efl the Data Science Institute

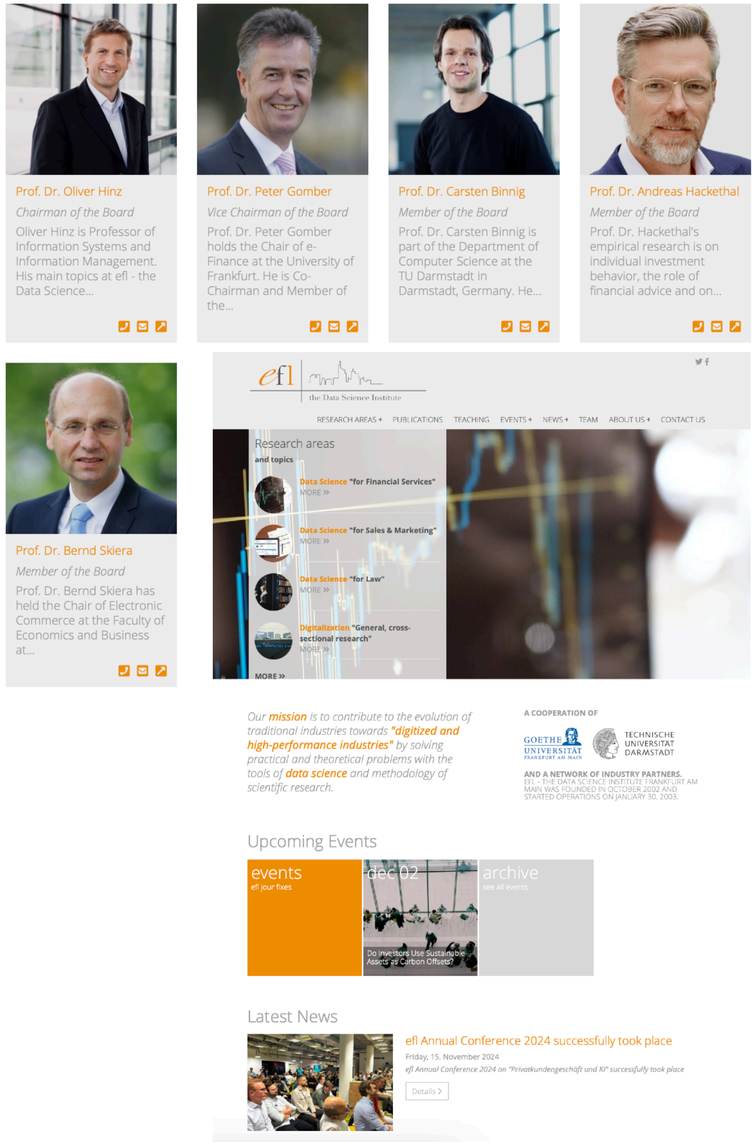
efl Python Course
Introduction

Version 1.0.0
2024-12-06
By Dr. Benjamin M. Henrich

About us: Who We Are?

Mission (Since 2019):

- Use expertise in Data Science to deliver cutting edge research in the fields of
  - Financial Services
  - Retail & Marketing
  - Health
  - Law
  - General, cross-sectional research



About us: The elf

Industry - Academic Partnership

Universities

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

TECHNISCHE UNIVERSITÄT DARMSTADT

Sponsors

FACTSET

AMERICAN EXPRESS

Deutsche Leasing

finanz informatik

DEUTSCHE BÖRSE GROUP

DZ BANK Gruppe

About US

Lecturers

M.Sc. Anjana Cordes

Research Assistant

Anjana Cordes joined the efl - the Data Science Institute as a research assistant in October 2022.

M.Sc. Tino Cestonaro

Research Assistant

Tino Cestonaro joined the team in April 2020. His research focuses on Market Microstructure, Financial Machine Learning,...

M.Sc. Florian Ewald

Research Assistant

Florian Ewald joined the efl research team in May 2023. His research focuses on the application of machine learning and...

Dr. Benjamin M. Henrich

Alumnus

Since March 2018, Benjamin M. Henrich is a research assistant of Prof. Dr. Oliver Hinz at the Chair of Information...

Agenda

Teaching

Day 1 (Python Course) (06.12.2024)

9:00 - 10:30 Uhr
Python Basics
Introduction and Primitive Data Types

10:40 - 12:10 Uhr
Data Structures
Tuples, Lists, Sets, Dictionaries

13:30 - 15:00 Uhr
Control Structures
Loops, Conditional Statements

15:15 - 16:45 Uhr
Structure of Functions and Application
Functions in Python

Day 2 (Python Course) (13.12.2024)

9:00 - 10:30 Uhr
LIbraries in Python
Os, numpy, re, csv

10:40 - 12:10 Uhr
Data Wrangling Libraries in Python
Pandas, dataframes

13:30 - 15:00 Uhr
Data Visualization
Matplotlib, seaborn

15:15 - 16:45 Uhr
Data Analysis with Python
Merging data, explanatory analysis

Day 3 (Data Science) (20.12.2024)

9:00 - 10:30 Uhr
Introduction to Data Science
Terminology and basic concepts

10:40 - 12:10 Uhr
Preprocessing
Normalization, train-test split

13:30 - 15:00 Uhr
Simple Model Application
Decision trees, random forests

15:15 - 16:45 Uhr
Deep Learning
Neural Networks, keras

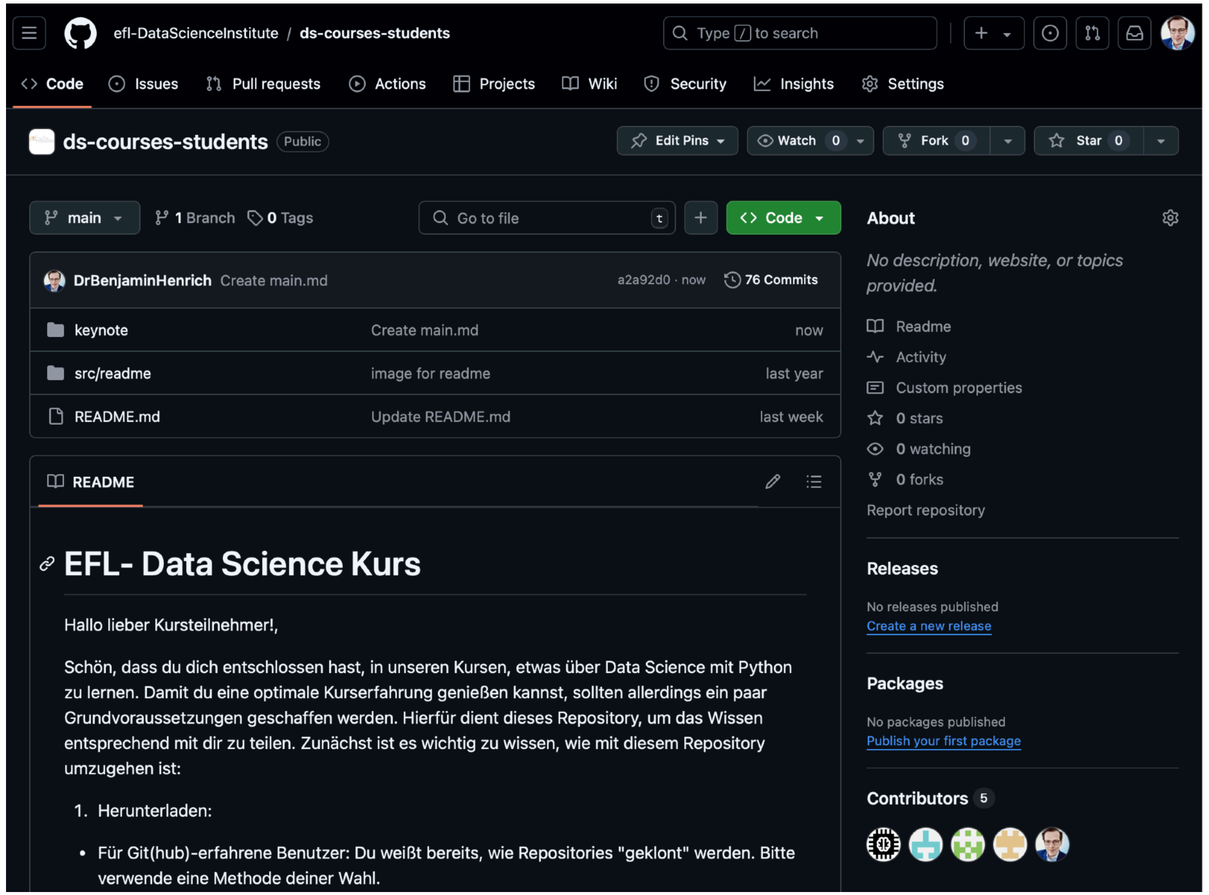
https://www.eflab.de/teaching

Certificates

- Completion of tasks from the last class
- Submission: Description of how the tasks were solved

- Formalities:
  - Min. 2 pages (Arial 11, 1.5 Line spacing, 3 CM Correction margin)
  - Code must be delivered separately (code folder)

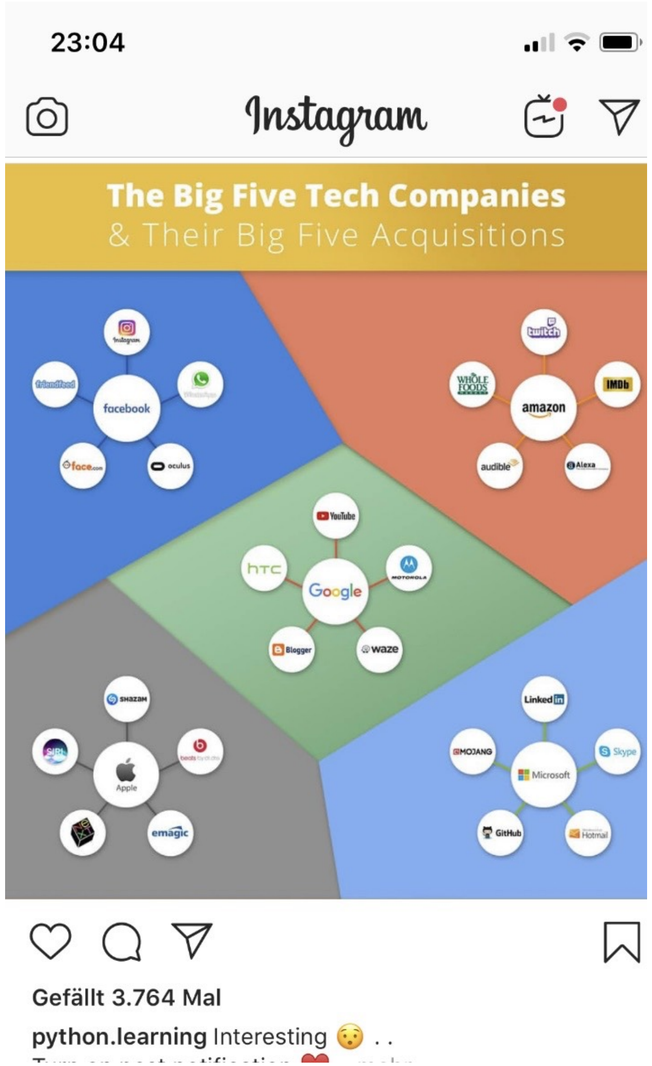
Course Material?



https://github.com/efl-DataScienceInstitute/ds-courses-students

Why Are You Here?
What is your Motivation?
Why Coding?

Why Coding?



NACH ABSCHLUSS IN ...

INFORMATIK, MATHEMATIK, WIRTSCHAFTSINFORMATIK

	TOP BERUFSTITEL*	TOP BRANCHE
1	IT Manager **92.275 €**	Banken **85.067 €**
2	IT-Projektmanager/in **86.695 €**	Chemie- und Erdöl-verarbeitende Industrie **84.033 €**
3	Senior Developer **78.635 €**	Konsum- und Gebrauchsgüter **82.586 €**
4	Analyst **74.469 €**	Unternehmensberatung, Wirtschaftsprüfung und Recht **82.336 €**
5	Software-Architekt/in **74.152 €**	Finanzdienstleister **81.331 €**

*Berufstitel ohne Management- & Personalverantwortung
Bruttodurchschnittsgehalt (inkl. variabler Bezüge), Mittelwert

11 Stepstones - Gehaltsreport

Why Coding? (Personal View)

Coding Rewires Your Brain!

Concept	Explanation
Abstraction	Hiding complex details and exposing only the essential functionality. Abstraction helps in simplifying system complexity, making the code more understandable and easier to manage.
Coupling	Refers to the degree of dependency between different modules or components of a system. **Tight coupling** means high interdependence, while **loose coupling** is preferred for better maintainability and flexibility.
Continuous Integration (CI)	A development practice where code changes are automatically built and tested to ensure they integrate well into the main codebase. It helps in identifying integration issues early and speeds up development cycles.
Divide and Conquer	A problem-solving strategy where a problem is broken down into smaller, manageable sub-problems. Each sub-problem is solved independently, and results are combined. Common in algorithms like Merge Sort and Quick Sort.
Modularity	The design principle that divides software into smaller, independent, and interchangeable modules or components. Each module is responsible for a specific functionality and can be developed, tested, and maintained independently.
Refactoring	The process of restructuring existing code without changing its external behavior. It improves code readability, reduces complexity, and makes it easier to maintain without introducing new bugs.
Never Repeat Yourself (DRY)	A principle in software development aiming to reduce repetition of software patterns. Repeating code leads to redundancy, maintenance issues, and bugs. The goal is to use reusable functions or modules.
Test-Driven Development (TDD)	A software development process where tests are written before the code itself. This ensures that the code meets its specifications and helps catch bugs early in the development cycle.

Fail Fast!

What is Coding?

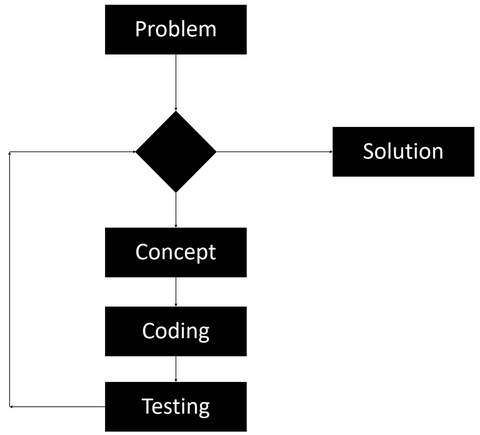
WIKIPEDIA
The Free Encyclopedia

Computer programming

Article Talk

From Wikipedia, the free encyclopedia

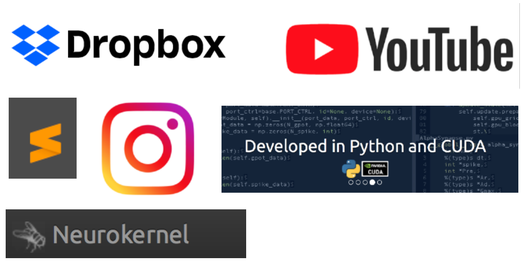
Computer programming or **coding** is the composition of sequences of instructions, called programs, that computers can follow to perform tasks.[1][2] It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.



https://en.wikipedia.org/wiki/Computer_programming

Motivation: Why Python?- Part I

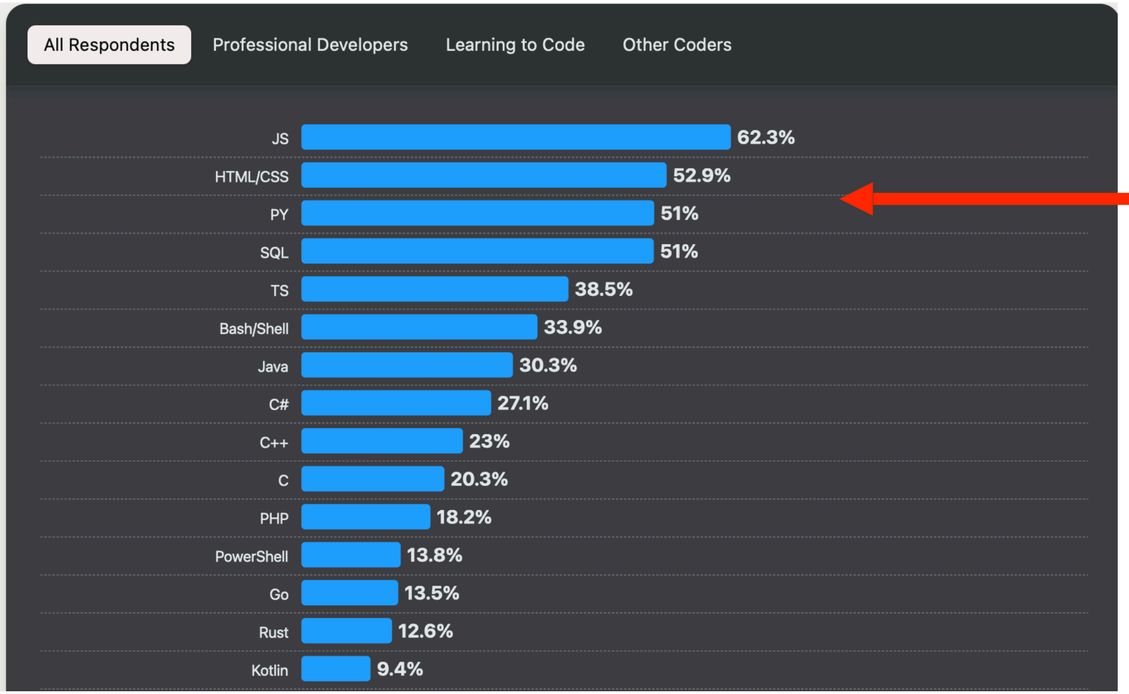
- Released in 1991 by Guido van Rossum
- With the explosive growth of ‘big data’ in disciplines such as bioinformatics, neuroscience and astronomy, programming know-how is becoming ever more crucial (Perkel 2015, p. 125).



Programming, scripting, and markup languages

JavaScript has been a mainstay in the developer survey and on Stack Overflow since our first survey. The most popular programming language has been JavaScript every year we have done the survey except for 2013 and 2014, when SQL was the most popular language.

Which **programming, scripting, and markup languages** have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the language and want to continue to do so, please check both boxes in that row.)



Motivation: Why Python - Part II

```java
public class Test {
    public static void main(String args[]) {
        String array[] = {"Hello, World", "Hi there, Everyone", "6"};
        for (String i : array) {
          System.out.println(i);
        }
    }
}
```

Java

```python
stuff = ["Hello, World!", "Hi there, Everyone!", 6]
for i in stuff:
    print(i)
```

Python

https://raygun.com/blog/java-vs-python/

Motivation: The Elementary Basics in Python

Pros Python:

- General purpose language
- Higher programming language
- It´s simple
- Fast to read
- Structuring by indenting
- No {} or ; => Faster to Code
- Data types are managed dynamically. There is no static type check like in java
- Widespread in science
- Extensive Support Libraries (important data science, math and many more)
- Integration Feature
- Productivity (Many Frameworks such as unit testing)

Ozgur, C., Colliau, T., Rogers, G., & Hughes, Z. (2017). MatLab vs. Python vs. R. Journal of data Science, 15(3), 355-371.
Abdel-Karim, B. M. (2022). Data Science. In Data Science: Best Practices mit Python (pp. 57-62). Wiesbaden: Springer Fachmedien Wiesbaden.

Cons of Python

- Python is an interpreted language, this is slow as compared to C/C++
- See Interpreted vs. Compiled programming languages
- Python can have runtime errors (dynamical typing feature)
- Consumes a lot of memory space
- Not easy to test

Small Satellite*

*) https://www.nbcnews.com/id/wbna43313086

May not in Python …

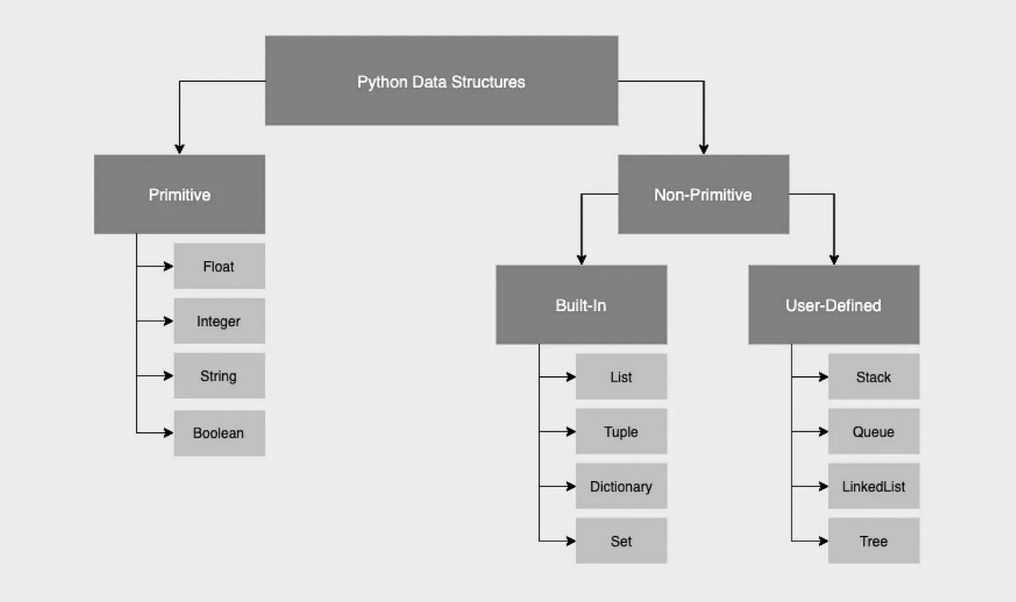
Prechelt, L. (2000). An empirical comparison of c, c++, java, perl, python, rexx and tcl. *IEEE Computer*, *33*(10), 23-29.
Newhall, T. K. (1999). Performance Measurement of Interpreted, Just-in-Time compiled, and Dynamically Compiled Executions. The University of Wisconsin-Madison.

Agreement

- Introduction to programming => All Questions are allowed!!!
- For beginners
- The module is interactive => Use your computer
- **We develop the solutions together!**
- Please be on time
- There are no dumb questions
- Nobody knows everything
- Copying solutions is plagiarism

A Glimpse Into Content

Primitive Data Structures

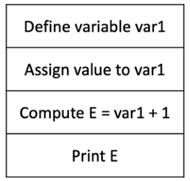


	Mutable	Ordered	Indexing	Duplicate Elements
Tuple		X	X	X
List	X	X	X	X
Set	X			
Dictionaries	X		X	

Control Structures in Python

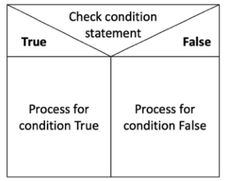
Process blocks

- No conditions
- Execute action inside of block and move to next block



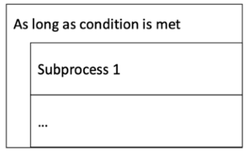
Branching blocks

- Check condition statement
- Execute action inside of respective block

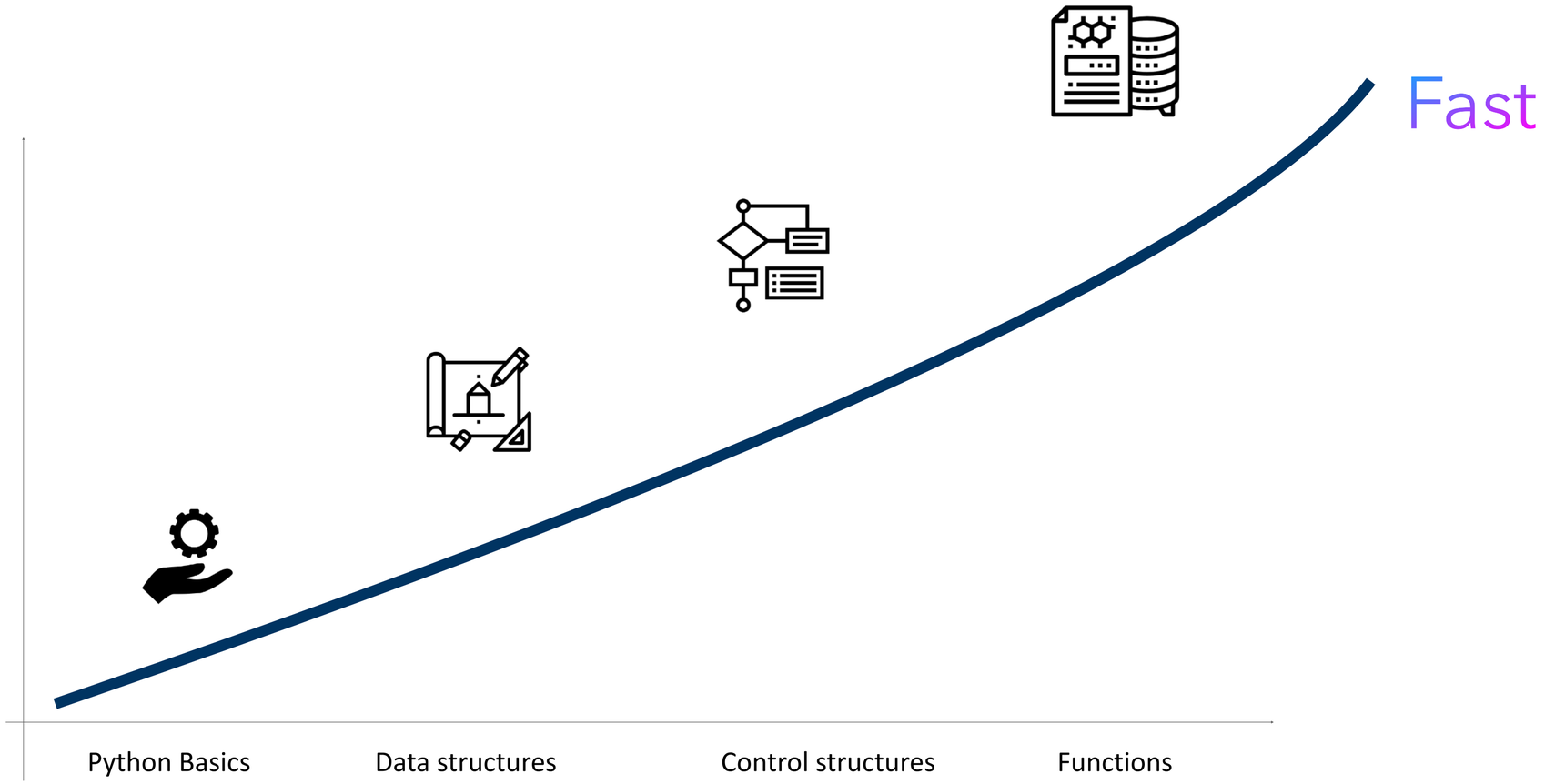


Loops

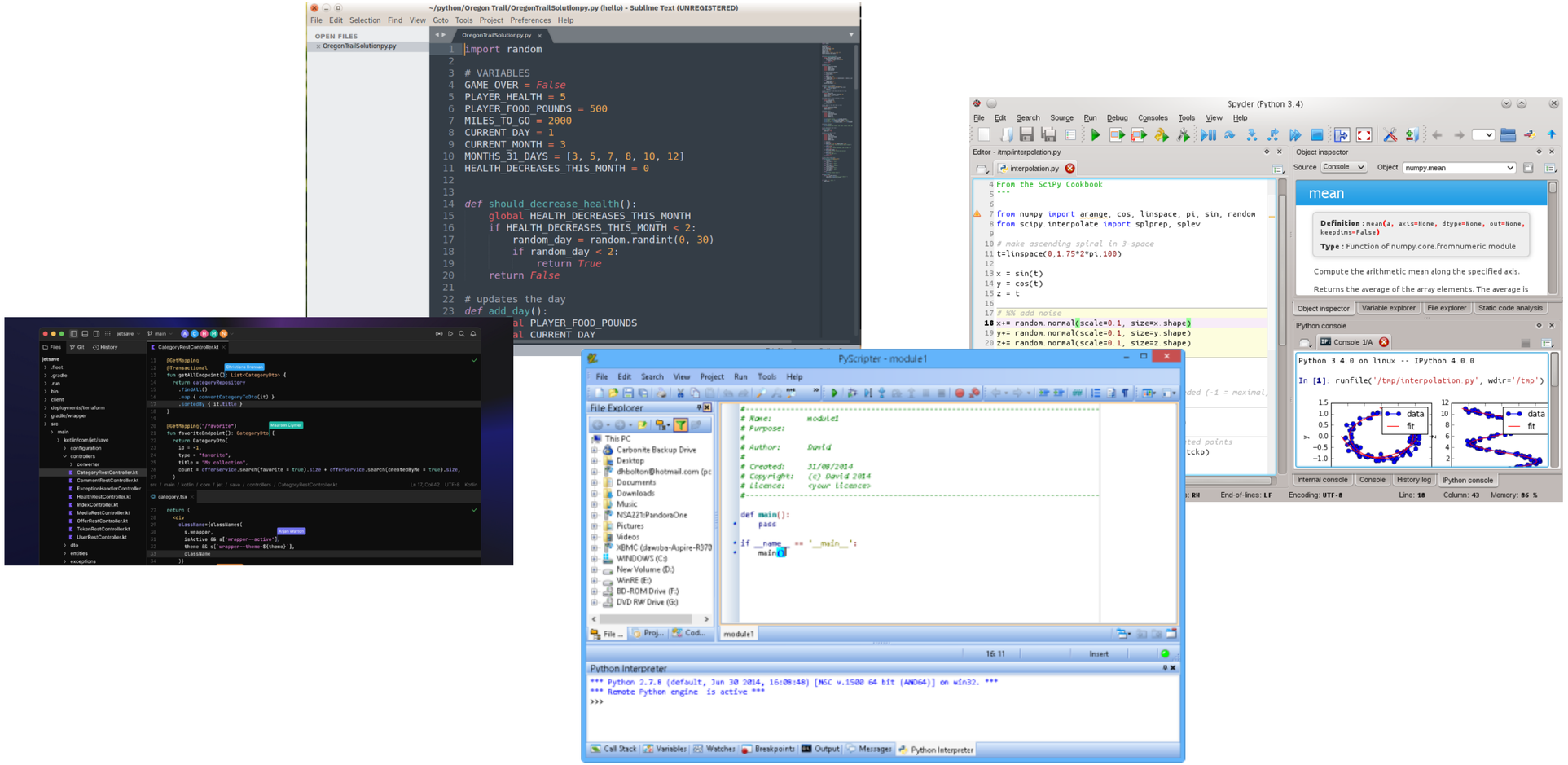
- Loop a process as long as particular condition is met



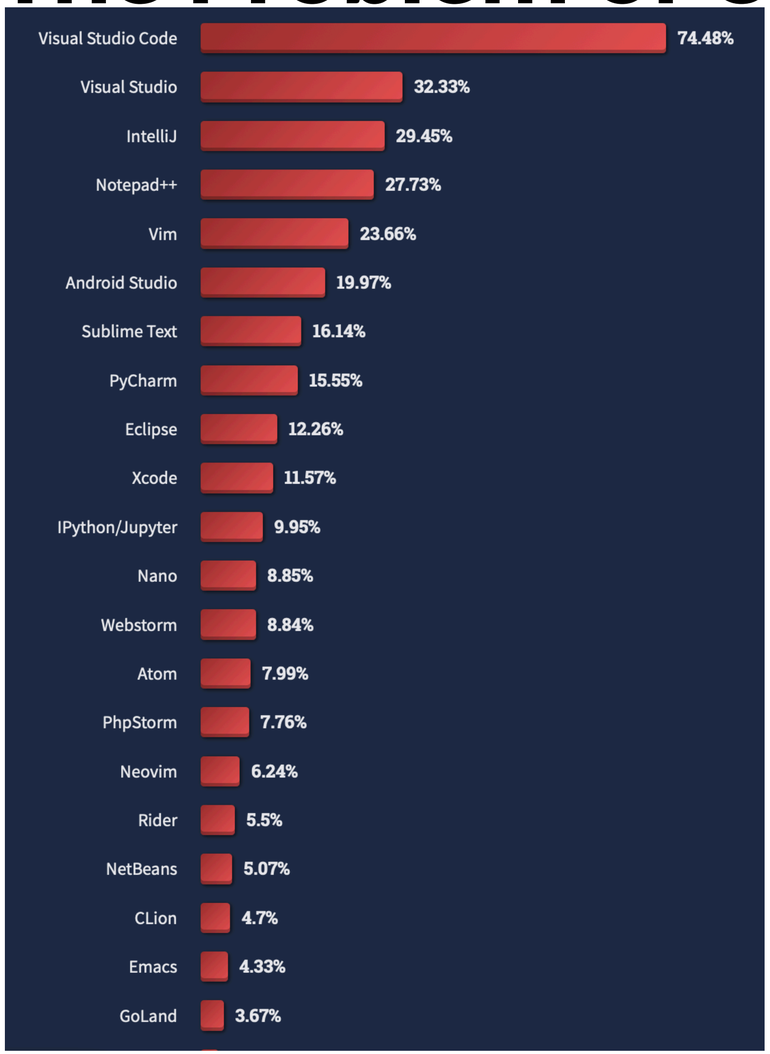
Note: Learning Curve for Today



Integrated Development Environment
IDE a Deeper look? The Problem of Choice

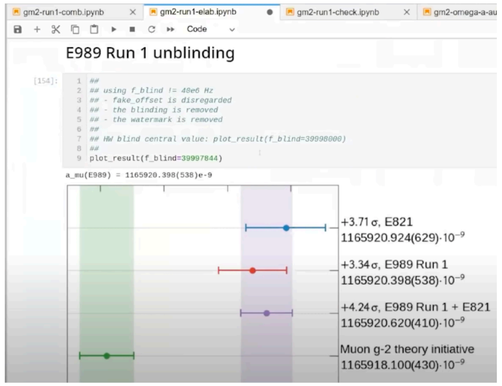


The Problem of Choice

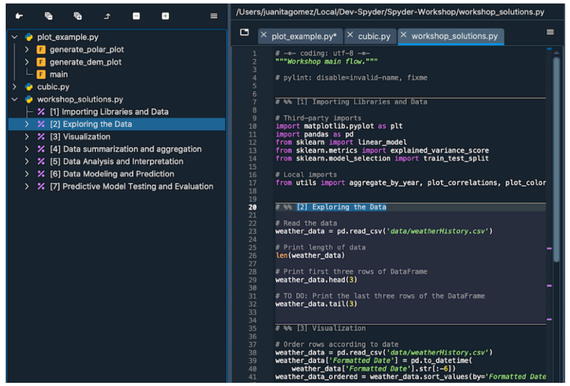


https://survey.stackoverflow.co/2022/#most-popular-technologies-new-collab-tools-prof

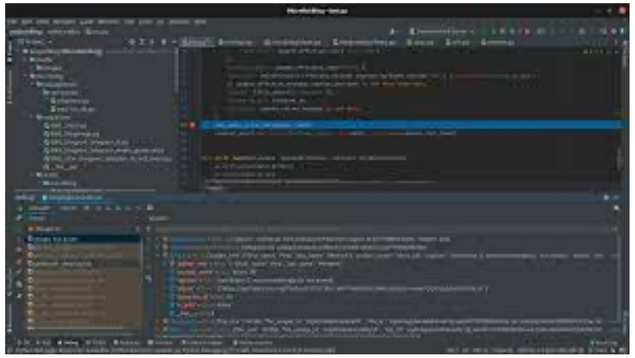
A Question of Style



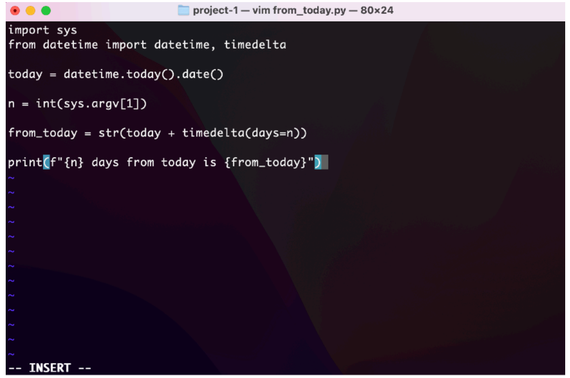
Notebooks (Research)*



Hybrid #%% (standard cell separator)



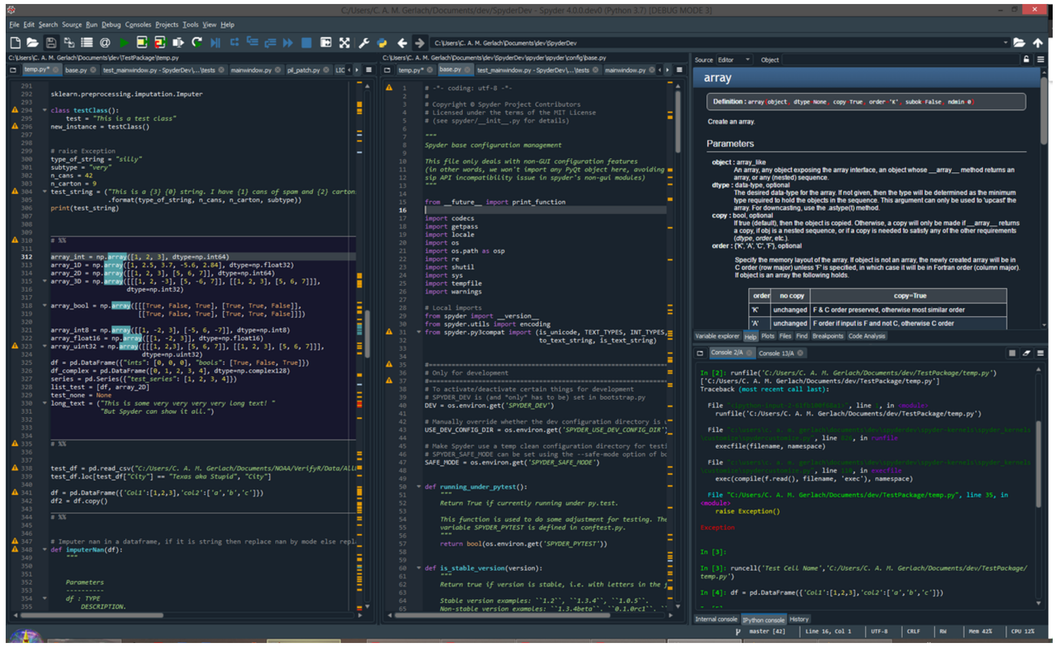
Classical IDE Coding



Coding via Terminal (Ultra Mode)

*) Famous experiment: https://www.youtube.com/watch?v=hL2w_FkTae8

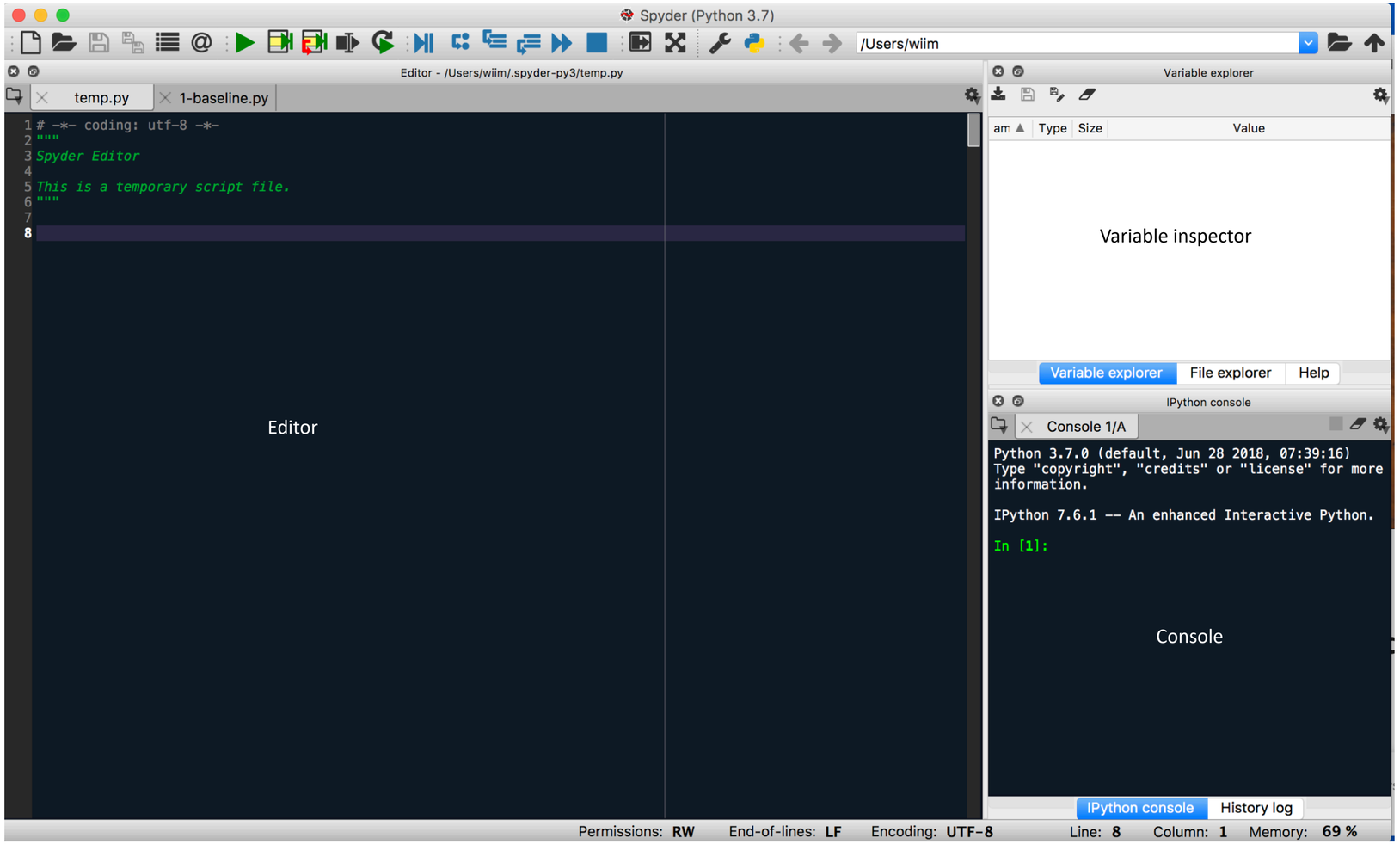
What We Use for This Course



Practice and Questions

- Take your computer and let's get started!
- Create a Folder ‘Day 1’ on our Desktop
- Launch Spyder and select the folder

In a Nutshell:
Integrated Development Environment



In a Nutshell: Hello World

```
print('Hello World')
```

Abstraction (Computer Science)
The first step - The concept of variable

Variable

Specific value

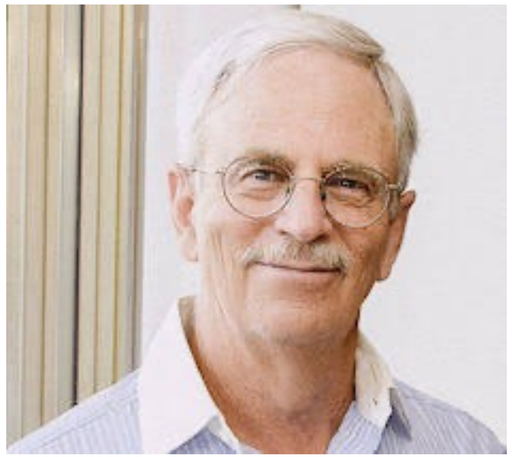
```
int_one = 1
```

Allocation

We assign the value 1 to the variable 'one’.
Now we can continue working with the variable ‘one’.
Advantage: We are independent of concrete value

The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context.

– John V. Guttag



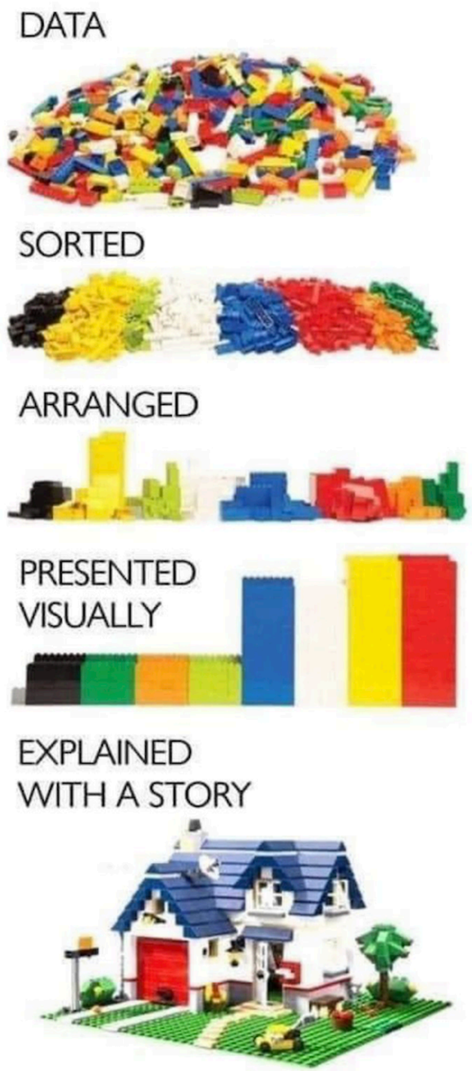
Different Building Blocks

- Integer: int (signed integers) – They are often just called integers or ints, are positive or negative whole numbers with no decimal point.

- Float: Float (floating point real values, double) – Also called floats, they represent real numbers and are written with a decimal point dividing the integer and fractional parts. Floats may also be in scientific notation, with E or e indicating the power of 10.

- Springs are a sequence of chars. We can create them simply by enclosing characters in quotes. “Hello World” is a String! Therefore strings in Python are bytes representing Unicode characters. In Detail: Python does not have a character data type, a single character is simply a string with a length of 1.

- Boolean: We can perform logical operations with True and False in combination with AND and OR.



Primitive Data Structures

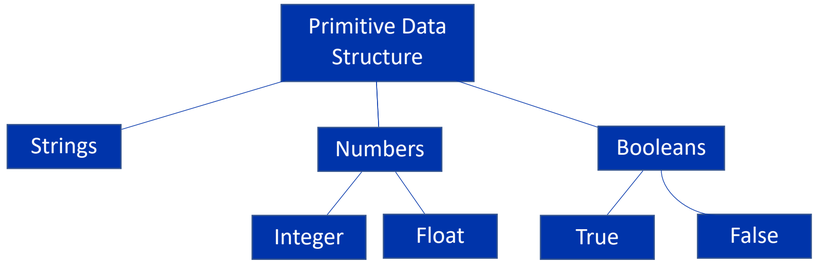
- Integer: int (signed integers) – They are often just called integers or ints, are positive or negative whole numbers with no decimal point.

- Float: Float (floating point real values, double) – Also called floats, they represent real numbers and are written with a decimal point dividing the integer and fractional parts. Floats may also be in scientific notation, with E or e indicating the power of 10.

- Springs are a sequence of chars. We can create them simply by enclosing characters in quotes. “Hello World” is a String! Therefore strings in Python are bytes representing Unicode characters. In Detail: Python does not have a character data type, a single character is simply a string with a length of 1.

- Boolean: We can perform logical operations with True and False in combination with AND and OR.

Primitive Data Structure: A First Overview



* Simplified Illustration

Python Documentation: https://docs.python.org/3/library/stdtypes.html
Other classification: Chun, W. (2001). Core python programming (Vol. 1). Prentice Hall Professional.

Documentation

Source Code Documentation

@author: name

@since: first implementation date

@version: date of last update

@source: if you using links etc.

@code: special code note

@param: if special parameter is used or you have to describe.

```
# I am a comment
@author: My Name
@since: 2022-10-03
@update: 2023-10-26
@version: v.0.0.2
print('Hello World')
```

More Information: PEP 8 – Style Guide for Python Code: https://peps.python.org/pep-0008/

Naming Convention

Select what you like! But be consistent!

- Names of attributes, variables, functions/methods* start with a small letter
  - may use letters without ß or similar
  - which points to the data type like i, s or l
- This is standard in professional software development.
- Camel Case: Compound words are written in programming language. Every new word is capitalized.

(More PeP8 Style)

- name = Is the name of…
- bscript_main = Simple code file that does something
- CName = Class (later more)
- float_name = Variable that saves a floating point (double)
- int_name = Variable that saves an integer value
- str_name = Variable that saves a string value
- b_name = Boolean for true or false values
- list_name = Object from type list
- method_name = Self-written function

(Camel Case Style)

- Name = Is the name of…
- bScriptName = Simple code file that does something
- CName = Class (later more)
- dName = Variable that saves a floating point (double)
- iName = Variable that saves an integer value
- sName = Variable that saves a string value
- bName = Boolean for true or false values
- LName = Object from type list
- fName = Self-written function

*) A **function** is a part of code that is called by the name definition. It can do an operations by the input parameters and can optionally return values).

A **method** is a part of an Object, is called by name that is associated with an object

Code is some kind of Art. Therefore...

Jackson Pollock Art

These are all approaches. Find your own style!

Let’s Do Coding

Thank You


YouTube
Suchen
Data Science in a Nutshell
Dr. Benjamin M. Abdel-Karim
Data Science in a Nutshell - Einleitung
TEILEN
SPEICHERN


Benjamin M. Abdel-Karim Hrsg.
Data Science
Best practices mit Python
EBOOK INSIDE
Springer Vieweg