Python Cheat Sheet

Xaloc

Python Basics:

- Hints:

- Be careful with blank spaces! They can make a big difference in the code.
- Your code will not run without the correct indentation!
- # This is a comment - use it to make a one line comment or to comment out a line
- """
  Everything in between three quote marks will be considered a comment - it can be used to make comments that span more than one line with line breaks in them
  """

- Numbers:

Python uses integer and float numbers. You can use the type function to check the value of an object:

```
type(3)      returns: <type 'int'>
type(3.14)   returns: <type 'float'>
```

- Inputs:

A = input()	Waits for you to enter some characters and saves them in A
B = int(input())	Waits for you to enter integers and saves them in B
input("Press ENTER")	Waits for you to press ENTER to continue - since there is no variable declared it won't save anything.
A = input("message")	Prints "message" and waits for you to enter a value that will be saved in A

Basic logic in Python

- if

- if condition1:
  .....# do something if condition1 is true
  elif condition2:
  .....# do something if condition2 is true
  else:
  .....# do something if both are false

- while:

- while condition:
  .....# while condition is true keep doing something, make sure that the condition will be false at some point

- for:

- for x in sequence
  .....# for x in the given sequence
  .....# do something for every item
  .....# the sequence can be a list,
  .....# elements from a string, etc.
- for x in range(10)
  .....# repeat something 10 times (from 0 to 9)
- for x in range(5,10)
  .....# repeat something 5 times (from 5 to 9)

- Logic tests

10 == 10	returns: True
10 == 11	returns: False
10!= 11	returns: True
"jack" == "jack"	returns: True
"jack" == "jake"	returns: False
10 > 10	returns: False
10 >= 10	returns: True
"abc" >= "abc"	returns: True
"abc" < "abc"	returns: False

Python Lists

- Python Lists

Lists are made form elements of any type (they can alternate types)

Using Lists in Python

Creation	
a_list = [5,3,'p',9,'e']	creates: [5,3,'p',9,'e']
Accessing items	
a_list[0]	returns: 5
Slicing	
a_list[1:3]	returns: [3,'p']
Length	
len(a_list)	returns: 5
count(item)	
returns how many times the item was found in the list.	
count(a_list('p')	returns: 1
Sorting - sort()	
a_list.sort()	returns: [3,5,9,'e','p']
Sorting without altering the list	
print(sorted(a_list))	returns: [3,5,9,'e','p']
Adding - append(item)	
a_list.append(37)	returns: [5,3,'p',9,'e',37]
Inserting - insert(position,item)	
insert(a_list.append(3),200)	returns: [5,3,200,'p',9,'e']
Retornar e remover - pop(position)	
a_list.pop()	returns: 'e' and the list becomes [5,3,'p',9] - deletes last element
a_list.pop(1)	returns: 3 and the list becomes [5,'p',9,'e'] - deletes element 1
Delete - remove(item)	
a_list.remove('p')	returns: [5,3,9,'e']
Insert	
a_list.insert(2,'z')	returns: [5,'z',3,'p',9,'e'] - insert in given position
Invert - reverse()	
reverse(a_list)	returns: ['e',9,'p',3,5]
Concatenating	
a_list+[0]	returns: [5,3,'p',9,'e',0]
a_list+a_list	returns: [5,3,'p',9,'e',5,3,'p',9,'e']
Find	
9 in a_list	returns: True
for x in a_list	returns the whole list, one element per line
.....print(x)	

Python Cheat Sheet

Xaloc

Other Elements

- Key words

Oper.	Description
print	prints on the screen
break	stops a loop if necessary
continue	restarts loop ignoring commands below
is	Tries an object identity
def	Used to create a new function defined by the user
return	Exit the function and returns a value
global	Access variables defined globally (outside of a function)
del	Deletes objects

- Libraries

Libraries are a collection of functions and methods that allow you to perform many actions without writing your code

Using Libraries in Python

import	imports a library inside a script
as	gives an alias to the library
from	imports a specific function from a library

Useful Libraries

numpy	library for maths
matplotlib	library for plotting
tkinter	library to create GUI
random	library for random numbers

- Functions

- def f_name(args):
  .....# operations done by the function
  return value_to_return

- Dictionaries

A dictionary is a list of keys and values, where you can access a value by its key. All keys must be different.

```
car = ["brand" : "Hummer",
       "model":"H2",
       "year":2009,]
```

- Tuples

A tuple is a list of values separated by a comma - very similar to a list but tuples are immutable (you are not allowed to change their values):

a_tuple = ('a','b','c')

Python Operators

Lets take a=10 and b=20 as example:

- Arithmetic operators

Op.	Description	Example
+	Addition	a + b returns: 30
-	Subtraction	a - b returns: -10
*	Multiplication	a * b returns: 200
/	Division	b / a returns: 2
%	Module	a % b returns: 0
**	Exponential	a**b returns: 10^{20}
//	Euclidean Division	9 // 2 returns: 4

- Comparison Operators

Basic comparation opration can be used in different ways for any type of value - numbers, strings, sequences, lists, etc. The answer will always be True or False.

Op.	Description	Example
<	Less than	a < b returns: True
<=	Less or equal	a <= b returns: True
==	Equal	a == b returns: False
>	Greater than	a > b returns: False
>=	Greater or equal	a >= b returns: False
!=	Different	a != b returns: True

- Logic Operators

The logic operators **and** and **or** Also return a Boolean value when used in a decision structure.

Op.	Description
and	If the result of both sides is true, returns: True
or	If one of the results on either side is true, returns: True
not	It is used to invert the result of any Boolean operation.

- String Operators

Using a=['Hello'] and b=['Python']

Oper.	Example
+	a + b returns: HelloPython
*	a*2 returns: HelloHello
.[ ]	a[1] returns: "e"
.[:]	a[1:4] returns: "ell"
in	H in a will give 1
not in	M not in a returns: 1

Examples

A few examples to see how actual python code looks like.

Simple Hello World with print

```
print("Hello World!")                    #prints Hello World!
```

Hello World with function

```
def hello():                             #Function definition
    print("Hello World!")

hello()                                  #Function called and Hello World! printed
```

Hello your name

```
name = input("Enter your name:")# Asks you to enter your name
print("Hello "+name)                     # Prints Hello and the value in name
```

Using a library

```
import random as rand                    #imports library random
                                         #sets library alias to rand
size = int(input("Enter dice size:")) #Asks for dice faces
print("You rolled a ",rand.randint(1,size))
                                         #Prints result of roll using randint
                                         #from the random library
```

if loop

```
x = float(input("Enter any number:")) #Aks number and stores in x
if x > 5.3:                              #checks if x>5.3
    print(x, " bigger than 5.3")         #only if previous condition True
elif x < 5.3:                            #checks if x<5.3
    print(x, " smaller than 5.3")        #only if previous condition True
else:                                    #if 1st and 2nd false
    print(x, "equals 5.3")               #prints equal
```

while loop

```
x = 10                                   #stores 10 in x
while x>0:                               #if x>0 enters loop
    print(x)                             #prints value of x
    x-=1                                 #substracts 1 to x same as x=x-1
                                         #result: 10,9,8,7,6,5,4,3,2,1
```

for loop

```
x = "hello"                              #stores hello in x
for c in x:                              #repeats loop for every character of x
    print(c)                             #prints current character
                                         #result: h,e,l,l,o
```