
pwd	Indica el nombre del directorio actual
cd	Cambiar directorio
mkdir	Crear directorio
touch	Crear fichero

try...except

Try ...except nos permiten capturar un posible error y hacer algo al respecto en vez de bloquear el programa.
La sintaxis es:

```
try:
  some_thing = can_trigger_a_syntax_errror() #o cualquier error
except SyntaxError:
  print("Error caught!")
```

Operadores Numéricos

Operador	Efecto
+	Suma
-	Resta
*	Multiplicación
/	División
a**b	Exponente
a%b	Módulo
a+=b	a= a+b

Operaciones con Strings

Operador	Efecto
"a"+"b"	Concatenar
'''<string>'''	Multi line string (3')

Operadores para el control de la ejecución

Operador	Efecto
==	Igual a
!=	No igual
>	Mayor que
>=	Mayor o igual
<	Menor que
<=	Menor o igual que

Bloque IF

```
If (condicion):
   <codigo>
elif (condicion):
   <codigo>
else:
   <codigo>
```

By **chembembere**

cheatography.com/chembembere/

Not published yet.
Last updated 12th May, 2019.

Métodos de ristras

Método

Resultado

"Hello world".upper()

"HELLO WORLD"

"Hello world".lower()

"hello world"

"Hello world".title()

"Hello World"

"Hello world".split()

['Hello', 'world'], podemos indicar el caracter por el que separar e incluso usar \n para indicar salto de linea o \t para indicar tabulador

" ".join(['Hello', 'world'])

"Hello world"

"Hello world".replace("H", "J")

"Jello world"

" Hello world ".strip()

"Hello world"

"{} {}".format("Hello", "world")

"Hello world"

"Hello world".find("w")

6 (indice donde la primer "w" se encuentra, si no encuentra nada devuelve -1)

Modulos

Importar un módulo:
from <nombre módulo> import <object>
Si quiero importar toda una librería:
import <nombre>
Para llamar a una función de un módulo:
<modulo>.function()
Para dar un alias a un módulo:

Modulos (cont)

import <modulo> as <alias>
Ver los métodos de un módulo:
import math
print(dir(math))w
Importar una función de otro fichero:
from <fichero> import <método>

Métodos de Random

```
# shuffle() will shuffle a sequence in place
arr = [1, 2, 3, 4]
random.shuffle(arr)
print(arr) # [3, 1, 4, 2]
# random() will return a random float value between
# 0.0 (inclusive) and 1.0 (exclusive)
print(random.random()) # 0.237...
print(random.random()) # 0.441...
# choices() is similar to choice(), but can return a list of k elements
# from a list, with possibly repeating values.
arr = [1, 2, 3, 4, 5]
print(random.choices(arr, k=3))
# [1, 1, 4]
```

Diccionarios

Definición:

Conjuntos de pares *clave* : *valor*

```
menu = {"oatmeal": 3, "avocado toast": 6, "carrot juice": 5, "blueberry muffin": 2}
```

Un diccionario no puede tener 2 claves con el mismo valor.
Una Tupla puede ser una clave para un diccionario ya que son valóres únicos inamovibles.

Diccionarios (cont)

Declarando un diccionario

```
empty_dict = {}
```

Ver si un diccionario está vacio

Opción 1:

```
dict1 = {}
if dict1:
    print("dict1 Not Empty")
else:
    print("dict1 is Empty")
if bool(dict1):
    print("dict1 Not Empty")
else:
    print("dict1 is Empty")
```

Opción 2

```
dict1 = {}
if len(dict1) == 0:
    print("dict1 is Empty")
```

Añadir valores a un diccionario

Opción 1

```
my_dict["new_key"] = "new_value"
```

Opción 2

```
sensors.update({"pantry": 22, "guest room": 25, "patio": 34})
```

List comprehension

```
drinks = ["espresso", "chai", "decaf", "drip"]
caffeine = [64, 40, 0, 120]
zipped_drinks=zip(drinks, caffeine)
drinks_to_caffeine={key:value for key, value in zipped_drinks}
```

By **chembembere**

cheatography.com/chembembere/

Not published yet.
Last updated 12th May, 2019.

Operaciones con diccionarios

Leer un valor de un diccionario

```
diccionario["key"]
```

Si 2 keys tienen diferente tipo, incluso si su valor es igual, se tratan como keys diferentes:

mydict = { }
mydict[5] = "value1"
mydict['5'] = "value2"
print(mydict)
# {5: 'value1', '5': 'value2'}

Si intentamos acceder a una key qie no está en el diccionario, nos devolverá un error.

Try/Except para leer una key

```
try:
print(caffeine_level["matcha"])
except KeyError:
print("Unknown Caffeine Level")
```

Si quieres saber qué *key* es la que provoca el error, puedes usar este código

try:
utah_list = population['Utah']
except KeyError as k:
print("Key " + str(k) + " does not exist")

Comprobar si existe una *key* en el diccionario

```
key_to_check = "Landmark 81"
if key_to_check in building_heights:
print(building_heights["Landmark 81"])
```

Obtener una *key* con get

building_heights = {"Burj Khalifa": 828, "-Shanghai Tower": 632, "Abraj Al Bait": 601, "Ping An": 599, "Lotte World Tower": 554.5, "One World Trade": 541.3}

Operaciones con diccionarios (cont)

#this line will return 632:
building_heights.get("Shanghai Tower")
#this line will return None:
building_heights.get("My House")

Se puede usar el get sin asignar a una variable, por ejemplo para condiciones. Además se puede indicar un valor por defecto (número o string)

```
stack_id=user_ids.get("superSt-ackSmash", 100000)
```

Extraer un valor de un diccionario con pop()

```
midiccionario.pop(320291, "No Prize")
```

Si no se pone valor por defecto y no se encuentra la clave se devuelve un KeyError exception

By **chembembere**

cheatography.com/chembembere/

Not published yet.
Last updated 12th May, 2019.