

DIT374 Python for Data Scientists

Period I, 2020

Organization

Staff

Dag Wedelin (Examiner): dag@chalmers.se

Marwa Naili (Course responsible): naili@chalmers.se

Shirin Tavara (Lecturer): tavara@chalmers.se

David Bosh (TA): davidbos@chalmers.se

Schedule

Schedule:

- Lectures: Monday and Wednesday at 10-12

See the TimeEdit:

https://cloud.timeedit.net/chalmers_test/web/public/ri10Q603570007QQ51ZZ156004yYW46976406008Q655.html

Schedule

Lab sessions: Wednesday and Friday at 13-15

- Supervised work on assignments

Online teaching: all lectures, labs and office hours will be online via Zoom

Schedule

Lab sessions:

- Slack DIT374: https://join.slack.com/t/dit374/shared_invite/zt-gl0v30d0-Gcw8lwwQJmN42ZJW2KiZew
- Use this link for the queue: <http://www.waglys.com/sipfdc>

Course Website

- The official course webpage is the Canvas page :
 - <https://canvas.gu.se/courses/36872>
- Slides, assignments and data will be posted after the lectures via the canvas site

Assignments

- Assignments will be posted on Canvas on Monday
- Deadlines: 1 week
- **Do not submit an incomplete Assignment!** We are available to help you, and you can receive a short extension if you contact us.

Grading

- Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)
- A passing grade for the entire course requires at least a passing grade for all assignments, both the assignments that are graded as pass/fail and those that are graded as VG/G/U.

Literature

- [1] C. Horstmann: Python for everyone 3rd ed., ISBN: 978-1-119-63829-2:
<https://www.chalmersstore.se/utlandsk-litteratur/python-for-everyone-1.html>
- [2] Python tutorial: <https://docs.python.org/3/tutorial/>
- [3] Python 3 course: http://www.python-course.eu/python3_course.php
- [4] w3schools, Python: <https://www.w3schools.com/python/default.asp>
- [5] NumPy & SciPy references: <http://docs.scipy.org/doc/>
- [6] W. McKinney: Python for Data Analysis, 2nd Edition. ISBN: 9781491957660:
https://www.amazon.com/gp/product/1491957662/ref=as_li_qf_asin_il_tl?ie=UTF8&tag=amazonaffi048-20&creative=9325&linkCode=as2&creativeASIN=1491957662&linkId=ca87c0dc52af4fefb49377651641428d

Student representatives

- Need 2-3 Volunteers
- If you're interested in being a student representative, please send me an email!

Course content

Polls

Do you have any programming knowledge?

A	B	C
Yes	Yes but not much	No

Polls

Did you use Python as a programming language before?

A	B	C
Basic	Advance	no

Polls

Do you have any knowledge about Data science?

A	B
Yes	No

Topics

Three main topics:

1. Programing with Python

- Basics of python
- Object oriented programming

2. Data structure and algorithms

- Orientation about algorithms and algorithm design principles
- Data structure

3. Data science

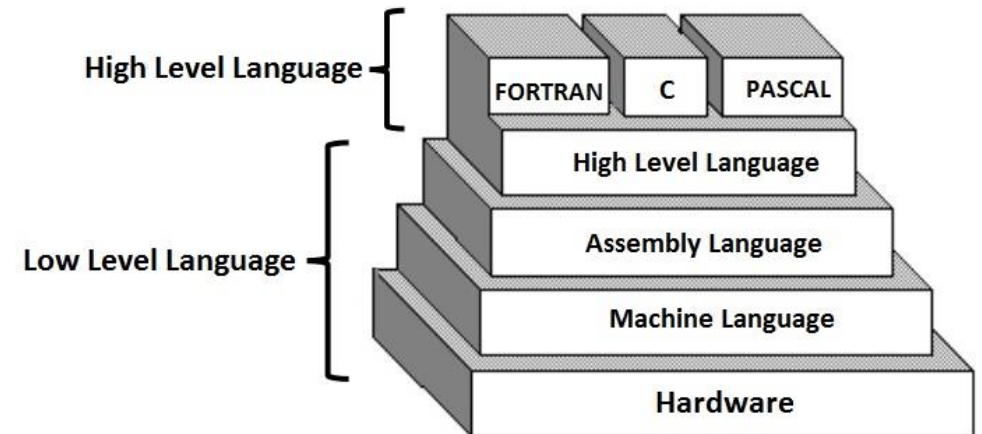
- Python for Data Science

Learning Goals

- Make efficient use of predefined data structures in Python
- Construct simple programs using classes and objects
- Analyze the efficiency of different algorithms, for example searching and sorting
- Use a standard library of data structures and algorithms in Python for solving tasks within the area of data science.

Programming language

- Programming language : a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks.
- Computer programs can be written in high and low level languages depending on the task and the hardware being used.



Programming language: Low level language

- Used to write programs that relate to the specific architecture and hardware of a particular type of computer.
- Closer to the native language of a computer, making them harder for programmers to understand.

Programming language: Low level language

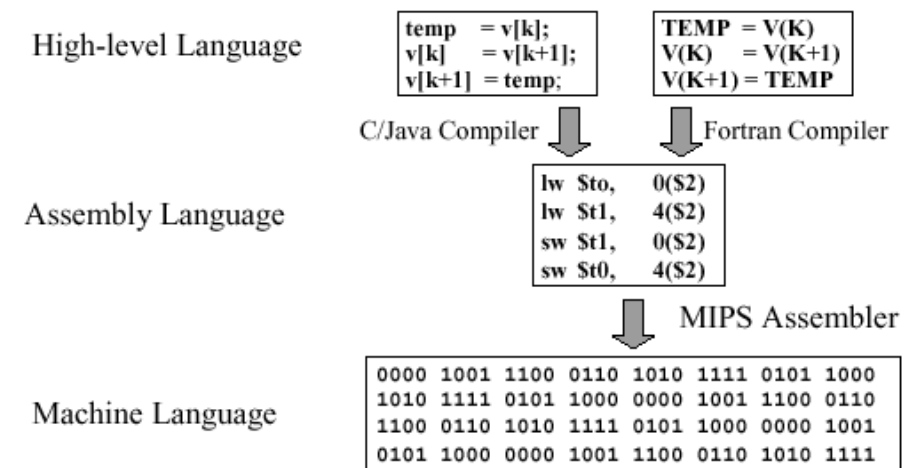
Machine language:

- Fundamental language of the computer's processor
- All programs are converted into machine language before they can be executed.
- Consists of combination of 0's and 1's

Programming language: Low level language

Assembly language:

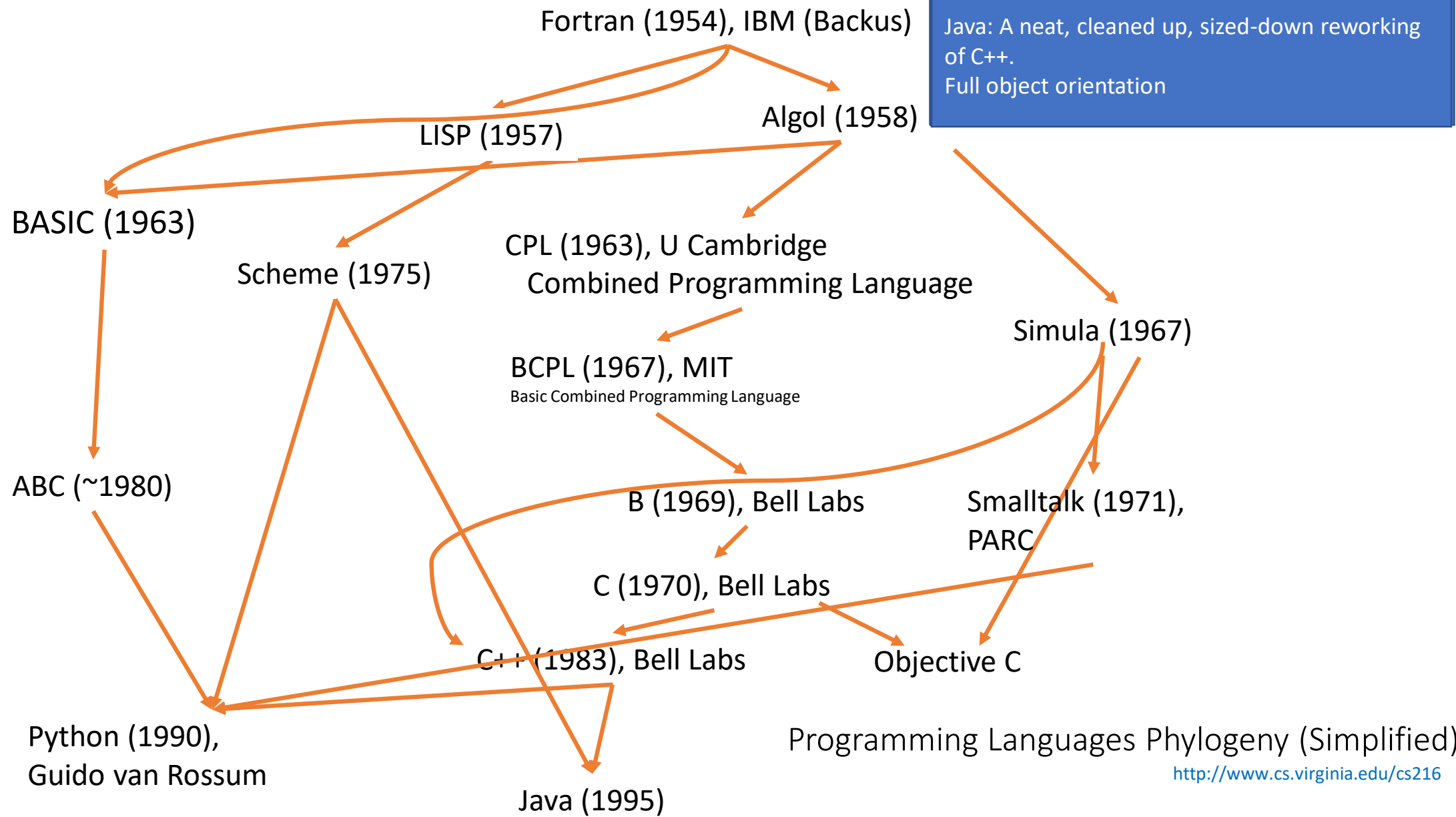
- Uses symbolic operation code to represent the machine operation code.
- Still used for developing code for specialist hardware, such as device drivers



Programming language: High level language

- Written in a form that is close to our human language, enabling the programmer to just focus on the problem being solved.
- No particular knowledge of the hardware is needed since it creates programs that are portable and not tied to a particular computer or microchip.

History of High level programming languages



Python

Guido van Rossum: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems."

Python

- Python is the most widely used data science programming language.
- It supports multiple paradigms, from functional to structured and procedural programming.

Python (3 versions)

- Version 1 (January 1994):
 - Included functional programming tools (lambda, map, filter and reduce)
 - Support complex numbers
- Version 2 (October 2000):
 - Introduced list comprehensions and generators
 - Unification with Python's types (written in C) and classes (written in python) into one hierarchy
- Version 3 (December 2008):
 - Still follow object oriented, structured, and functional programming paradigms but within such broad choices (the details were intended to be more obvious in Python 3.0 than they were in Python 2.x).

Python is slow!!

Python is Dynamically Typed rather than Statically Typed:

- At the time the program executes, the interpreter doesn't know the type of the variables that are defined.

/ C code */*

```
int a = 1;  
int b = 2;  
int c = a + b;
```



C Addition

1. Assign <int> 1 to a
2. Assign <int> 2 to b
3. call binary_add<int, int>(a, b)
4. Assign the result to c

python code

```
a = 1  
b = 2  
c = a + b
```



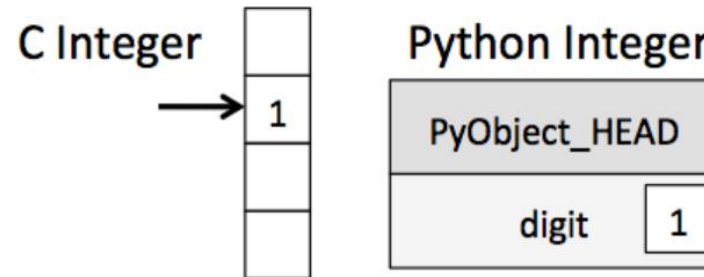
Python Addition

1. Assign 1 to a
 - 1a. Set a->PyObject_HEAD->typecode to integer
 - 1b. Set a->val = 1
2. Assign 2 to b (same as 1)
3. call binary_add(a, b)
 - 3a. find typecode in a->PyObject_HEAD
 - 3b. a is an integer; value is a->val
 - 3c. find typecode in b->PyObject_HEAD
 - 3d. b is an integer; value is b->val
 - 3e. call binary_add<int, int>(a->val, b->val)
 - 3f. result of this is result, and is an integer.
4. Create a Python object c
 - 4a. set c->PyObject_HEAD->typecode to integer
 - 4b. set c->val to result

Python is slow!!

Python is Dynamically Typed rather than Statically Typed:














- Python's object model can lead to inefficient memory access



<http://jakevdp.github.io/blog/2014/05/09/why-python-is-slow/>

Programming language

Language Ranking: **Trending**

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	94.9
3	C	  	91.4
4	C++	  	87.5
5	JavaScript		74.9





Why Python?

- Extensive selection of libraries
- Code simplicity
- High flexibility
- Platform independence
- Constant support from the developer community
- Lot of documentation

Why Python for data science

Dealing with complex problems and involves four major steps - data collection & cleaning, data exploration, data modeling and data visualization.



	 Facebook	 Instagram
Quora	NETFLIX	 Dropbox

<https://www.heliossolutions.co/blog/why-choose-python-for-artificial-intelligence-and-machine-learning/>

Python interpreter

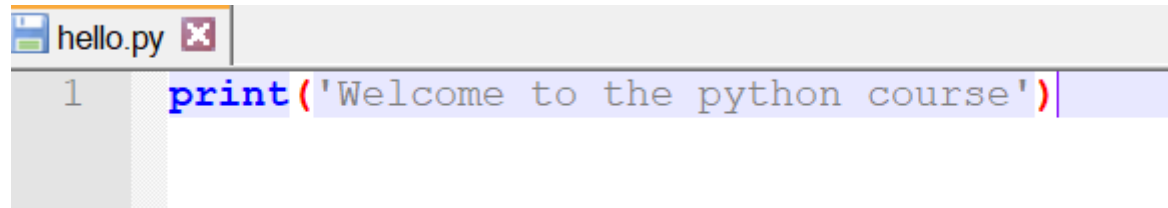
- Python is an interpreted language.
- The python interpreter runs a program by executing one statement at a time.
- The standard interactive python interpreter can be invoked on the command line:

```
C:\Users\naili>python
Python 3.7.5rc1 (tags/v3.7.5rc1:4082f600a5, Oct  1 2019, 20:28:14) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> n = 10
>>> print(n)
10
>>> type(n)
<class 'int'>
>>>
```

Python interpreter

Running python programs can be done by calling python with a .py file as a first argument

```
C:\Users\naili\Desktop>python hello.py  
Welcome to the python course
```

A screenshot of a code editor window titled 'hello.py'. The editor shows a single line of Python code: `print('Welcome to the python course')`. The line is numbered '1' on the left. The code is syntax-highlighted, with 'print' in blue, the opening parenthesis in red, the string in quotes in black, and the closing parenthesis in red. A vertical cursor is positioned at the end of the line.

```
1 print('Welcome to the python course')
```


Software

You have multiple options when installing Python (Make sure to get version 3.7 or later, <https://www.python.org/downloads/>)

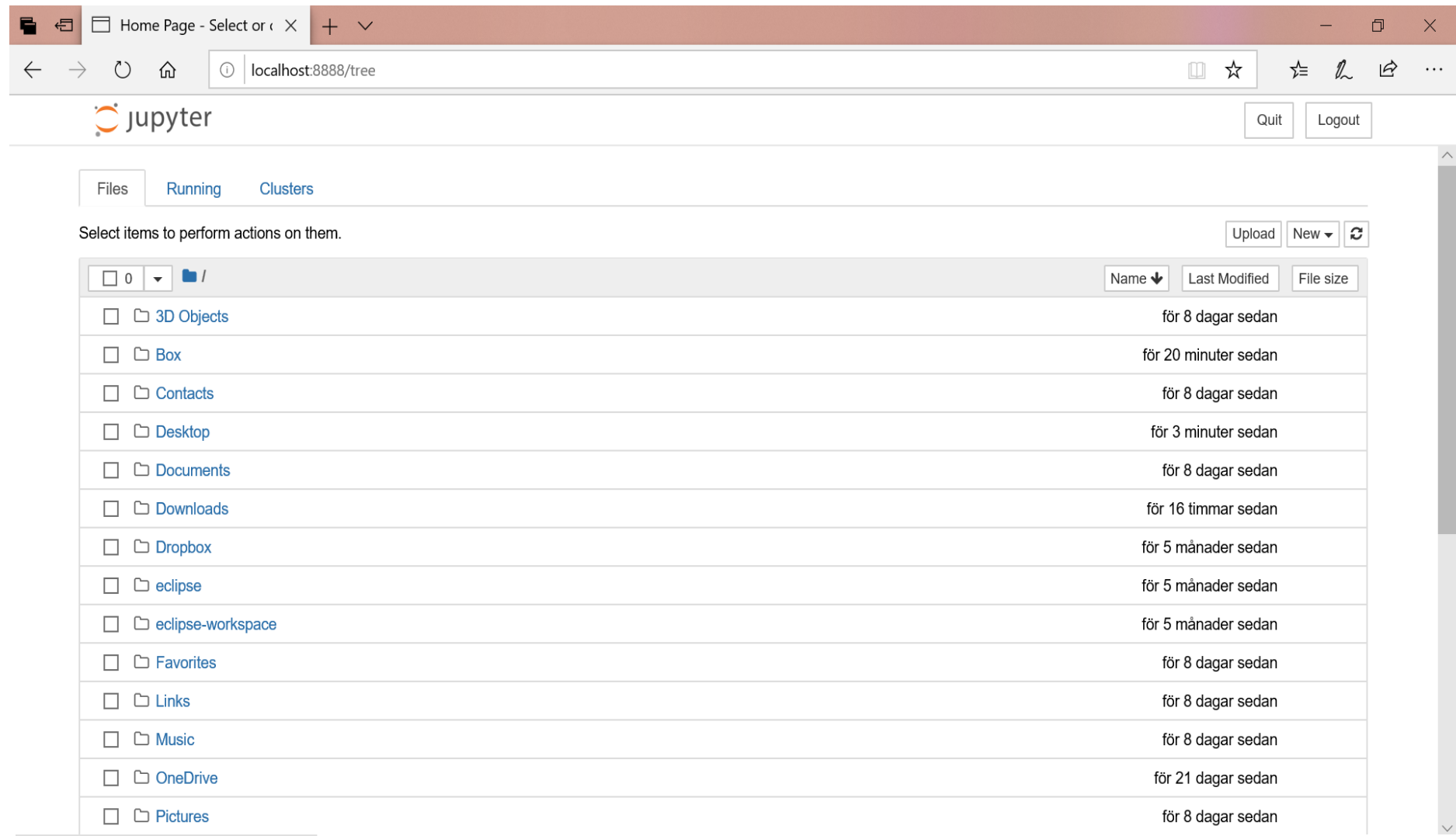
- Anaconda (<https://www.anaconda.com/products/individual#linux>)
Python distribution, which includes several useful libraries. (Installation: <https://docs.continuum.io/anaconda/install/>)
- The editor PyCharm
(<https://www.jetbrains.com/pycharm/download/#section=windows>)
- miniconda (<https://docs.conda.io/en/latest/miniconda.html>)
- Jupyter Notebooks (included in Anaconda)
- Spyder (included in Anaconda)
- ...

Software

- All software are free and can be downloaded from the web.
- Here are some instructions
at <https://it.portal.chalmers.se/itportal/GenStud/Python>

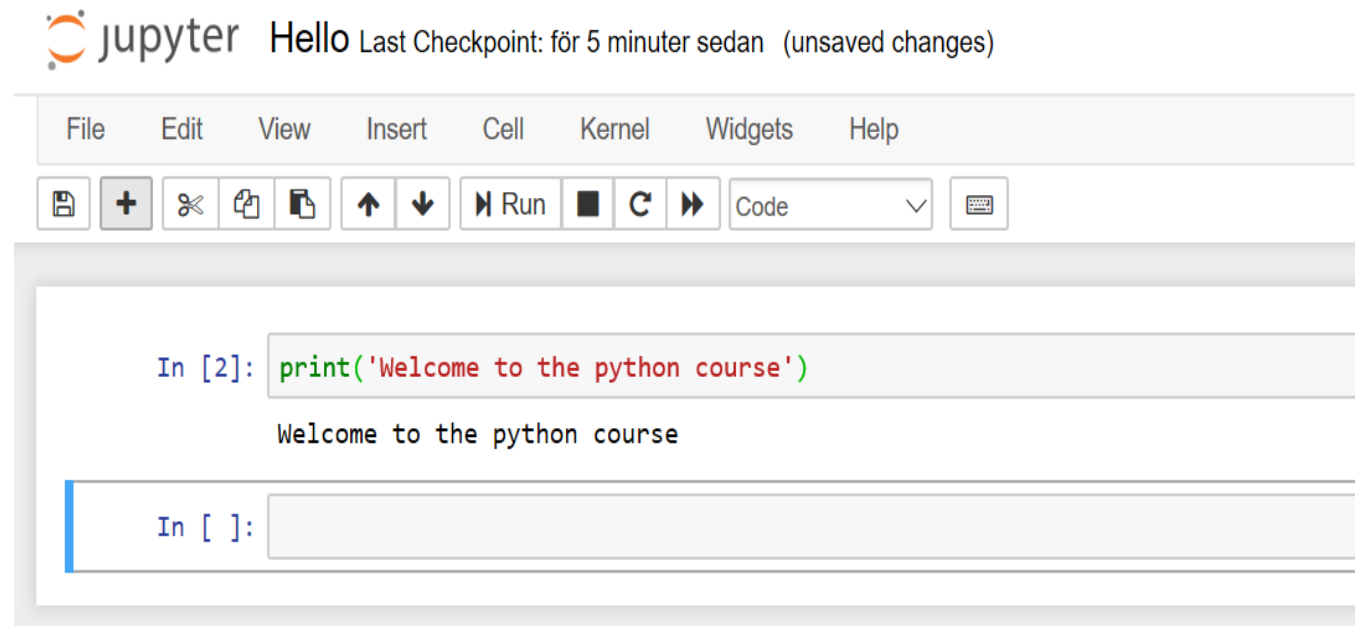
Your first Python program

- 1- Open Navigator
- 2- Run python In a Jupyter Notebook
- 3- Create a new Notebook with the Python version you installed



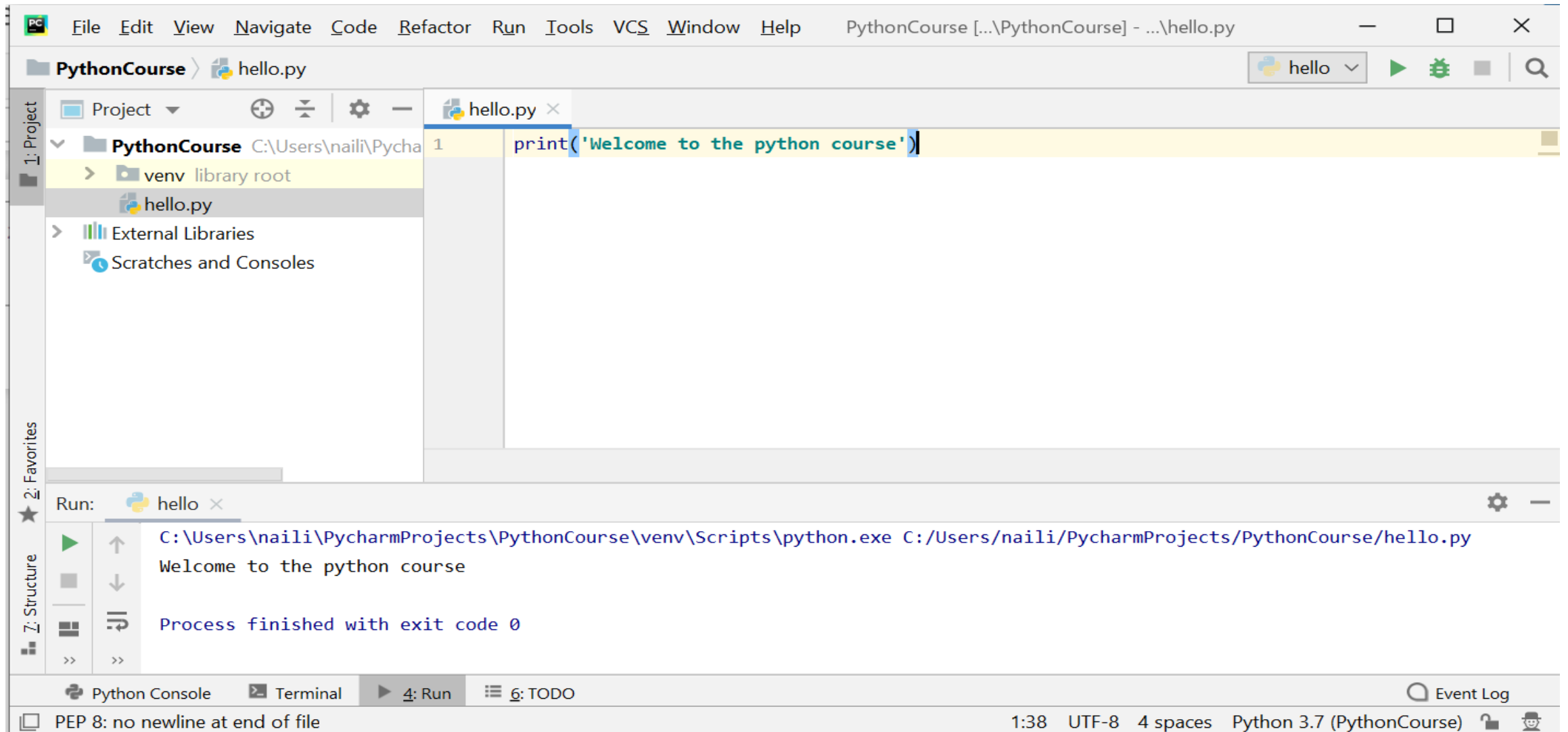
Your first Python program

- In the first line of the Notebook, type :
`print('Welcome to the python course')`
- Save your Notebook by clicking the save and checkpoint icon (or select File and Save and Checkpoint in the top menu).
- Run your new program by clicking the Run button (or selecting Cell - Run All from the top menu).



Documentation: <https://jupyter-notebook.readthedocs.io/en/stable/>

Your first Python program



Input/Output: Print

Print function allows to write into the standard output

```
print "Python for Data Scientist"
```

SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Python for Data Scientist")?

```
print ("Python for Data Scientist")
```

Python for Data Scientist

The use of print is different according to the version of the python program: 2.x and 3.x

Output: Print

The arguments of the print function are the following ones:

`print(value1, ..., sep=' ', end='\n', file=sys.stdout, flush=False)`

```
x = 3.14
print('x = ', x)

print ('x = \n', x)

y = 4
print(x, y, sep=';')
```

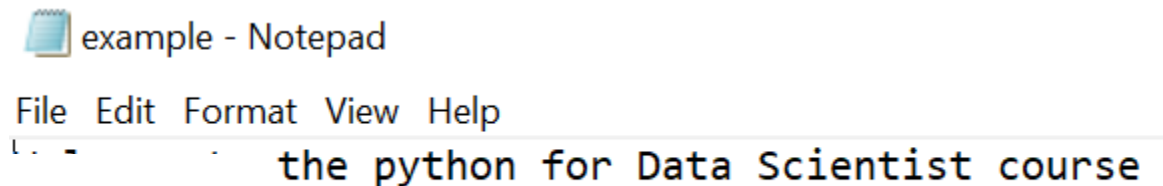
```
x = 3.14
x =
3.14
3.14;4
```

Print

- The output of the print function is send to the standard output stream (sys.stdout) by default.
- By redefining the keyword parameter "file" we can send the output into a different stream e.g. sys.stderr or a file:

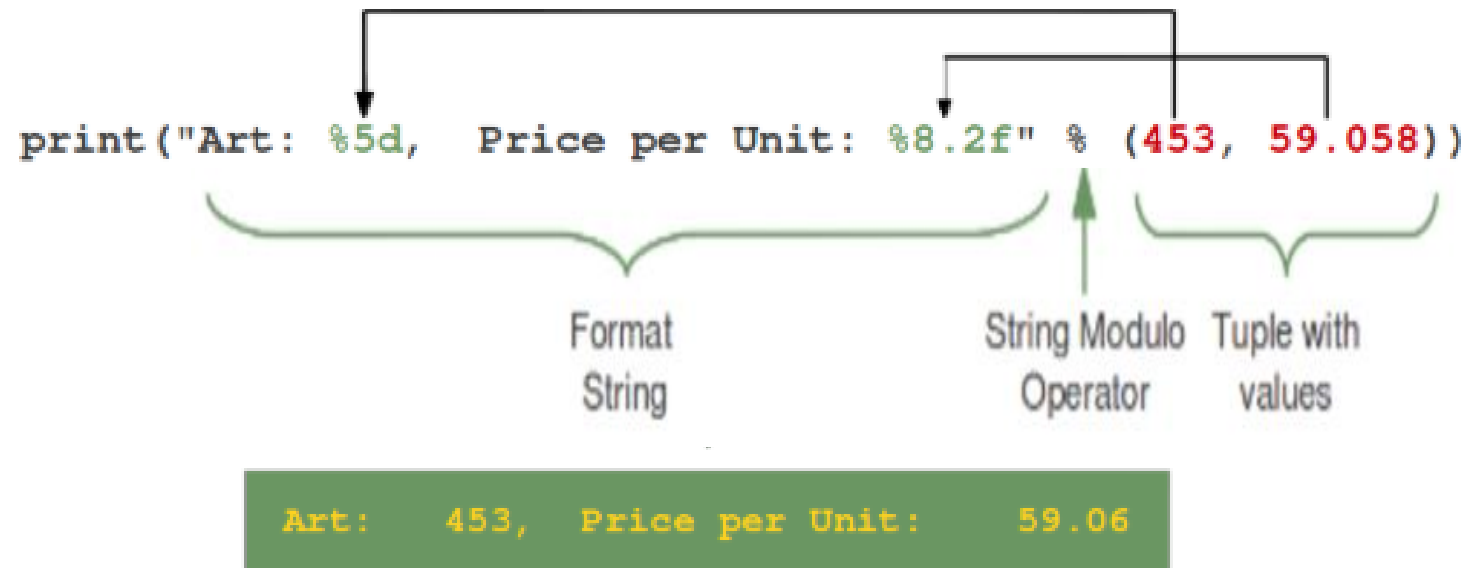
```
f = open('example.txt', 'w')
print("Welcome to the python for Data Scientist course", file=f)
f.close()
```

```
import sys
# output into sys.stderr:
...
print("Error: 42", file=sys.stderr)
```



Error: 42

Formatted Output: printf

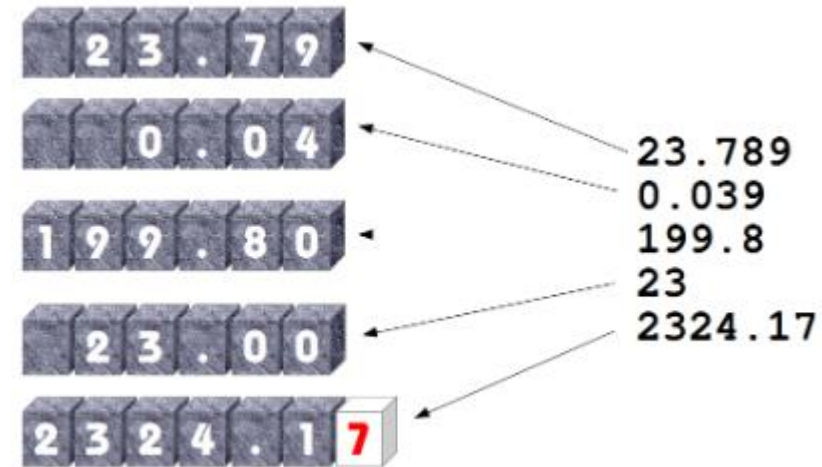
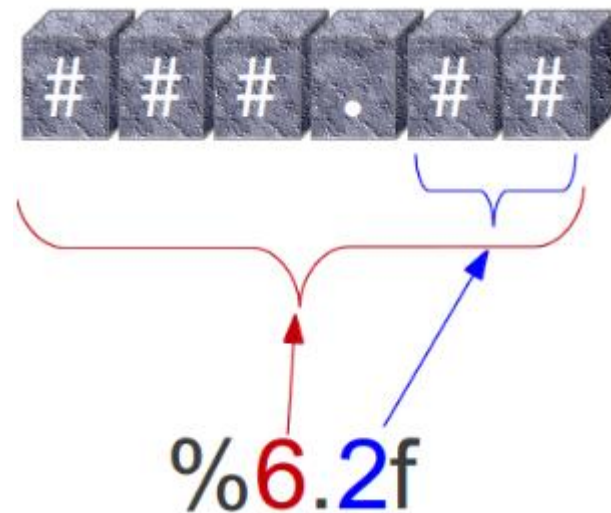


Formatted Output: printf

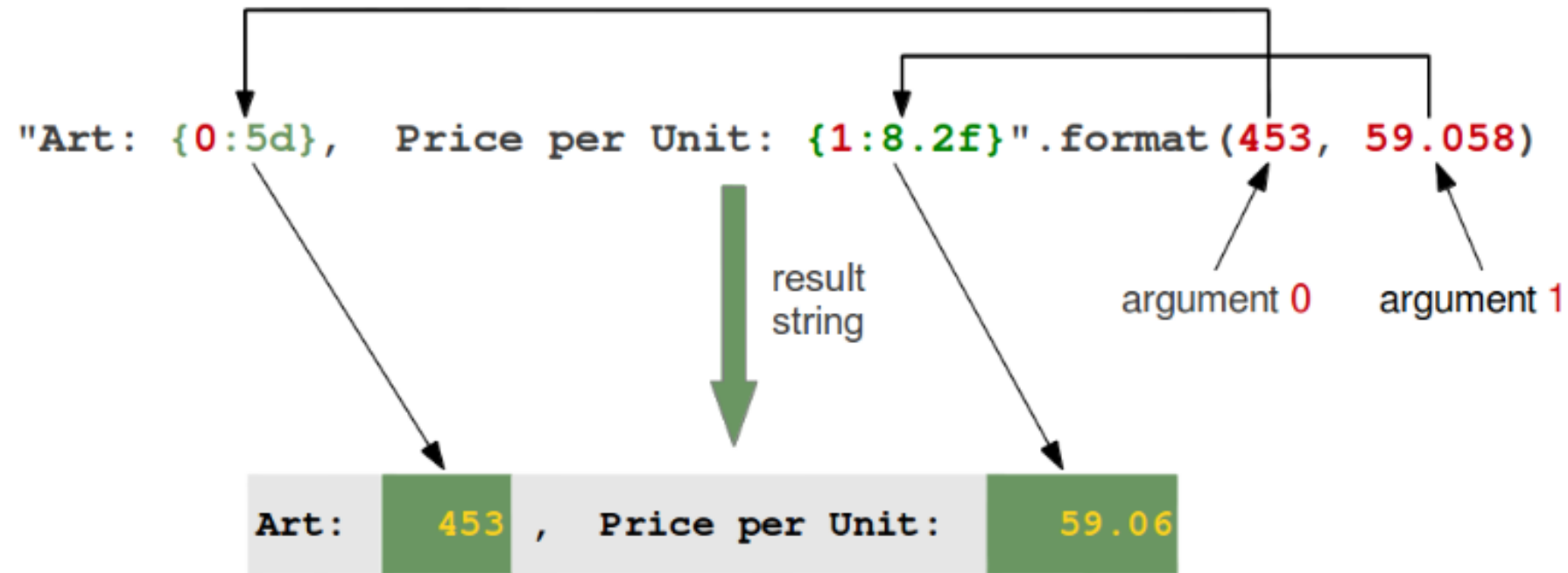
%8.2f?

→ [flags][width][.precision]type

Example:



Formatted Output: The string method 'format'



```
print('price_x:{0:5d}, price_y:{1:8.3f}'.format(345,33.9897))
```

```
price_x: 345, price_y: 33.990
```

Input: input()

```
input([prompt])
```

```
number = input("Enter number ")  
name = input("Enter name ")  
  
print("\n")  
print("Printing type of an input value")  
print("type of number", type(number))  
print("type of name", type(name))
```

Enter number 4
Enter name Alex

Printing type of a input value
type of number <class 'str'>
type of name <class 'str'>

whatever you enter as input, the
input() function always converts it
into a string.

Input: input()

```
first_number = int(input("Enter first number "))  
second_number = int(input("Enter second number "))
```

Enter first number 5
Enter second number 8

```
print("\n")  
print("First Number:", first_number)  
print("Second Number:", second_number)  
sum1 = first_number + second_number  
print("Addition of two number is: ", sum1)
```

First Number: 5
Second Number: 8
Addition of two number is: 13

Input/output: from a file

```
f = open("file.txt", "r")
```

#read: returns the whole text, but you can also specify how many characters you want to return

```
print(f.read())
```

#readline: read one line of the file:

```
print(f.readline())
```

```
print(f.readline())
```

#It is a good practice to always close the file when you are done with it.

```
f.close()
```



file - Notepad

File Edit Format View Help

```
Hello! Welcome to the python course.  
This file is for testing purposes.
```

Input/output: from a file

```
f = open("file2.txt", "a")  
f.write("Hello!")  
f.close()
```

#open and read the file after the appending:

```
f = open("file2.txt", "r")  
print(f.read())
```

```
f = open("file2.txt", "w")  
f.write("Hello again!")  
f.close()
```

#open and read the file after writing:

```
f = open("file2.txt", "r")  
print(f.read())
```

To create a new file in Python, use the `open()` method, with one of the following parameters:

- "x" - Create - will create a file, returns an error if the file exist
- "a" - Append - will create a file if the specified file does not exist
- "w" - Write - will create a file if the specified file does not exist

Basics of python

Basics of Python:

- Data types and structure
- Branching and iteration
- Functions, decomposition and abstraction
- Object oriented programming