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

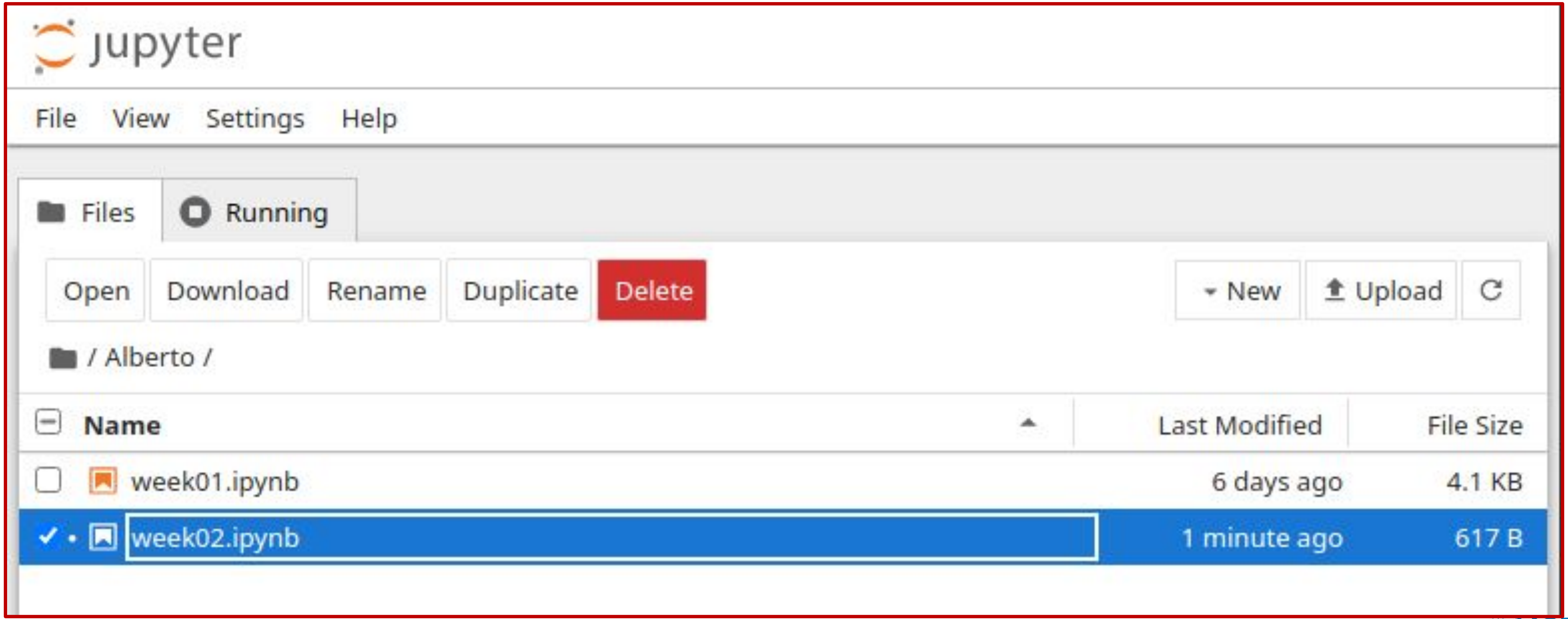
Jupyter Notebook Setup

In a browser:

```
192.168.10.4:8888
```

Password: **ist**

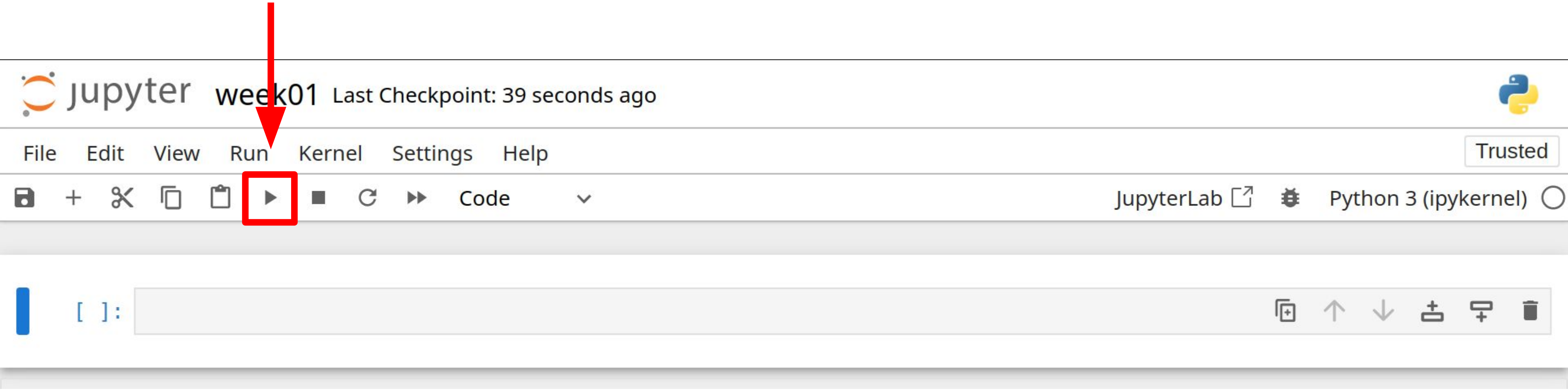
Jupyter Notebook Setup



Name	Last Modified	File Size
week01.ipynb	6 days ago	4.1 KB
week02.ipynb	1 minute ago	617 B

Jupyter Notebook Structure

Run Cell



Recap: Variables

A variable is a named container that stores data or values.

```
x = 42
y = "Hello"
```

Variable declarations must contain a variable name followed by an equals sign (**=**).

```
variable = "I am a variable"
also_valid_variable_name = "I am also a variable"
```

Recap: Output

The **print** function can be used to display variables and values

```
print("Hello World!")
print(123)
```

```
x = 42
print(x)
```

```
y = "Hello"
print(y)
```

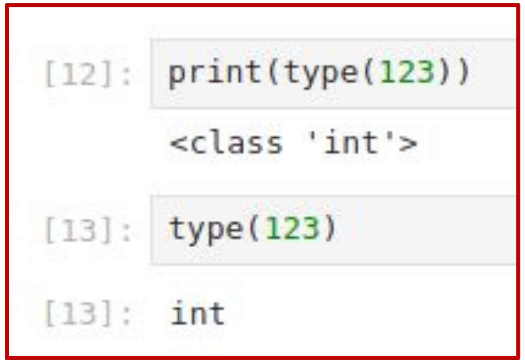
Recap: Data Types

Python has 4 primitive data types:

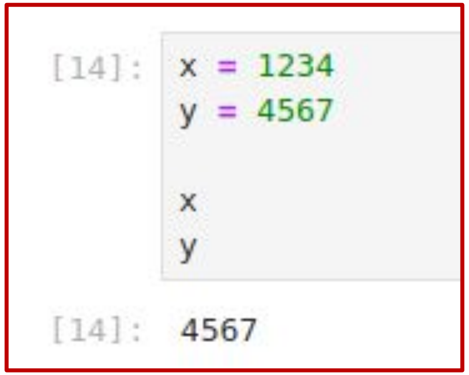
int	`42`	`1200`	`1_200`	`-3`
float	`3.14`	`0.00001`	`-2.1`	
str	`"Hello"`	`"A"`	`"I am a full sentence!"`	
bool	`True`	`False`		

Recap: Notebook TIP!

Jupyter Notebooks will automatically print the return value of the final line in a Notebook cell.



```
[12]: print(type(123))

      <class 'int'>

[13]: type(123)

[13]: int
```



```
[14]: x = 1234
      y = 4567


      x
      y

[14]: 4567
```

Recap: Changing the value of variables

You can mutate the value you assign to a variable

```
x = 42
print(x)

x = 200
print(x)

x = "Hello"
print(x)
```

Recap: Arithmetic Operations

You can perform arithmetic with variables

```
x = 9
y = 3
print(x + y)
print(x - y)
print(x * y)
print(x / y)
```

What is the output type of the division operation?

Recap: Arithmetic TIP!

Incrementing variables can be done with shorthands:

```
x = 100


x = x + 10
print(x)
x = x * 5
print(x)
x = x - 40
print(x)
```

```
x = 100


x += 10
print(x)
x *= 5
print(x)
x -= 40
print(x)
```

Left and Right statements are **identical**

Recap: Type Casting

You can convert from one type to another

```
x = "123"
y = int(x)
print(y)
print(type(y))
```

```
x = "23.88"
y = float(x)
print(y)
print(type(y))
```

Input

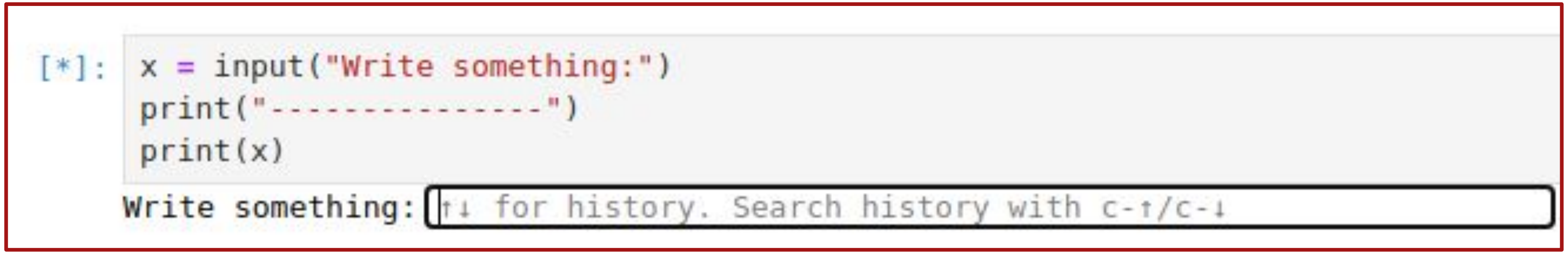
The **input** function can be used to take data from the user

```
x = input()
print(x)
```

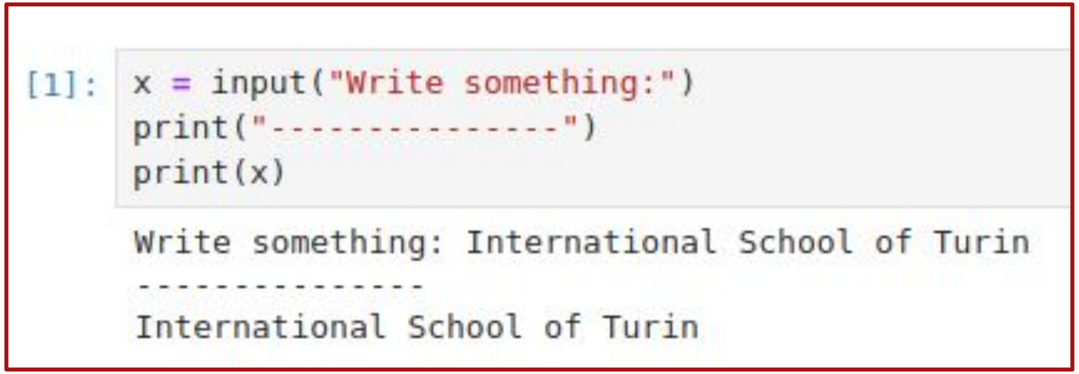
```
x = input("Write something:")
print(x)
```

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
```

Input

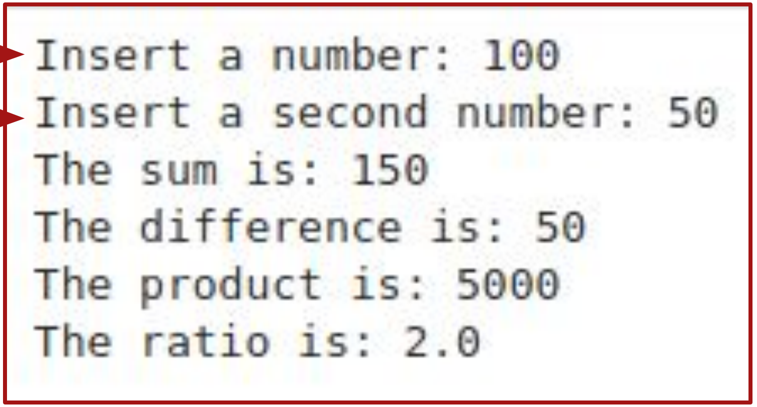
The **input** function can be used to take data from the user

TIP: Combine input with type casting!

```
x = int(input("Write a number:"))
print(x + 100)
```

Exercise

Write a program that asks the user for 2 numbers and prints out their sum, difference, product and ratio.



```
Insert a number: 100
Insert a second number: 50
The sum is: 150
The difference is: 50
The product is: 5000
The ratio is: 2.0
```

TIP: Use **`input`** to ask the user for numbers!

Exercise Solution

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

print("The sum is: " + str(x + y))
print("The difference is: " + str(x - y))
print("The product is: " + str(x * y))
print("The ratio is: " + str(x / y))
```

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
```

Chaining Comparisons

- **not** (negation)

```
not True
```

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

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

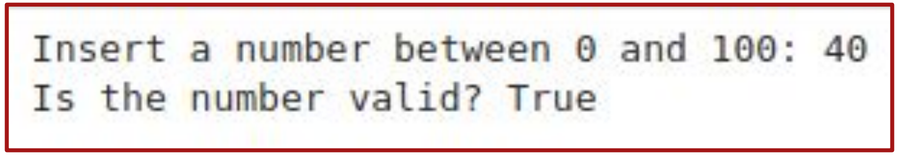
```
(5 < 6) and (5 < 10)
```

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

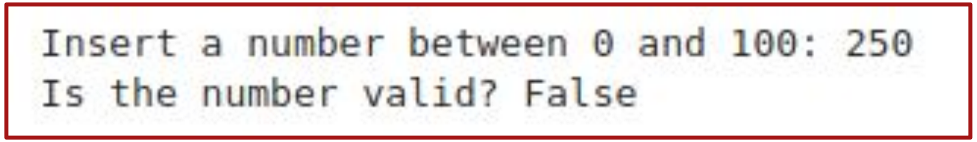
```
(5 < 3) or (5 < 10)
```

Exercise

Write a program that asks the user for a number and checks that the number is between 0 and 100.



```
Insert a number between 0 and 100: 40
Is the number valid? True
```



```
Insert a number between 0 and 100: 250
Is the number valid? False
```

TIP: Use **`input`** to ask the user for numbers!

Exercise Solution

```
x = int(input("Insert a number between 0 and 100:"))
is_valid = (x >= 0) and (x <= 100)
print("Is the number valid? " + str(is_valid))
```

```
x = int(input("Insert a number between 0 and 100:"))
print("Is the number valid? " + str((x >= 0) and (x <= 100)))
```

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 **do not** need an else block, it's optional.

If-Statements

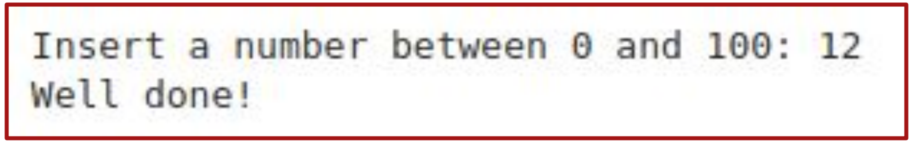
Anatomy of an if-statement:

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

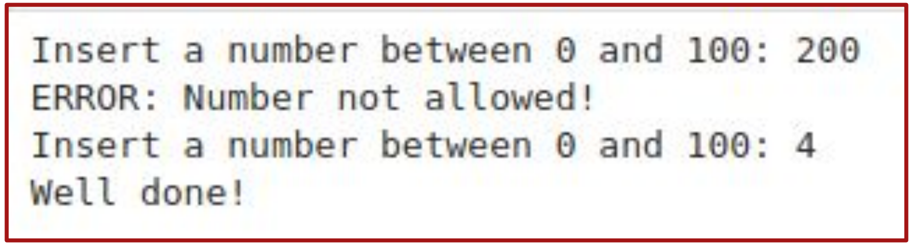
```
x = 5

if x < 10:
    print("Hello")
    print("World")
```

Exercise

Write a program that asks the user for a number between 0 and 100, if it is not valid it asks the user to input it again.



```
Insert a number between 0 and 100: 12
Well done!
```



```
Insert a number between 0 and 100: 200
ERROR: Number not allowed!
Insert a number between 0 and 100: 4
Well done!
```

Exercise Solution

```
x = int(input("Insert a number between 0 and 100:"))

if (x < 0) or (x > 100):
    print("ERROR: Number not allowed!")
    x = int(input("Insert a number between 0 and 100:"))

print("Well done!")
```

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")
```

End of Class

See you all next week!