Python for Data Science and Machine Learning

School Year 2023-2024

IST

Course Structure

1	Introduction to Python	OCT 19	OCT 26	NOV 9	NOV 16	NOV 23	NOV 30
2	Introduction to Data Science	DEC 7	DEC 14	DEC 21	JAN 11	JAN 18	
3	Introduction to Machine Learning	JAN 25	FEB 1	FEB 8	FEB 22		
4	Data Science & ML Challenge	FEB 29	MAR 7				

= Core Topics = Optional Topics

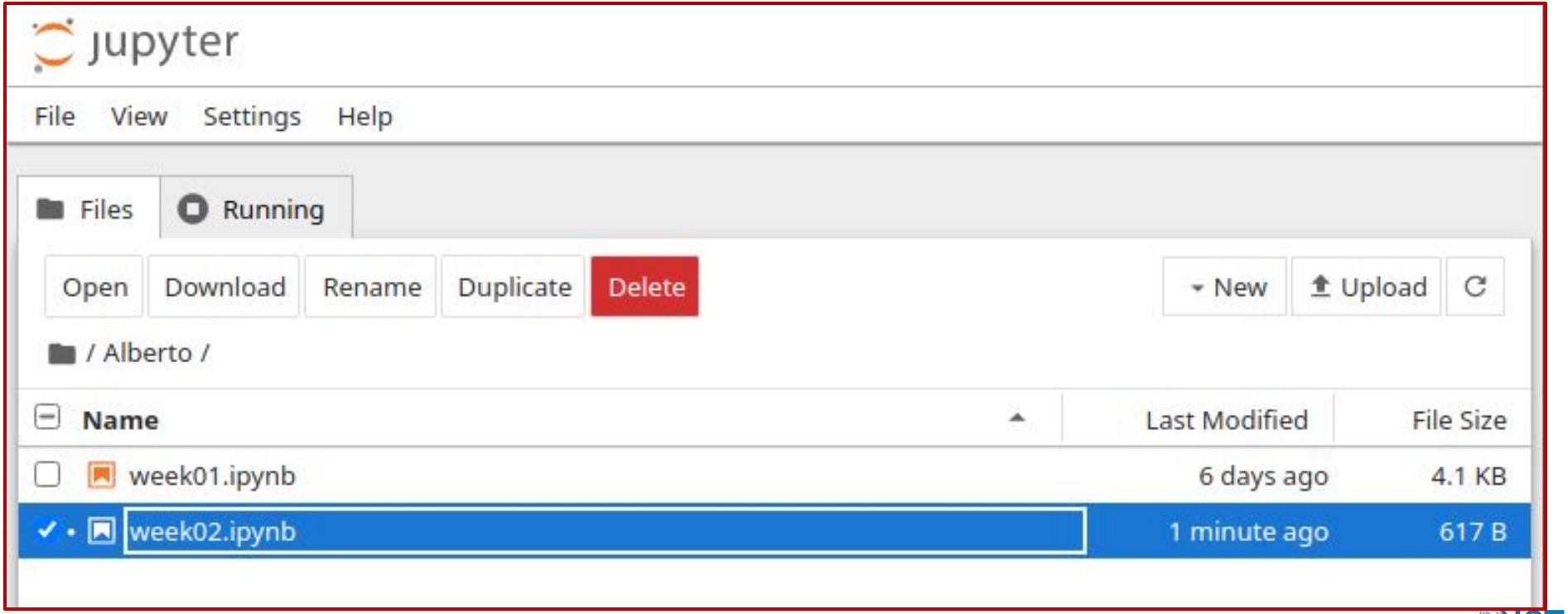
Jupyter Notebook Setup

In a browser:

```
192.168.10.4:8888
```

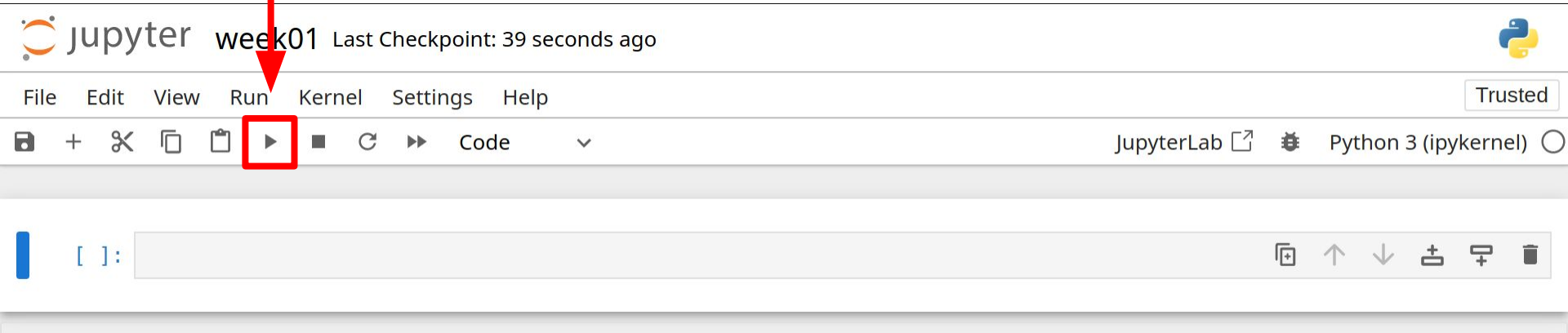
Password: **ist**

Jupyter Notebook Setup



Jupyter Notebook Structure

Run Cell



Recap: Input

The **input** function can be used to take data from the user

```
[*]: x = input("Write something:")
     print("---------------")
     print(x)

     Write something: ↑↓ for history. Search history with c-↑/c-↓
```

```
[1]: x = input("Write something:")
     print("---------------")
     print(x)

     Write something: International School of Turin
     ---------------
     International School of Turin
```

Recap: Comparisons

- 5 is larger than 3

```
5 > 3
```

- -5 is larger than 9

```
-5 > 9
```

- 2 is the same as 2

```
2 == 2
```

- 2 is less than 6

```
2 < 6
```

Recap: Chaining Comparisons

- **not** (negation)

```
not True
```

```
not (5 < 3)
```

- **and** (both must be true)

```
(5 < 6) and (5 < 10)
```

- **or** (either must be true)

```
(5 < 3) or (5 < 10)
```

Recap: If-Statements

Allow for branches in your code!

```
x = 5

if x < 10:
   print("X is small")
else:
   print("X is large")
```

```
x = 20

if x < 10:
   print("X is small")
else:
   print("X is large")
```

NOTE: You **<u>do not</u>** need an else block, it's optional.

More Mathematical Functions

Two new mathematical functions:

- **max** (the largest element)

```
max(10, 15, 4)
```

```
max(2, -2, 0, -3)
```

- **min** (the smallest element)

```
min(10, 15, 4)
```

```
min(2, -2, 0, -3)
```

Exercise

Complete **Ex 3.0** that asks the user for 3 numbers between 0 and 100, if they are valid it prints the max, min and average.

```
Insert the first number: 4
Insert the second number: 8
Insert the third number: 10
Max number: 10
Min number: 4
Average number: 7.333333333333333
```

```
Insert the first number: 4
Insert the second number: 8
Insert the third number: 200
ERROR: One or more numbers are invalid.
```

Exercise Solution

```
x = int(input("Insert the first number:"))
y = int(input("Insert the second number:"))
z = int(input("Insert the third number:"))

if (x < 0) or (x > 100) or (y < 0) or (y > 100) or (z < 0) or (z > 100):
   print("ERROR: One or more numbers are invalid.")
else:
   max number = max(x, y, z)
   min number = min(x, y, z)
   avg_number = (x + y + z) / 3

   print("Max number: " + str(max number))
   print("Min number: " + str(min number))
   print("Average number: " + str(avg_number))
```

If-Statement nesting

You can nest multiple if-statements within each other.

```
x = 5

if x < 10:
    if x < 5:
        print("X is less than 5")
    else:
        print("X is between 5 and 10")
else:
    if x < 15:
        print("X is between 10 and 15")
    else:
        print("X is greater than 15")
```

If-Statement chaining

You can chain multiple conditions with **elif**.

What is the difference between these two snippets of code?

```
x = int(input())

if x < 3:
   print("X is less than 3")
elif x < 10:
   print("X is less than 10")
elif x < 25:
   print("X is less than 25")
```

```
x = int(input())

if x < 3:
   print("X is less than 3")
if x < 10:
   print("X is less than 10")
if x < 25:
   print("X is less than 25")
```

Exercise

Complete **Ex 3.1** that asks the user for a number.

If the number is divisible by 2 it prints “Fizz”.

If the number is divisible by 3 it prints “Buzz”.

If the number is divisible by both 2 and 3 it prints “FizzBuzz”.

```
Insert a number: 6
FizzBuzz
```

```
Insert a number: 9
Buzz
```

```
Insert a number: 20
Fizz
```

Exercise Solution 1

```
x = int(input("Insert a number:"))

if (x % 2 == 0) and (x % 3 == 0):
    print("FizzBuzz")
elif x % 2 == 0:
    print("Fizz")
elif x % 3 == 0:
    print("Buzz")
```

Exercise Solution 2

```
x = int(input("Insert a number:"))

is_div_2 = (x % 2 == 0)
is_div_3 = (x % 3 == 0)

if is_div_2 and is_div_3:
    print("FizzBuzz")
elif is_div_2:
    print("Fizz")
elif is_div_3:
    print("Buzz")
```

Exercise Solution 3

```
x = int(input("Insert a number:"))

response = ""

if x % 2 == 0:
    response += "Fizz"
if x % 3 == 0:
    response += "Buzz"

print(response)
```

While-Loops

Allows you to repeat instructions

With an **if-statement**:

```
x = int(input("Insert num < 5: "))

if x >= 5:
   print("ERROR! Wrong number")
   x = int(input("Insert num < 5: "))

print("CORRECT!")
```

With a **while-loop**:

```
x = int(input("Insert num < 5: "))

while x >= 5:
   print("ERROR! Wrong number")
   x = int(input("Insert num < 5: "))

print("CORRECT!")
```

While-Loops

Anatomy of a while-loop:

1. Uses the **`while`** keyword
2. Ends with a colon (**`:`**)
3. Uses **tabs** for spacing from the outside scope

```
x = 5
while x < 10:
    print(x)
    x += 1
```

Exercises

Complete the following exercises:

- **Ex 3.3**: Complete the following program that prints out all the numbers from 1 to 15
- **Ex 3.4**: Complete the following program that prints out the first 15 multiples of 6

Exercise 3.3 - Solution

```
x = 1

while x <= 15:
    print(x)
    x += 1
```

Exercise 3.4 - Solution

```
x = 6

while x <= 6 * 15:
    print(x)
    x += 6
```

```
x = 1

while x <= 15:
    print(x * 6)
    x += 1
```

For-Loops

Repeat a <u>specific</u> amount of times

With a **while-loop**:

```
x = 0

while x < 5:
    print(x)
    x += 1
```

With a **for-loop**:

```
for x in range(5):
    print(x)
```

For-Loops

Anatomy of a for-loop:

1. Uses the **for** keyword
2. Ends with a colon (**:**)
3. Takes up to 3 parameters:
   a. Start number (optional, default is 0)
   b. End number
   c. Step-size (optional, default is 1)

```
for x in range(10):
    print(x)
```

```
for x in range(2, 10):
    print(x)
```

```
for x in range(2, 10, 3):
    print(x)
```

Exercises

Complete the following exercises:

- **Ex 3.5**: A program to print all numbers from 0 to 15
- **Ex 3.6**: A program to print all numbers from 5 to 20
- **Ex 3.7**: A program to print all even numbers from 10 to 30
- **Ex 3.8**: A program that prints out the first 15 multiples of 6

Exercise 3.5 - Solution

```
for x in range(16):
    print(x)
```

Exercise 3.6 - Solution

```
for x in range(5, 21):
    print(x)
```

Exercise 3.7 - Solution

```
for x in range(10, 32, 2):
    print(x)
```

Exercise 3.8 - Solution

```
for x in range(1, 16):
    print(x * 6)
```

Remember: String Operations

Remember the following string operation shorthands:

Repetition:

```
x = "*"
print(x * 5)
```

Concatenation:

```
x = "***"
y = "..."
print(x + y + x)
```

Exercise

Complete the **3.9** program.

- It takes an input **num** from the user.
- It should print out a full square pattern of size **num**.

Example if **num=5**:

```
*****
*****
*****
*****
*****
```

Exercise 3.9 - Solution

```
num = int(input("Insert the size of the square: "))

for x in range(num):
    print("*" * num)
```

Exercise

Complete the **3.10** program.

- It takes an input **num** from the user.
- It should print out a triangular pattern of size **num**.

Example if **num=5**:

```
....*
...**
..***
.****
*****
```

Exercise 3.10 - Solution

```
num = int(input("Insert the size of the triangle: "))

for x in range(1, num + 1):
    n_stars = x
    n_dots = num - n_stars
    stars = n_stars * "*"
    dots = n_dots * "."
    print(dots + stars)
```

End of Class

See you all next week!