HSLU Lucerne University of Applied Sciences and Arts

Python for Data Science: SW01

Script Language
Basic Data Types
Mathematical Operations

Information Technology

February 20, 2025

FH Zentralschweiz

Content

- Administrative / Introduction
  - Course material
  - Python: interpreter vs. compiler language
- Intelligent Development Environment (IDE)
  - PyCharm installation
  - PyCharm introduction with local project
  - Python console vs. script
  - Virtual development environment: venv
- Python Basics
  - Variables and basic data types
  - Mathematical operations

Main goals

PyCharm installed and running | First Python exercises

HSLU Lucerne University of Applied Sciences and Arts

Administrative / Introduction

Information Technology

February 20, 2025

Teaching team

Andreas Melillo

Dozent

andreas.melillo@hslu.ch

Ramón Christen

Sen. Wissenschaftlicher Mitarbeiter / Doktorand

ramon.christen@hslu.ch

Tim Giger

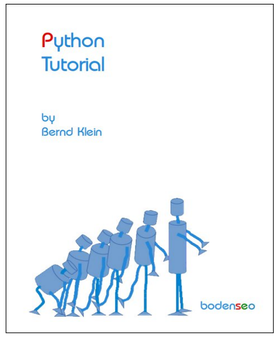
Wissenschaftlicher Mitarbeiter

tim.giger@hslu.ch

Course material

Our lecture is based on:

Einführung in Python 3 (4. Auflage) or **Python Tutorial** (english version)



Bernd Klein
Hanser Verlag
ISBN: 978-3-446-46379-0

URL:

- https://www.python-kurs.eu/ (german)
- https://www.python-course.eu/ (english)
- https://www.python-course.eu/bernd_klein_python_tutorial_a4.pdf (english)

Online course on tutorials.eu (coupon only valid for a few days)

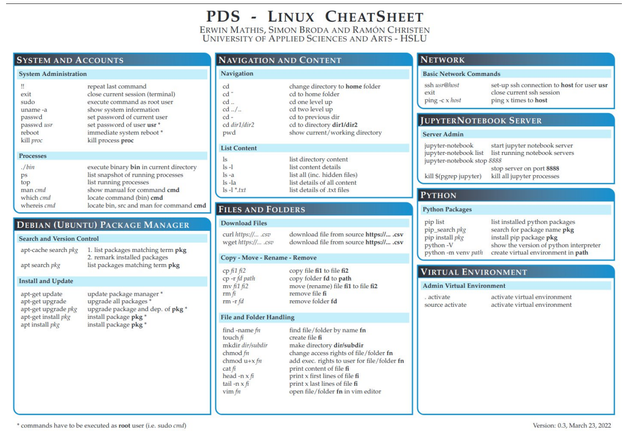
- **The Complete Python 3 Masterclass – From Beginner to Pro**
  https://academy.tutorials.eu/p/the-complete-python-3-masterclass-from-beginner-to-pro-live?coupon_code=HSLUSS252&product_id=5867708

Course material

Official Python documentation
https://docs.python.org/3/

Ad-hoc exercises on ilias
PDS course container > section “Exercises”

Cheat sheets
Linux: on ilias in PDS course container > section “Supporting Material”
Python: official Python beginner’s guide https://wiki.python.org/moin/BeginnersGuide



Before we start

Comment on course content:

- The course content is currently rebuilt
- Agenda (-> ILIAS) for the second half of the semester may still change

Comment on typical lecture plan:

- Start with theoretical inputs, with live coding examples
- End with exercise sessions
- Passive streaming on onsite lectures

Comment on inhomogeneous backgrounds:

- How many of you have experience with programming?
- How many of you have experience with python?
- We try our best to keep the lessons interesting for everybody!

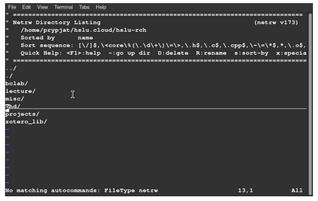
Python: interpreter vs. compiler language

Interpreter (script) Language

- Ex: Bash, JavaScript, Matlab, Python, R, Ruby

Human interpretable source code

Interpreter (interprets and sends code to CPU)



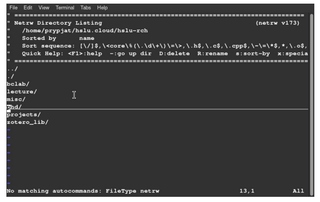
Console / UI

Compiler Language

- C, C++, C#, Java

Human interpretable source code

Compiler (optimizer)



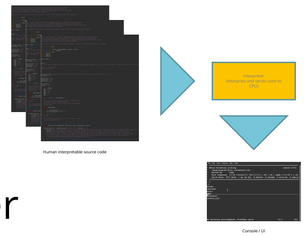
Console / UI

Machine (CPU) code package

Python: interpreter vs. compiler language

Interpreter (script) Language

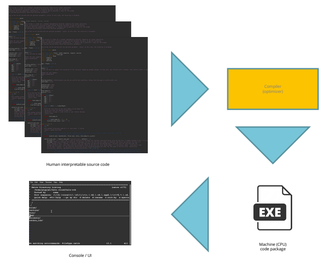
- Ex: Bash, JavaScript, Matlab, Python, R, Ruby



- Code **line by line** sent to interpreter
- Interpreter provides console
  -> ad-hoc commands
- No optimization
- Slow in execution

Compiler Language

- C, C++, C#, Java



- Compiles source code and returns an executable file
- Compiler can optimize code
  -> executable usually compressed
- Executable can never be completely de-compiled
  -> hard to reverse-engineer

HSLU Lucerne University of Applied Sciences and Arts

Intelligent Development Environment IDE

Information Technology

February 20, 2025

Intelligent Development Environment (IDE)

Intelligent Development Environments (IDE) typically comprise supporting additional tools for convenient programming.

Provided tools may include:

- Debugger
- Deployment chain control
- File browser
- Spell checker
- Terminal
- Todo list
- Remote development
- Remote execution
- Version control interface

Alternatives:

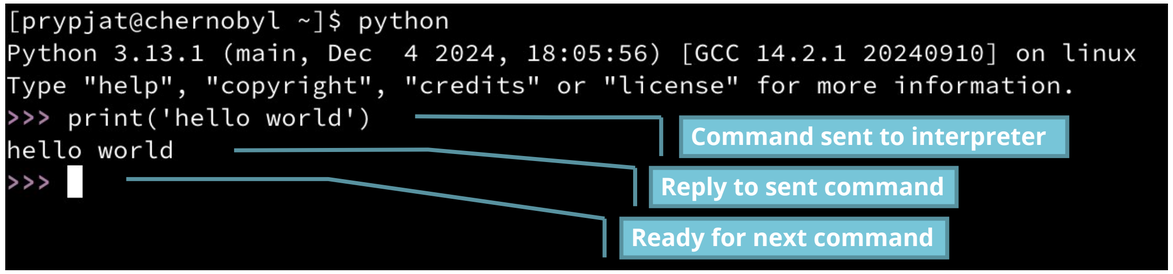
- Any text editor

Python console vs. script

Python is an interpreter language. That means, every single instruction is sent **sequentially** to the interpreter.

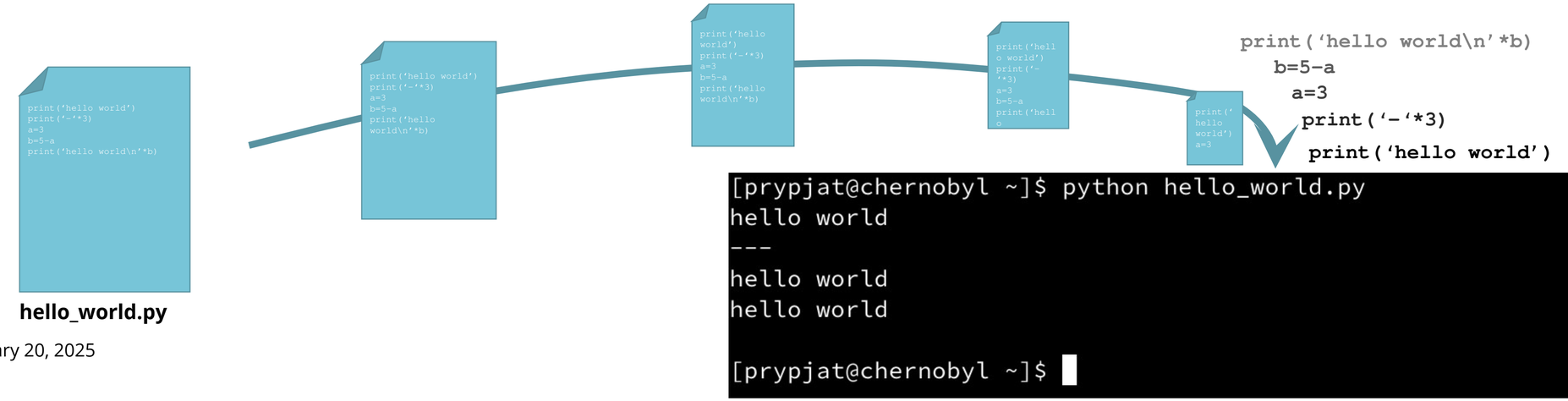
Python **console** is the direct interface (connection) to the interpreter.

- Sending a command in the console to the interpreter, immediately returns the result on the next line.



A Python **script** is a collection of commands.

- Executing a Python script means, sending all contained commands consecutively to the interpreter.

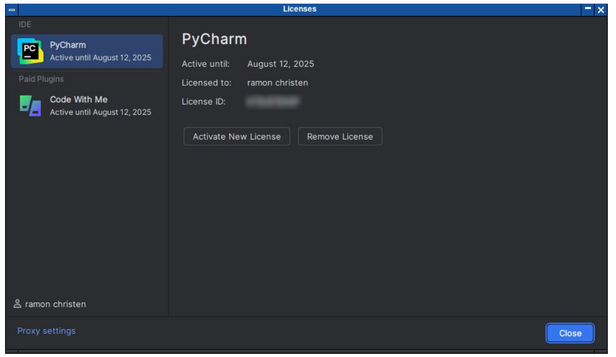


PyCharm installation

PyCharm is provided by JetBrains and available in two editions:
(for Mac, Windows or Linux)

1. Community edition
   - free license
   - limited functionality

2. Professional edition
   - free for student and lecturer with annual education license
   - supports remote app development
   - Jupyter Notebook integration

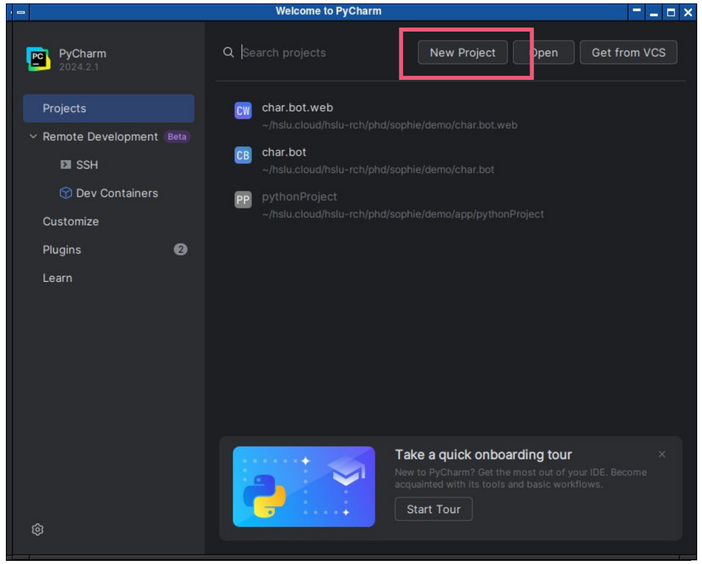


Installation of professional edition (including Python interpreter):

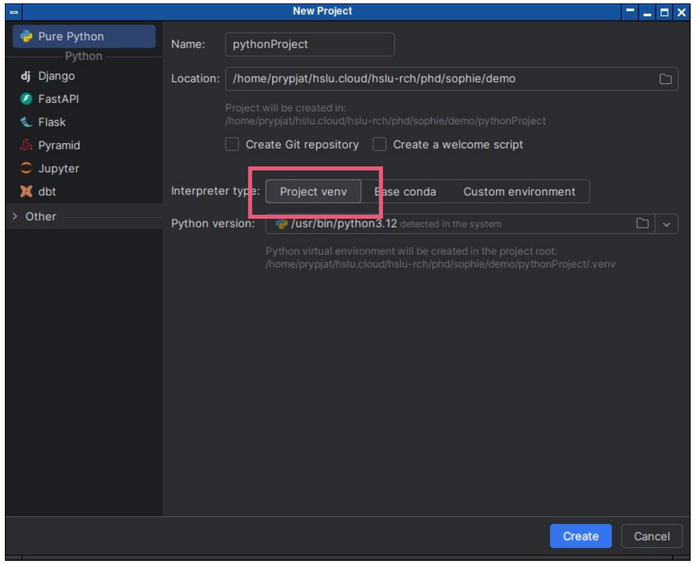
1. download PyCharm professional from https://www.jetbrains.com/
   for Linux also available in snap: `snap install pycharm-professional`
2. install IDE as any other regular app
3. create JetBrains education account with university's mail address by registering your app
   (without an open project: settings (gear on bottom left) > Manage Licenses… | else: Menu > Help > Register…)

PyCharm introduction with local project

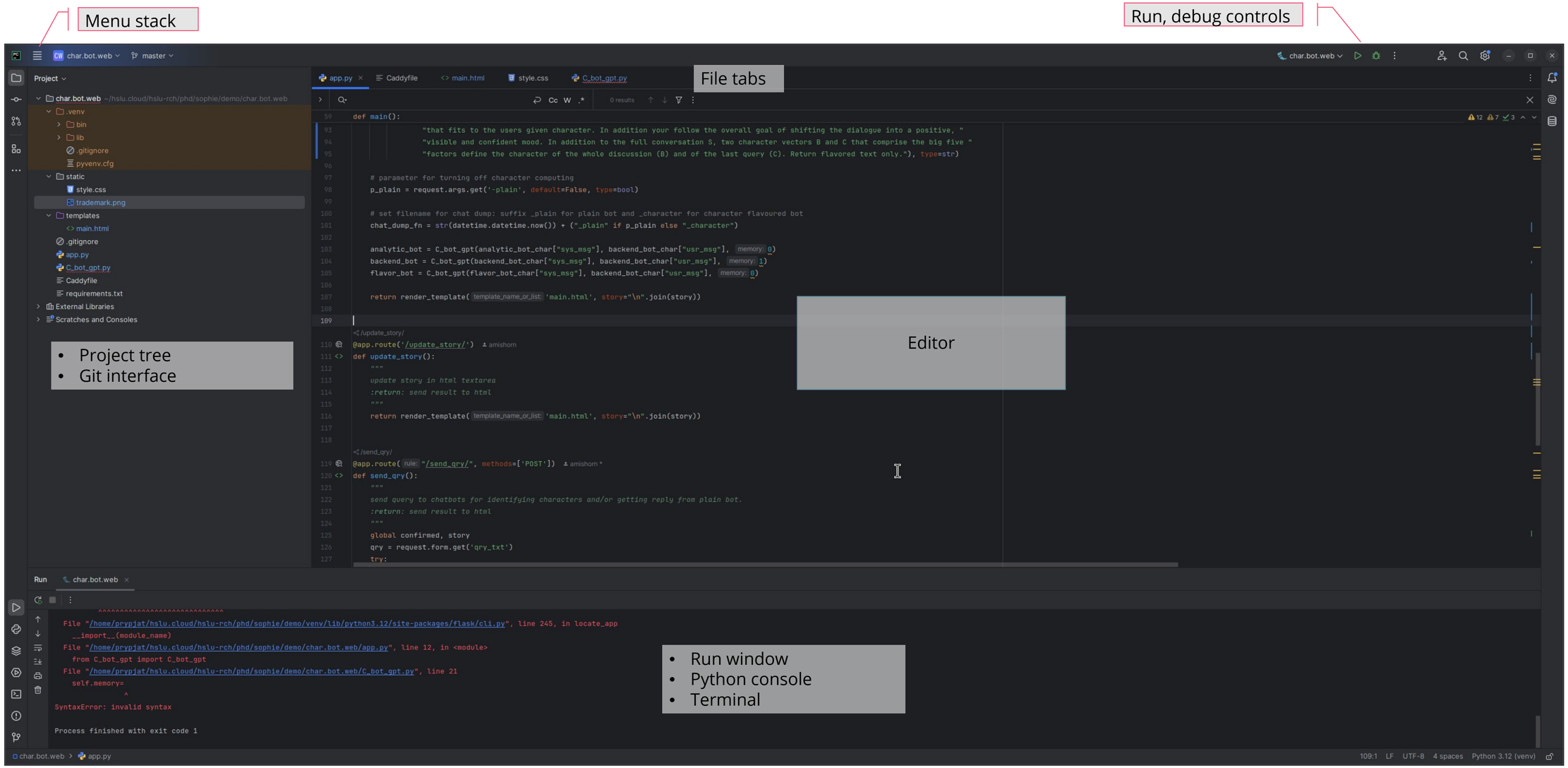
1. Start PyCharm
2. Create New Project



1. Set appropriate project name
2. Select project location
3. Choose project venv (virtual environment will be created in project root)



PyCharm introduction with local project

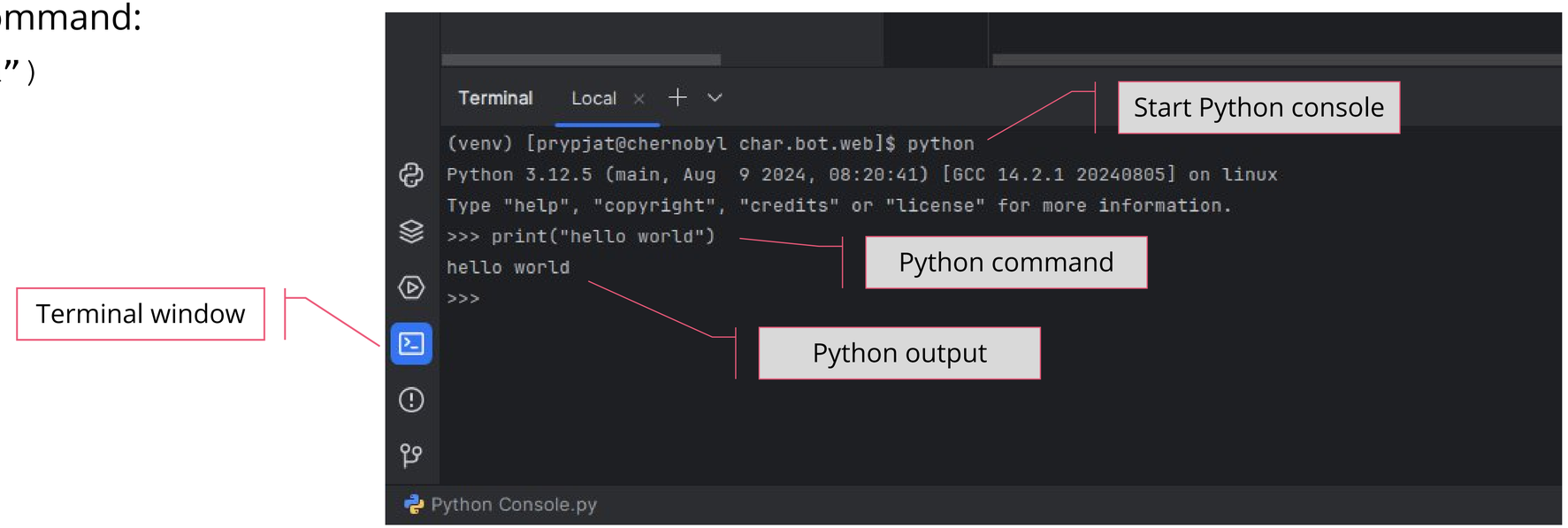


PyCharm introduction with local project

First “hello world” with Python:

1. Switch to the console window at the bottom
2. Activate the terminal
3. Start the Python console
4. Enter following print command:

```
print(“hello world”)
```

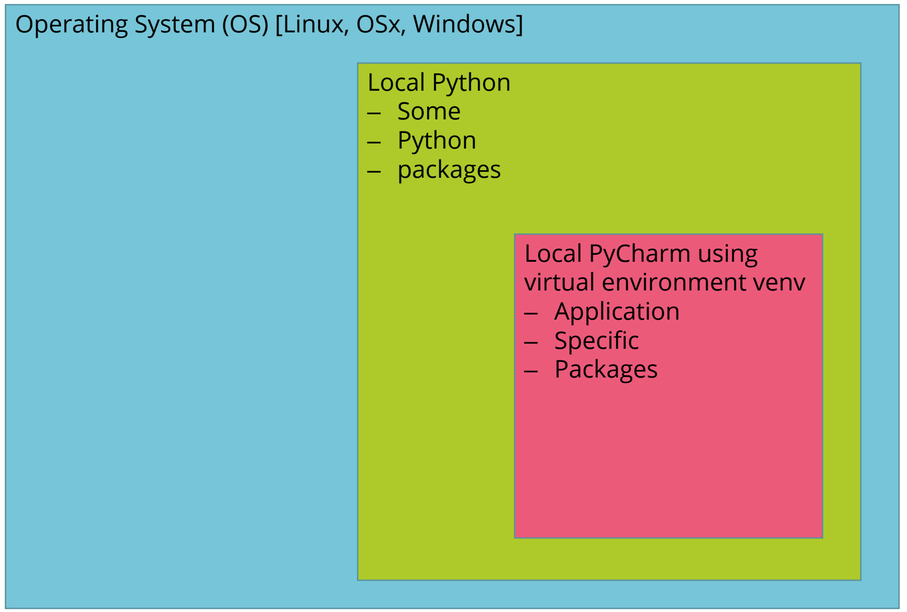


Virtual development environment: venv

- Python comes with some built-in packages and functions e.g. `print(), abs(), str(), open()`
- Typically, applications do not completely rely on built in functions but refer to external packages.
- External packages can be installed by the Package Installer for Python (pip).
- For avoiding dependency issues and keeping development environment clean, developer program applications in virtual environments (venv).

Create virtual environment using Python:

```
Python3 –m pip venv venvName
```



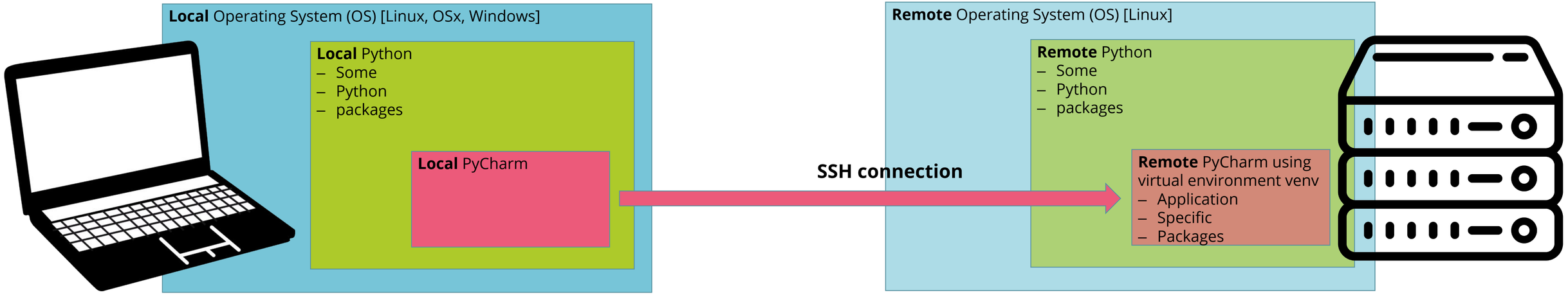
Pycharm for remote application development

Data scientists often deal with large data sets and computationally heavy machine learning algorithms.

Consequently:

- Data sets are stored on remote server.
- Python scripts have large execution times (days, weeks, month).
- Expensive computer resources (GPU, RAM, SSD) are shared in virtualized machines.
- Python scripts are developed on remote systems.

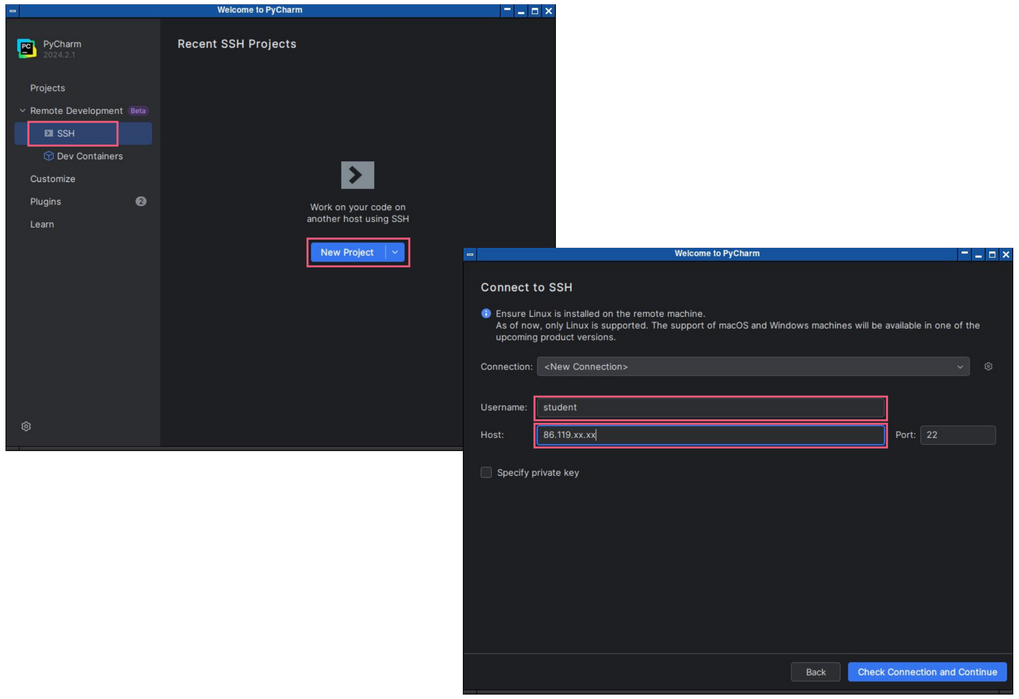
PyCharm allows for remote app development using secure shell (ssh) connection.



PyCharm for remote application development

Setup for remote application development:

1. Ensure the remote machine is up and running. For this course:
   - Go to spark.switch.ch
   - Login with HSLU switch account
   - Enter your personal server IPv4 (86.119.xx.xx) and password (e.g. bd19a4)
   - Start your machine
2. Start PyCharm
   - Close open project (Menu > File > Close Project)
   - Select SSH in Remote Development (left pane)
   - Click on “New Project”
   - Select connection (or create a new one)
     > enter username: **student**
     > enter host: **86.119.xx.xx**
   - Click on “Check Connection and Continue”
3. Select Python interpreter in: `/home/student/venv`



Wrap up setup process

- PyCharm Professional: installed
- Frist python project: created with dedicated virtual environment
- Virtual linux machine: started and connected to
- Remote development: project set up within PyCharm

-> Now we can start to python!

HSLU Lucerne University of Applied Sciences and Arts

Python Basics

Information Technology

February 20, 2025

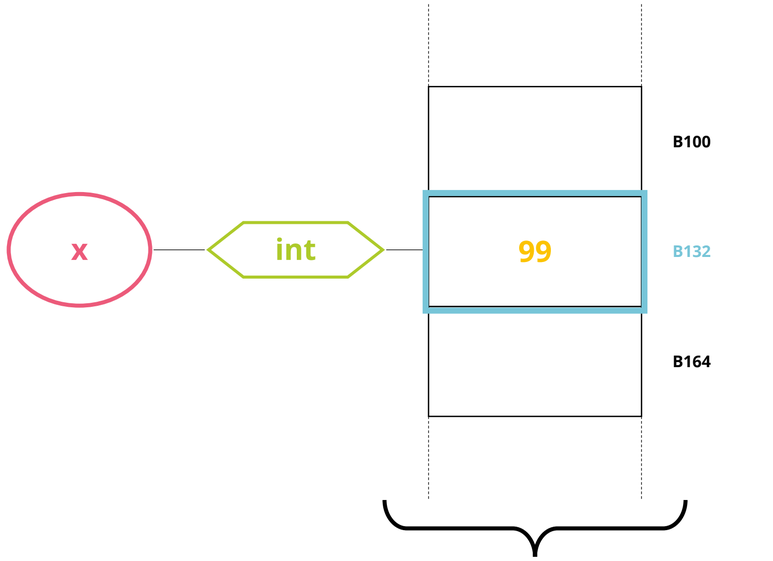
Variables and basic data types

Variables are names referring to a particular or undefined value.

- In math we know them from equations such as:
  $18 = 2 * x$ > x represents a particular value or
  $a^2 = b^2 + c^2$ > a, b and c can be arbitrary values holding the equation.
- In programming languages we use variables as **reference** to a particular storage location.

In Python, variables always consist of:

- **Name**
- **Data type**
- **Storage location**
- **Value**



... some **memory locations** from address **B100**

Variables and basic data types

Python does **not** require any **type** for variable definitions. It assumes the type from the value.

Python	C, C# or Java
`x = 99  (name = value)`	`int x = 99  (type name = value)`

Variable names…

- **have to** start with a letter [a-z,A-Z] (or in special cases with an underscore “_”).
- are case sensitive.
- start with lower cases (convention).
- with multiple words are separated with an underscore “hello_world”.
  -> Camel case (e.g. helloWorld) is not really used (allowed).

Variables and basic data types

Basic data types known in Python:

- Integer (int) `1, 2, 3, ... 100, ... 87293`
- Floating point numbers (float) `3.1415962535898`
- Strings (str) `“hello ‘my’ world” or ‘hello “my” world’`
- Boolean (bool) `True, False`
- Complex numbers `5 + 3j`

List types

- List `[1, “hello”, 3]` `->` `simple collection of elements, different types possible`
- Tuple `(1, “hello”, 3)` `->` `difference to list: tuples are immutable`
- Dictionary `{“a”:1, “b”:”hello”, “c”:3}` `-> key value pairs`
- Set `{1, “hello”, 3}` `->` `collection of unique elements`

Mathematical operations

Python has the following built-in mathematical operations.

Operator	Definition	Example
x + y	sum	9 + 4 = 13
x - y	subtraction	9 - 4 = 5
x * y	multiplication	9 * 4 = 36
x / y	division	9 / 4 = 2.25
x // y	integer division	9 // 4 = 2
x % y	reminder (or modulo)	9 % 4 = 1
x ** y	power	9 ** 4 = 6561

Mathematical operators

Two particular mathematical operators can have different meanings depending on the context:

1. Sum (**+**):
   Except for numerical values, the sum operation means appending an element to another
   -> appending usually requires two elements of same data types!
   - `‘hello’ + ‘ ‘ + ‘world’:` `‘hello world’`
   - `[1,2,3] + [4,5]:` `[1,2,3,4,5]`
2. Multiplication (*****):
   This operation **always** has to be applied with an **integer** (number).
   - `‘hello|’ * 3:` `‘hello|hello|hello|’`
   - `[1,2,3] * 3:` `[1,2,3,1,2,3,1,2,3]`

Finally, the two operations can be combined to:

- `‘hello|’ * 3 + ‘hello’:` `‘hello|hello|hello|hello’`

Boolean operations

Two boolean values can be combined differently:

`and`

	True	False
True	True	False
False	False	False

```
True and False == False
```

`or`

	True	False
True	True	True
False	True	False

```
True or False == True
```

`xor (^)`

	True	False
True	False	True
False	True	False

```
True ^ False == False
```

Any boolean operation can be inverted with the keyword: **not**

```
True and not False == True
```

```
not True or False == False
```

```
not(True ^ False) == False
```

Course alignment

- Experiment with python scripts, consoles and basic math operations
  -> setup a new project
  -> set the interpreter to the venv
  -> write your python instructions with basic math operations, variable assignments and print commands
- Course book: data types, variables, comments
- Online course (tutorials.eu):
  - Section 2: Python basics
  - Section 3: Python basics part 2
  - Section 4: Control structures

HSLU Lucerne University of Applied Sciences and Arts

School of Computer Science and Information Technology
Research
Ramón Christen
Research Associate Doctoral Student

Phone direct +41 41 757 68 96
ramon.christen@hslu.ch

HSLU T&A, Competence Center Thermal Energy Storage
Research
Andreas Melillo
Lecturer

Phone direct +41 41 349 35 91
andreas.melillo@hslu.ch