Python 3.x Cheat Sheet

Sample code can be found at:

https://github.com/magdapoppins/python-course-basics

Find me on Twitter:

https://twitter.com/magdapoppins

Variables

Defining a variable:

name = “Nina”

You can also change that value later:

name = “Niilo”

Printing a variables value

message = “Hello, Python!”

print(message)

Getting user input

name = input(“What is your name?”)

** Note that the return value of input() will be of type String!*

Data types

Name	**Example values**
String	“Hello”, “test”, “123”
Int	2, 5, 100
Float	2,55, 7,544, 100,1
Boolean	True, False

Converting to other data types

Converting string to int:

age = int(input(“How old are you?”))

Converting int to string:

print(“I am “ + str(age) + “ years old.”)

Arrays

Arrays are collections of values. The values are surrounded by square brackets.

friends = [“Ana”, “Lee”, “Minna”, “Hasan”]

ages = [1, 3, 6, 3, 2]

New values can be added using .append(new_value).

friends.append(“Luke”)

Arrays are mutable, i.e. you can modify their contents.

friends[2] = “Nora”

Arrays can be multidimensional.

dataset = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]

Indexing

The values of the array can be accessed by indexing. Indexes start from zero.

0	1	2	3
“Ana”	“Lee”	“Minna”	“Hasan”

friends[0] is “Ana”
friends[2] is “Minna”
friends[-1] is “Hasan”

friends[2:3] is [“Minna”, “Hasan”]
friends[1:] is [“Lee”, “Minna”, “Hasan”]
friends[:1] is [“Ana”]

Conditions

If and elif (else if) are followed by a statement that evaluates to True or False. If there is no match, else will be executed.

```
height = 140

if height > 150:
        print(“You can ride the rollercoaster!”)
elif height > 130:
        print(“You can ride if you have a parent with you.”)
else:
        print(“Sorry, you cannot ride this rollercoaster.”)
```

Logical operators

6 == 6	Equals
“Hello” != “Hi	Not equals
5 < 6	Larger/smaller than
5 <= 5	Larger/smaller than or equal

not
or
and

Functions

Defining a function

In this function, calculate_area is the function name and length and width are its parameters.

```
def calculate_area(length, width):
    return length * width
```

Calling a function

The function must be defined before you can call it.

```
area = calculate_area(2, 10)
```

Loops

For-loops

```
for name in friends:
    print(“Hello, “, name)


for i in range(10):
    print(“This is round:”, i)
```

While-loops

```
while True:
    hobby = input(“Tell me one of your hobbies!:”)
    hobbies.append(hobby)
    break
```

Exceptions

Some common Python exceptions are:

SyntaxError	Invalid syntax
TypeError	Wrong data type (for example using a String as an Int).
ImportError	Trouble with importing a module.
IndexError	Trying to access an index that does not exist, for example index 6 of a list with 4 values.
AttributeError	Attribute reference or assignment fails.

When you get an error message:

1) What line does it occur on?
2) What is the exception type?

Try Except

```
try:
        print(age + 1)
except TypeError:
        print(“Wrong data type for age!”)
except:
        print(“Somenthing went wrong!”)
```

Modules

Modules are ready made pieces of code that you can use in your own projects. They can be imported using the import keyword.

```
import my_module_name
```

Random

```
import random
```

Choose a random integer between 0 and 10.

```
my_random_number = random.randint(0, 10)
```

Choose a random value from an array.

```
colors = [“green”, “blue”, “yellow”]
my_random_color = random.choice(colors)
```

OS

```
import os
```

If the file is present, write to it.

```
if os.path.exists(‘C:/flowers.txt’):
        with open(‘C:/flowers.txt’, ‘w’) as flowerfile:
                flowerfile.writelines(“Hello?”)
```

Turtle

```
import turtle

jenny = turtle.Turtle()
my_screen = turtle.Screen()
my_screen.bgcolor(“cyan”)

jenny.forward(100)
jenny.left(90)
jenny.goto(x, y)
jenny.circle(r)
```

```
jenny.penup()
jenny.pendown()
jenny.pencolor("green")
jenny.color("hotpink")
jenny.beginfill()
jenny.endfill()
```

See: https://docs.python.org/3/library/turtle.html

PyGame

```
import pygame

pygame.init()
display = pygame.display.set_mode((100, 100))
character_image = pygame.image.load('cat.png')

def character(x, y):
    display.blit(character_image, (x, y))

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            quit()

    display.fill((255,255,255))
    character(50, 50)
```

See: https://www.pygame.org/

Command line

Step backwards:

cd ..

Step into directory:

cd my-directory

List directory contents:

ls

Make a new directory:

mkdir my-new-directory

Git and version control

Initialize repository

git init

Check repository status

git status

Add all content to coming commit

git add .

Make a commit

git commit -m “What you did.”

Push to remote

git push