Python Scracth Book

DEA

2022-07-04

Contents

Chapter 1

Acknowledgement

Chapter 2

Introduction

Jupyter Lab

to install on mac: `pip3 install jupyterlab` on terminal

to upgrade pip: `pip3 install --upgrade pip` on terminal

check python version

```
!python -V
```

Python

`Python` is what is called an interpreted language. Compiled languages examine your entire program at compile time, and are able to warn you about a whole class of errors prior to execution. In contrast, Python interprets your script line by line as it executes it. Python will stop executing the entire program when it encounters an error (unless the error is expected and handled by the programmer, a more advanced subject that we'll cover later on in this course).

```
# Check the Python Version

import sys
print(sys.version)
```

```
## 3.10.3 (v3.10.3:a342a49189, Mar 16 2022, 09:34:18) [Clang 13.0.0 (clang-1300.0.29.30)]
```

[Tip:] `sys` is a built-in module that contains many system-specific parameters and functions, including the Python version in use. Before using it, we must explicitly import it.

2.1 Types of objects in Python

Python is an object-oriented language. There are many different types of objects in Python. Let's start with the most common object types: strings, integers and floats. Anytime you write words (text) in Python, you're using character strings (strings for short). The most common numbers, on the other hand, are integers (e.g. -1, 0, 100) and floats, which represent real numbers (e.g. 3.14, -42.0).

Object Types

- integer = 10
- float = 10.1
- string = "Hello"
- boolean = True

float to integer

```
a = 10.123
type(a)
```

```
## <class 'float'>
```

```
b = int(a)
b
```

```
## 10
```

```
type(b)
```

```
## <class 'int'>
```

numeric to string

```
a_string = str(a)
a_string
```

```
## '10.123'
```

string to numeric

```
float("1.1")
```

```
## 1.1
```

it does not transform directly to integer here

```
int("1.123")
```

but it works when transforming to float then integer.

```
int(float("10.123"))
```

boolean

```
bl = True
bl
```

```
## True
```

```
type(bl)
```

```
## <class 'bool'>
```

boolean to numeric

`True` becomes 1

```
int(bl)
```

```
## 1
```

numeric to boolean

0 becomes `False` all other numbers are `True`

```
bool(-100)
```

```
## True
```

```
bool(0)
```

```
## False
```

2.2 Expressions: Mathematical Operations

```
5 + 5 * 10  - 2 / 5
```

```
## 54.6
```

integer division

```
11 // 2
```

```
## 5
```

modulo: remainder

```
10 % 3
```

```
## 1
```

2.3 String Operations

-string object are in single quote or double quote

2.3.1 Indices

Positive Index In python **indices** start with 0 In R **indices** start with 1

Negative Index `-1` corresponds to the last element

```
myname = "Davut Emrah Ayan"

print("Object myname:", myname)
# examples
```

```
## Object myname: Davut Emrah Ayan
```

```
print(myname[0], 'is the first element of myname object')
```

```
## D is the first element of myname object
```

```
print(myname[6], 'is the 6th element of myname object')
```

```
## E is the 6th element of myname object
```

```
print(myname[-1], 'is the last element of myname object')
```

```
## n is the last element of myname object
```

2.3.2 Slicing

Full version

object[from : to : increment]

```
print("Object:", myname)
```

```
## Object: Davut Emrah Ayan
```

```
print("From 0 index to 5th index, by 1, is", myname[0:5:1])
```

```
## From 0 index to 5th index, by 1, is Davut
```

Short version

```
print("Object myname:", myname)
```

```
## Object myname: Davut Emrah Ayan
```

```
print(myname[0:5], "is the first 5 element of the object")
```

```
## Davut is the first 5 element of the object
```

Shorter version

Numeric string is easier to see.

```
num = "0123456789"
print(num[::2], "every 2 other element")
```

```
## 02468 every 2 other element
```

```
print(num[::3], "every 3 other element")
```

```
## 0369 every 3 other element
```

```
print(num[::4], "every 4 other element")
```

```
## 048 every 4 other element
```

```
print(num[::5], "every 5 other element")
```

```
## 05 every 5 other element
```

Length of an object : `len()`

it is character length or element length

```
len(myname)
```

```
## 16
```

2.3.3 Concatenate

```
statement = "KU" + " is the best!"
statement
```

```
## 'KU is the best!'
```

```
myname[0:5] + " is the best!"
```

```
## 'Davut is the best!'
```

Multiplication with strings

```
myname[0:6] * 3
```

```
## 'Davut Davut Davut '
```

Strings are Immutable

Chapter 3

Pandas Library

run `pip3 install pandas` on rstudio terminal or mac terminal or jupyter notebook

3.1 read data

```
import pandas as pd
pd.set_option('display.max_columns', None)
```

3.1.1 csv file

IBM sample data: I could not run with "https" because I did not have a certificate installed. So, I go on with "http" and it worked.

```
data_link = "http://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DS01

recipes = pd.read_csv(data_link)
```

3.2 Explore Data

```
#recipes.head()
```

3.2.1 Get the dimensions of the dataframe.

```
#recipes.shape
```

Chapter 4

RANDOM

```
i = int(input("sayi gir = "))

for x in range(i+1) :
  y = x*5
  print(x,"x",5,"=",y)
```

Chapter 5

Applications

Some *significant* applications are demonstrated in this chapter.

5.1 Example one

5.2 Example two

Chapter 6

Final Words

We have finished a nice book.