University of Basel

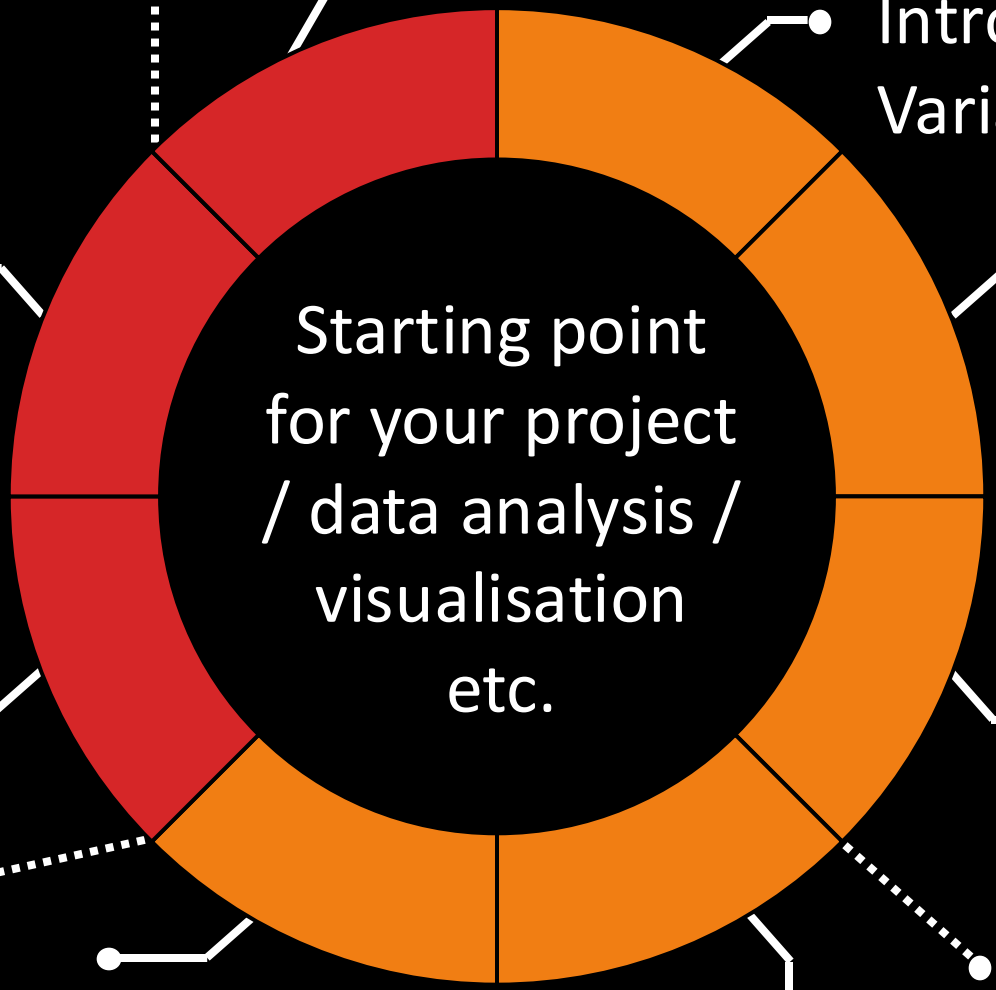
SDSC

Python Crash Course

Dr. Maxim Samarin
Senior Data Scientist @ Swiss Data Science Center

30th September / 1st / 7th / 8th October 2024

Overview



Next Tuesday →

Advanced Examples / Exercises / Q&A

Introduction,
Variables & Arithmetic

Data Structures

Conditional
Statements

Today ↑

Iterations

Plotting

Tomorrow →

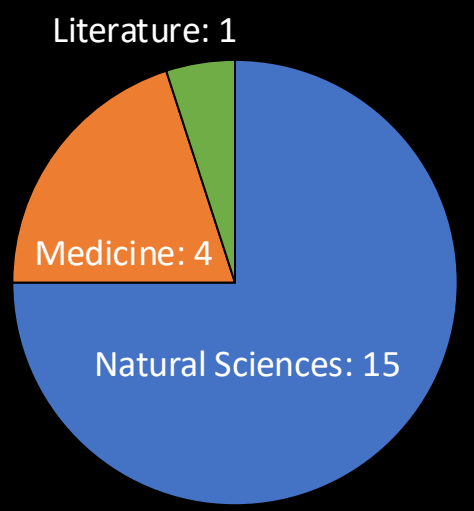
↑ Next Monday

NumPy & Pandas

Subroutines &
Object-oriented
Programming

Your Experience and Goal

- What is your programming experience so far and an intermediate goal for you?



Pierre ⇢ Ramón ⇢ Eduardo ⇢ Ángel ⇢ Sucheta ⇢ Galya ⇢
⇢ Dominik ⇢ Thabo ⇢ Mirela ⇢ Giorgi ⇢ Yasemin ⇢
⇢ Cassandra ⇢ Babatunde ⇢ Erika ⇢ Rafael ⇢ Nora ⇢
⇢ Micheline ⇢ Constantin ⇢ Subashree ⇢ Jennyfer

My Programming Career

- School classes programming in Java
- Studying Physics: C++, MatLab, Python
- Project: Analysis & Graphical User Interface in C, Python
- Physics modelling in Fortran, Python
- M.Sc. thesis: Data Mining & simulations in R, Python
- Ph.D. thesis: Machine / Deep Learning research in Python ...
- Data Scientist and ML / AI Researcher : Python …

Python vs. other Languages

- Python is a dynamically typed programming (script) language
- Code is interpreted
- A lot of details *under the hood*
- Zen of Python: Simple, explicit, sparse, readable, practical

→ Fast prototyping, quick to learn

- Statically typed: fix type and object
- Code is compiled
- A lot of explicit control
- Efficient implementations, elaborate syntax

→ Fast and powerful codes

Python & C++ Examples


```python
name = input("Please enter your name: ")
print("Good morning,", name)
```

```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {
    string name;

    cout << "Please enter your name: ";
    cin >> name;
    cout << "Good morning, " << name << endl;

    return 0;
}
```

- Both have the same output:

  Please enter your name: Monty
  Good morning, Monty

Start your Working Environment

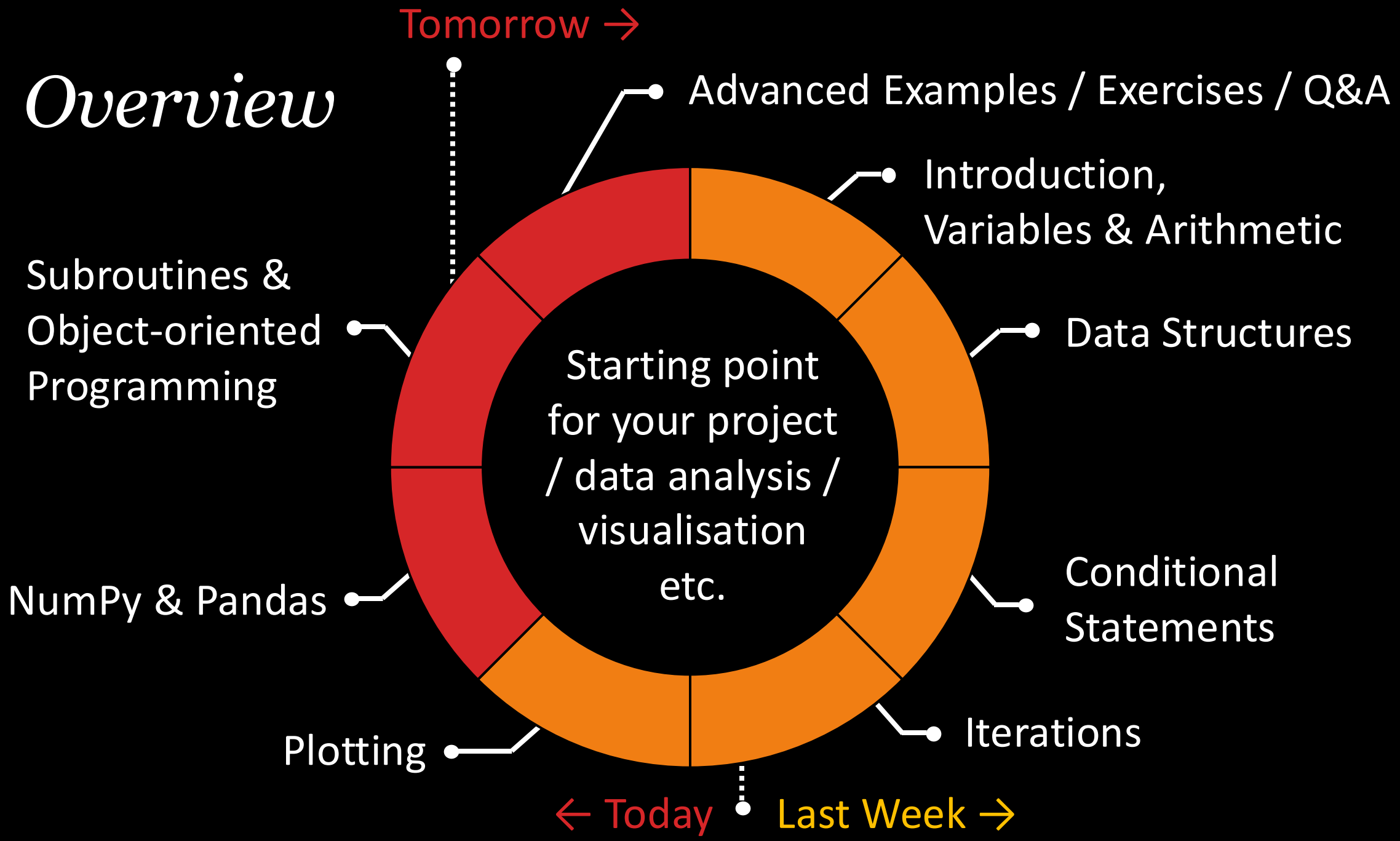
1. Access the course environment via Noto link provided in E-Mail and sign in with your SWITCH edu-ID ("*Use your Switch AAI login*")
2. Or: Download new material, start Jupyter Lab and open notebooks

- Suggestions on how to work in this course:

: Follow presentation, while executing scripts yourself, making adjustments and notes in your own notebook

: Follow presentation, switch to programming environment for exercises

Overview


Tomorrow →
Advanced Examples / Exercises / Q&A
Introduction,
Variables & Arithmetic
Data Structures
Conditional
Statements
Iterations
Last Week →
← Today
Plotting
NumPy & Pandas
Subroutines &
Object-oriented
Programming
Starting point
for your project
/ data analysis /
visualisation
etc.

Different Ways to Execute Scripts

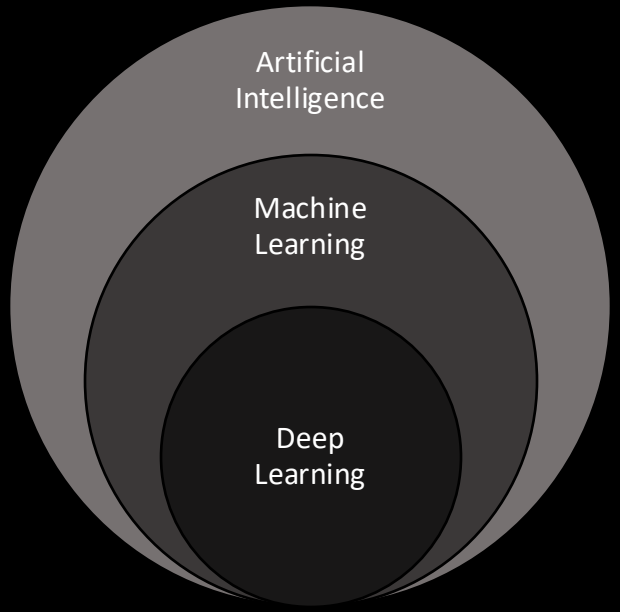
- Jupyter Notebooks (as in this class): Document-style, combining executable scripts with rich documentation and formatting; LaTeX can be used
- Terminal: Execute scripts via the terminal
  1. Write your script in an editor and save it to a *my_script.py* file
  2. Execute the script in the terminal with *python my_script.py*
- Spyder: Integrated development environment (IDE) combining an editor with a terminal and other useful functionalities
- Or any other editor / IDE such as Atom, Sublime Text, PyCharm etc.

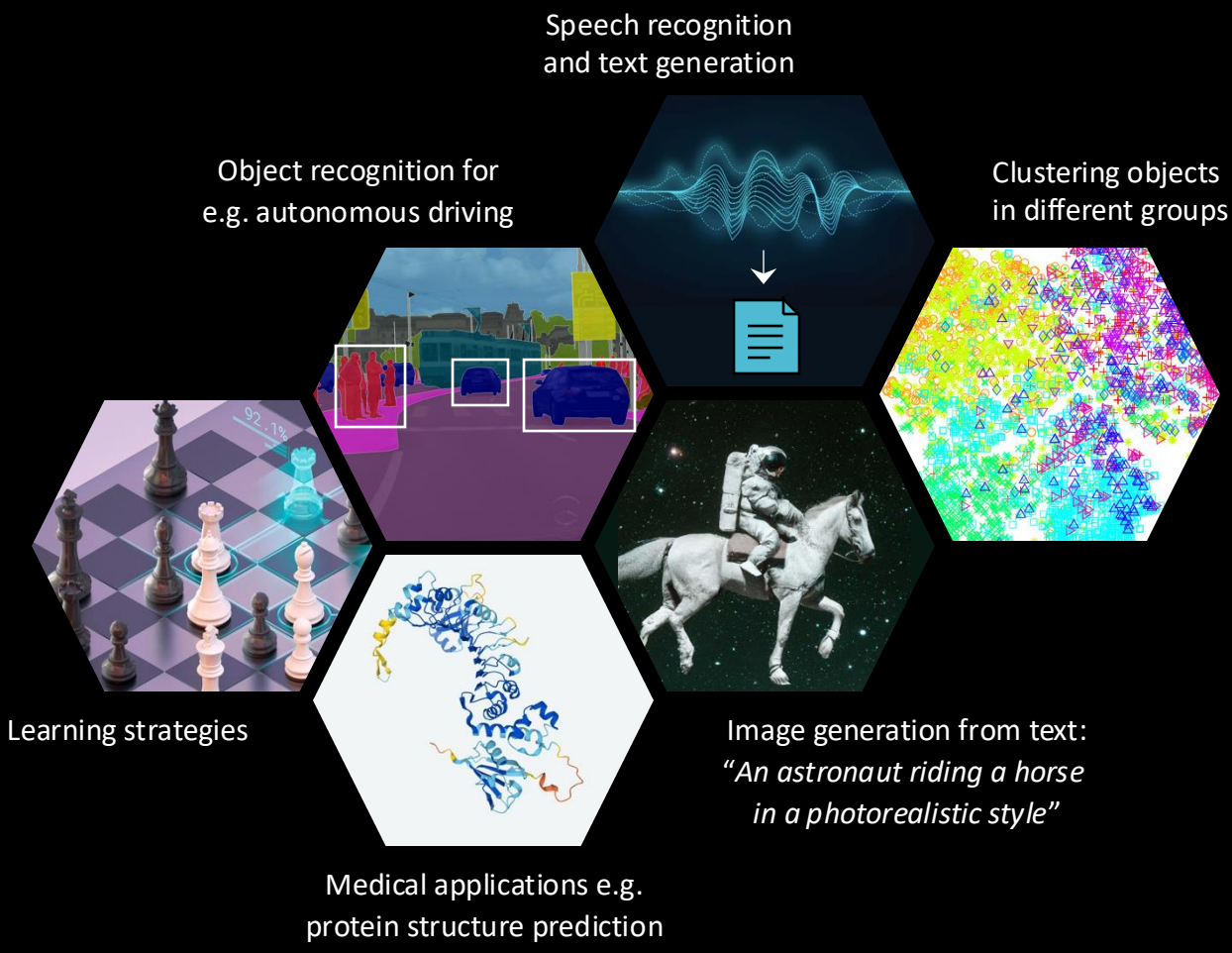
Selection of References

- Quick overview with interactive tutorials on some basic topics (similar to the course) and more advanced concepts: https://www.learnpython.org/
- More detailed overview with interactive tutorials on a lot of topics: https://www.w3schools.com/python/default.asp
- One of my favourites with tutorials on specific (advanced) topics, easy-to-read books: https://www.realpython.com/
- Overview of resources for beginners: https://wiki.python.org/moin/BeginnersGuide

Advanced Python & Machine Learning

- Next course (next spring):
  - More advanced Python concepts
  - Introduction to Machine Learning





Suggestions for the Feedback

- Preparation information / YouTube videos adequate?
- What did / didn’t you like about Noto?
- Content appropriate: anything missing for you?
- Too fast or slow, shallow or deep?
- Insightful exercises?

Thank you and good luck as new Pythonistas! ☺

Some Additional Links

- Download the course material from https://github.com/samarinm/pythonCC
- On script languages: https://en.wikipedia.org/wiki/Scripting_language
- On the difference between interpreter and compiler: https://www.programiz.com/article/difference-compiler-interpreter
- On dynamic and static type checking: https://en.wikipedia.org/wiki/Type_system#Type_checking