Mathematics and Computer Science for Modeling
Unit 1: Introduction to Programming in Python

Daniel Sabinasz
based on materials by Jan Tekülve and Daniel Sabinasz

Institut für Neuroinformatik, Ruhr-Universität Bochum

25.09.2023
Why this course?

- Anyone with a Bachelor’s degree in any of the cognitive sciences can start this Master’s degree
- You will then be exposed to lectures from all of the cognitive science disciplines
Why this course?

▶ Not all of you will have the same level of background knowledge for all of the lectures

▶ The preparatory courses are here to help you bridge that gap

▶ Goal here: Bring you on a similar level regarding mathematics and computer science skills

▶ … which will hopefully make it easier for you to get through the Master programme

▶ The course is not mandatory, but highly recommended
Course concept

- The course is split into lecture parts and exercise parts
Exam

- At the end of the course, there will be a written exam (04.10. at 3 pm)
- The exam is graded, but this is only for your feedback and won’t enter into your average grade
About Me

- My name is Daniel Sabinasz
- B.Sc. computer science and M.Sc. cognitive science
- PhD candidate at the Institute for Neural Computation
- Working on mathematical modeling of the neural processes that underlie language understanding
- Email me with any questions you might have: daniel.sabinasz@ini.rub.de
Dates

1. Mon 25.09. 15-17:30
2. Tue 26.09. 09:00-11:30, 15-17:30
3. Wed 27.09. 15-17:30
4. Thu 28.09. 15-17:30
5. Fri 29.09. 15-17:30
6. Mon 02.10. 09:00-11:30, 15-17:30
7. Wed 04.10. 15-17:30
# Course Structure

<table>
<thead>
<tr>
<th>Unit</th>
<th>Title</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro to Programming in Python</td>
<td>Variables, if Statements, Loops, Functions, Lists</td>
</tr>
<tr>
<td>-</td>
<td>Full-Time Programming Session</td>
<td>Deepen Programming Skills</td>
</tr>
<tr>
<td>2</td>
<td>Functions in Math</td>
<td>Function Types and Properties, Plotting Functions</td>
</tr>
<tr>
<td>3</td>
<td>Linear Algebra</td>
<td>Vectors, Trigonometry, Matrices</td>
</tr>
<tr>
<td>4</td>
<td>Calculus</td>
<td>Derivative Definition, Calculating Derivatives</td>
</tr>
</tbody>
</table>
## Course Structure

<table>
<thead>
<tr>
<th>Unit</th>
<th>Title</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Integration</td>
<td>Geometrical Definition, Calculating Integrals</td>
</tr>
<tr>
<td>6</td>
<td>Differential Equations</td>
<td>Properties of Differential Equations</td>
</tr>
<tr>
<td>-</td>
<td>04.10.23: Test</td>
<td></td>
</tr>
</tbody>
</table>
Lecture Slides/Material

Use the following URL to access the lecture slides:

https://www.ini.rub.de/teaching/courses/preparatory_course_mathematics_and_computer_science_for_modeling_winter_term_2023
Getting Started

▶ Install Anaconda: https://www.anaconda.com/distribution/

▶ Download the document “Jupyter notebook” for Unit 1 (filename “unit1.ipynb”) from the course website

▶ Start the program “Anaconda-Navigator”. Find the application “Jupyter Notebook” and click on “launch”.

▶ (Alternative: Start the program “Anaconda Prompt”. Wait for a prompt to appear and then enter “jupyter notebook” into that prompt)

▶ Navigate to the directory where you saved the “unit1.ipynb” file and click on that file
Getting Started

You are now presented with a so-called Jupyter Notebook, a document that allows you to execute existing Python code and write your own Python code while being guided by narrative text.

First steps

Print

The `print` function writes something to the screen.

```python
In [1]: print("Hello World!")
```

Scripts

A script is a series of commands. Code is executed from top to bottom - one line after each other.

```python
In [1]: print("Hello There!")
    print("I haven't seen you in a while.")
    print("How are you?")
```

You can write comments in your code using the '#' character.

```python
In [1]: print("Hello!") #This is a comment
    # Lines that start with # are ignored
    print("How are you?")
    #print("I am bored!") This line is ignored
```

Exercise: First steps

Write a program that prints your name to the screen two times.

```python
In [1]:
```
Getting Started

First steps

Print

The `print` function writes something to the screen.

```python
In [1]: print("Hello World!")
Hello World!
```