



合肥大學
HEFEI UNIVERSITY



Programming with Python

5. Programme Erstellen und Ausführen

Thomas Weise (汤卫思)

tweise@hfuu.edu.cn

Institute of Applied Optimization (IAO)
School of Artificial Intelligence and Big Data
Hefei University
Hefei, Anhui, China

应用优化研究所
人工智能与大数据学院
合肥大学
中国安徽省合肥市

Programming with Python



Dies ist ein Kurs über das Programmieren mit der Programmiersprache Python an der Universität Hefei (合肥大学).

Die Webseite mit dem Lehrmaterial dieses Kurses ist <https://thomasweise.github.io/programmingWithPython> (siehe auch den QR-Kode unten rechts). Dort können Sie das Kursbuch (in Englisch) und diese Slides finden. Das Repository mit den Beispielprogrammen in Python finden Sie unter <https://github.com/thomasWeise/programmingWithPythonCode>.



Outline



1. Einleitung
2. Das Erste PyCharm Projekt und Programm
3. Program im Terminal ausführen
4. Programm in Python Interpreter in PyCharm Eingeben
5. Programm in Python Interpreter in Terminal Eingeben
6. Zusammenfassung





Einleitung



Einleitung

- Jetzt haben wir PyCharm und Python installiert.



Einleitung

- Jetzt haben wir PyCharm und Python installiert.
- Nun wollen wir unser erstes Python-Programm schreiben und ausführen.



Einleitung



- Jetzt haben wir PyCharm und Python installiert.
- Nun wollen wir unser erstes Python-Programm schreiben und ausführen.
- Das Programm soll einfach "Hello World!" zum standard output stream (stdout) schreiben.

Einleitung



- Jetzt haben wir PyCharm und Python installiert.
- Nun wollen wir unser erstes Python-Programm schreiben und ausführen.
- Das Programm soll einfach "Hello World!" zum standard output stream (stdout) schreiben.
- Es besteht daher nur aus dem Statement `print("Hello World!")`.

Einleitung



- Jetzt haben wir PyCharm und Python installiert.
- Nun wollen wir unser erstes Python-Programm schreiben und ausführen.
- Das Programm soll einfach "Hello World!" zum standard output stream (stdout) schreiben.
- Es besteht daher nur aus dem Statement `print("Hello World!")`.

```
1 print("Hello World!")
```

↓ `python3 very_first_program.py` ↓

```
1 Hello World!
```

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen.

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen
 1. Wir können das Programm in der PyCharm Integrated Development Environment (IDE) in eine Python-Datei schreiben und in PyCharm ausführen.

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen
 1. Wir können das Programm in der PyCharm Integrated Development Environment (IDE) in eine Python-Datei schreiben und in PyCharm ausführen.
 2. Wir können das Programm auch in einem normalen Text-Editor schreiben.

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen
 1. Wir können das Programm in der PyCharm Integrated Development Environment (IDE) in eine Python-Datei schreiben und in PyCharm ausführen.
 2. Wir können das Programm auch in einem normalen Text-Editor schreiben.
Python-Programme sind ja im Grunde normale Text-Dateien.

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen
 1. Wir können das Programm in der PyCharm Integrated Development Environment (IDE) in eine Python-Datei schreiben und in PyCharm ausführen.
 2. Wir können das Programm auch in einem normalen Text-Editor schreiben. Python-Programme sind ja im Grunde normale Text-Dateien. Dann können wir das Programm mit dem Befehl `python3 programName` im Terminal ausführen.

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen
 1. Wir können das Programm in der PyCharm Integrated Development Environment (IDE) in eine Python-Datei schreiben und in PyCharm ausführen.
 2. Wir können das Programm auch in einem normalen Text-Editor schreiben. Python-Programme sind ja im Grunde normale Text-Dateien. Dann können wir das Programm mit dem Befehl `python3 programName` im Terminal ausführen.
 3. Wir können auch die Python-Interpreter Konsole in PyCharm öffnen und das Programm Zeile-für-Zeile eintippen und ausführen.

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen
 1. Wir können das Programm in der PyCharm Integrated Development Environment (IDE) in eine Python-Datei schreiben und in PyCharm ausführen.
 2. Wir können das Programm auch in einem normalen Text-Editor schreiben. Python-Programme sind ja im Grunde normale Text-Dateien. Dann können wir das Programm mit dem Befehl `python3 programName` im Terminal ausführen.
 3. Wir können auch die Python-Interpreter Konsole in PyCharm öffnen und das Programm Zeile-für-Zeile eintippen und ausführen.
 4. Natürlich können wir genausogut den Python-Interpreter im normalen Terminal öffnen und die Befehle Zeile-für-Zeile dort eintippen.

Programme Ausführen



- Es gibt 4 grundsätzliche Methoden, Python-Programme auszuführen
 1. Wir können das Programm in der PyCharm Integrated Development Environment (IDE) in eine Python-Datei schreiben und in PyCharm ausführen.
 2. Wir können das Programm auch in einem normalen Text-Editor schreiben. Python-Programme sind ja im Grunde normale Text-Dateien. Dann können wir das Programm mit dem Befehl `python3 programName` im Terminal ausführen.
 3. Wir können auch die Python-Interpreter Konsole in PyCharm öffnen und das Programm Zeile-für-Zeile eintippen und ausführen.
 4. Natürlich können wir genausogut den Python-Interpreter im normalen Terminal öffnen und die Befehle Zeile-für-Zeile dort eintippen.
- Schauen wir uns diese Möglichkeiten einmal an.



Das Erste PyCharm Projekt und Programm



Das Erste PyCharm Projekt und Programm

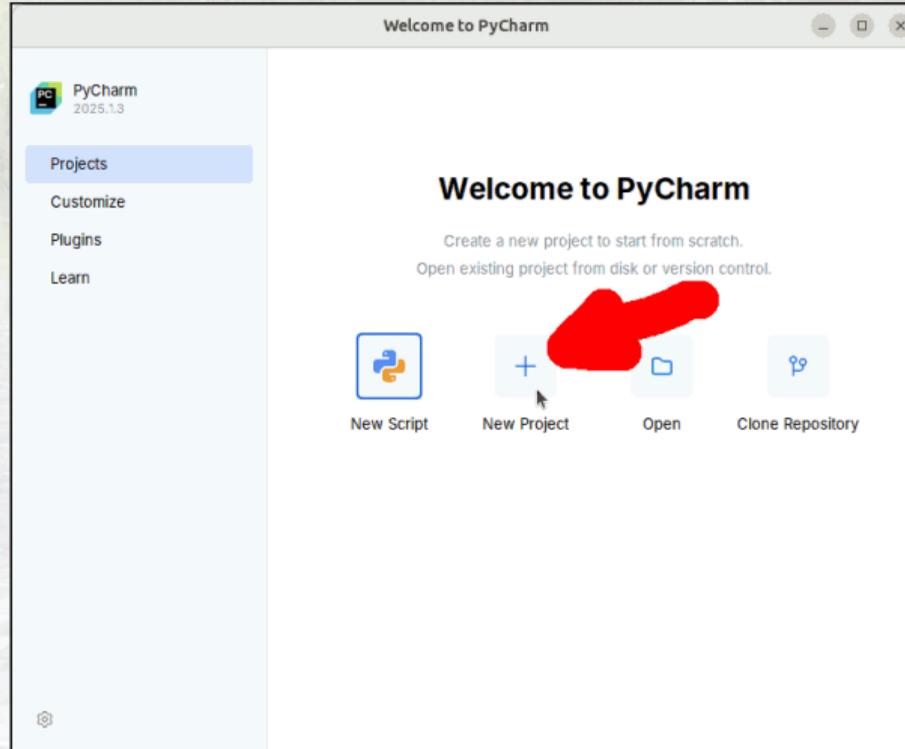
- OK, los geht's.



Das Erste PyCharm Projekt und Programm



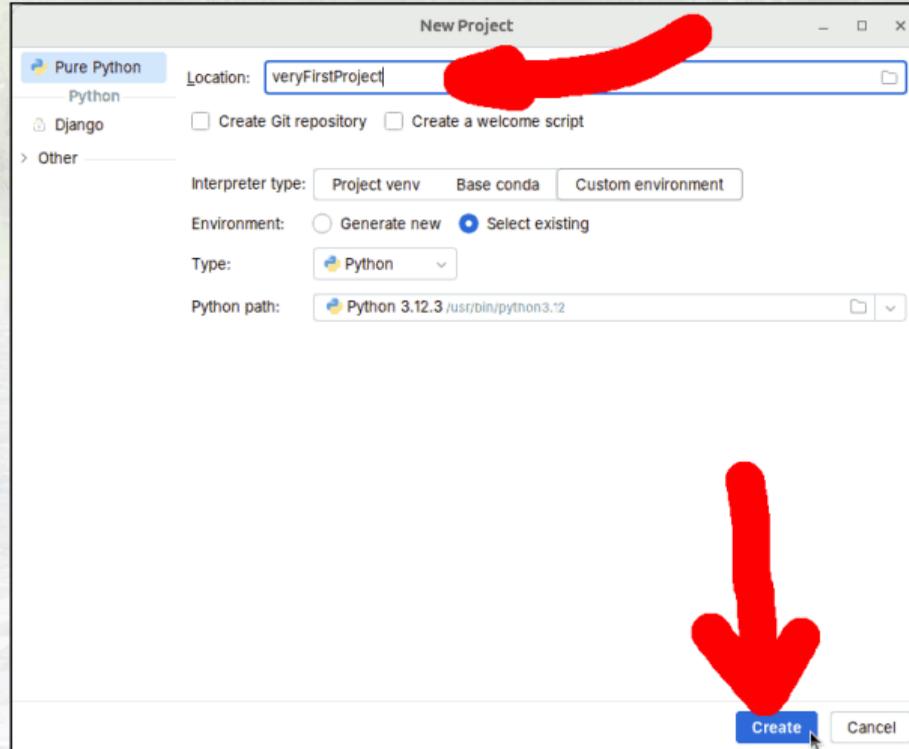
- Um ein neues Projekt in PyCharm zu erstellen, klicken wir auf **New Project** im Willkommensbildschirm.



Das Erste PyCharm Projekt und Programm



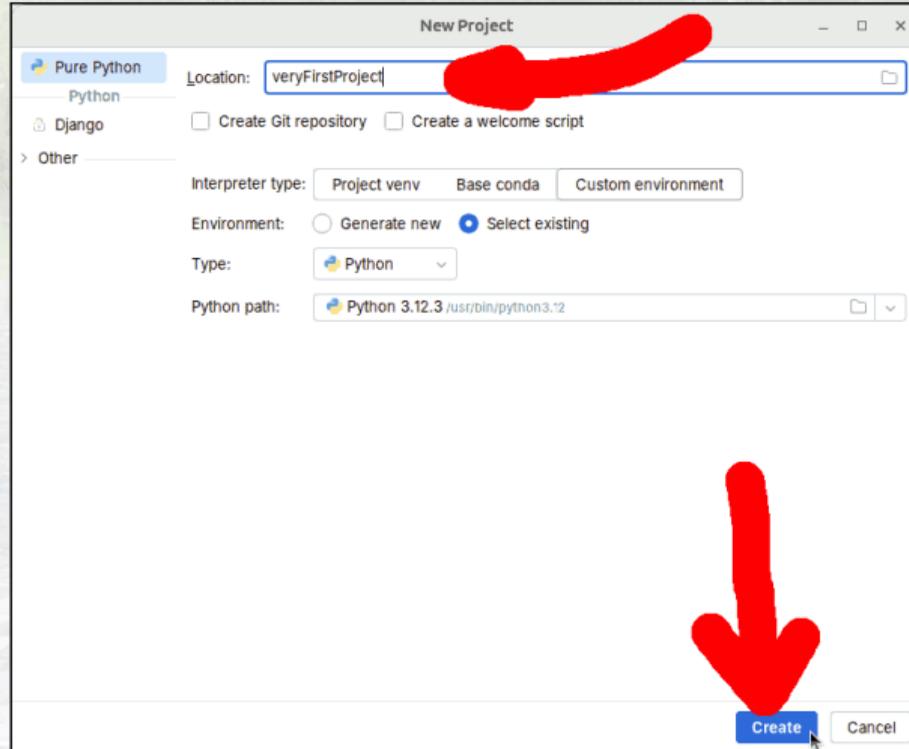
- Wir wählen links `Pure Python` aus und dann einen Namen für das Projekt.



Das Erste PyCharm Projekt und Programm



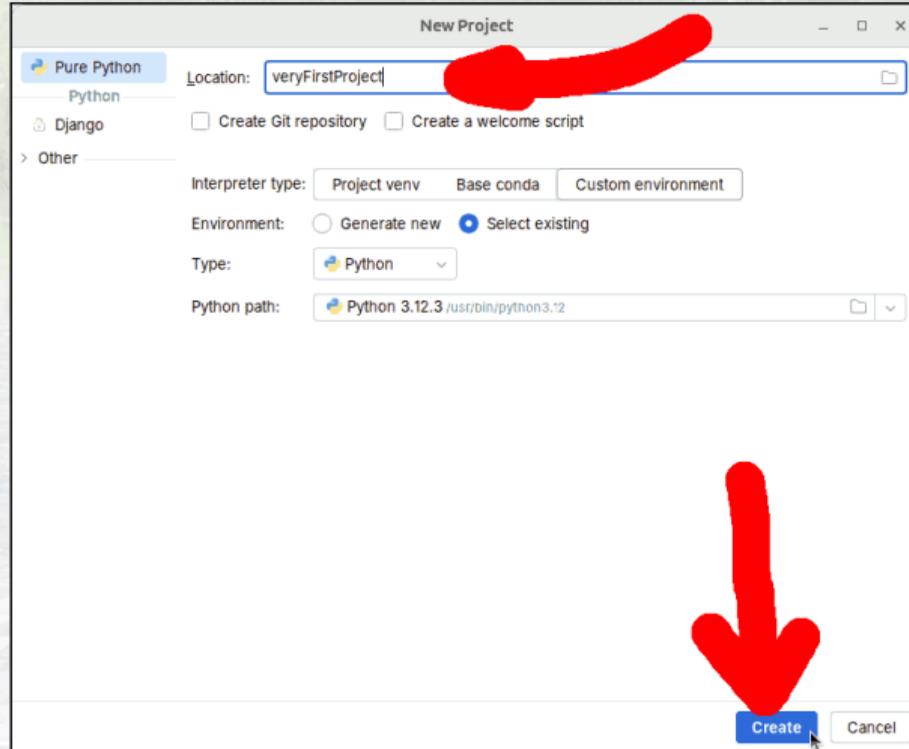
- Wir wählen links `Pure Python` aus und dann einen Namen für das Projekt.
- Wir wählen `veryFirstProject` als Name.



Das Erste PyCharm Projekt und Programm



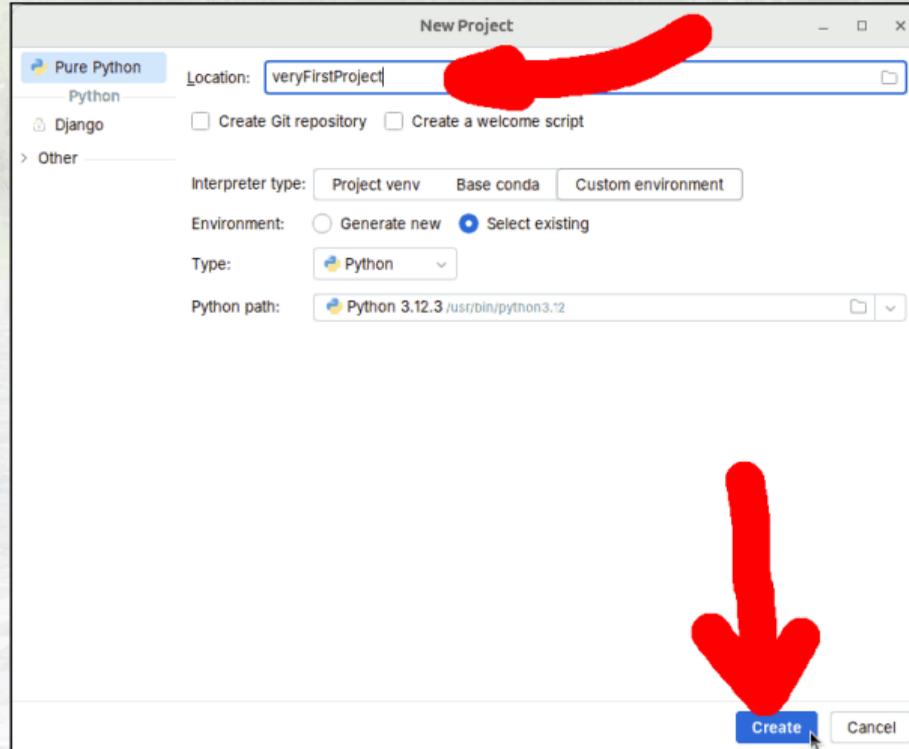
- Wir wählen veryFirstProject als Name.
- Wir wählen auch das Verzeichnis aus, in dem das Projekt gespeichert werden soll.



Das Erste PyCharm Projekt und Programm



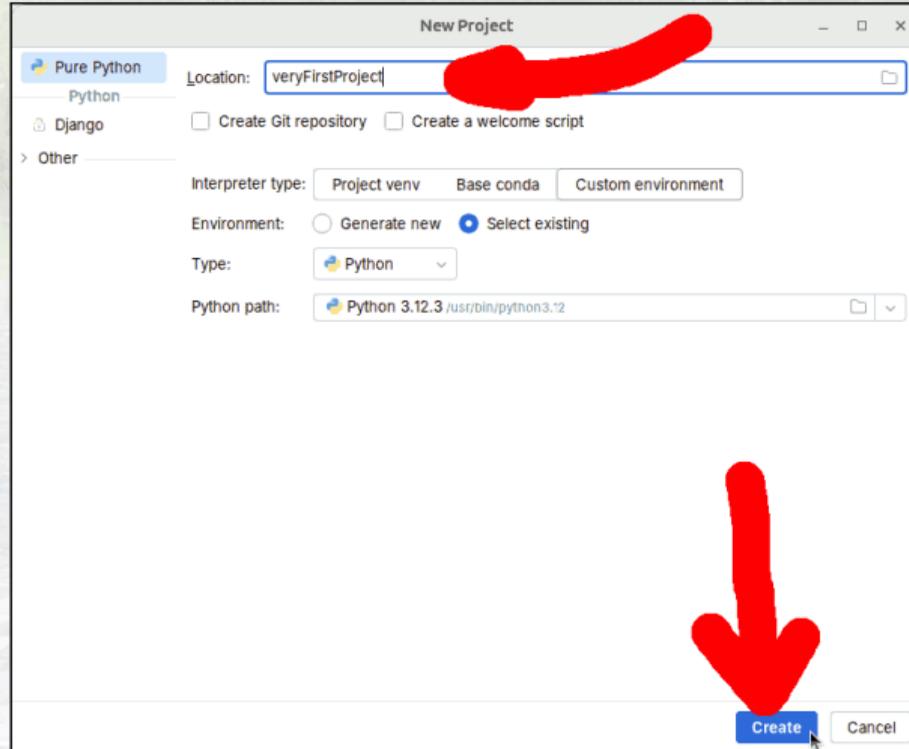
- Wir lassen die anderen Einstellungen auf den Standardwerten und/oder wählen unsere Python-Installation als **Custom Environment** aus.



Das Erste PyCharm Projekt und Programm



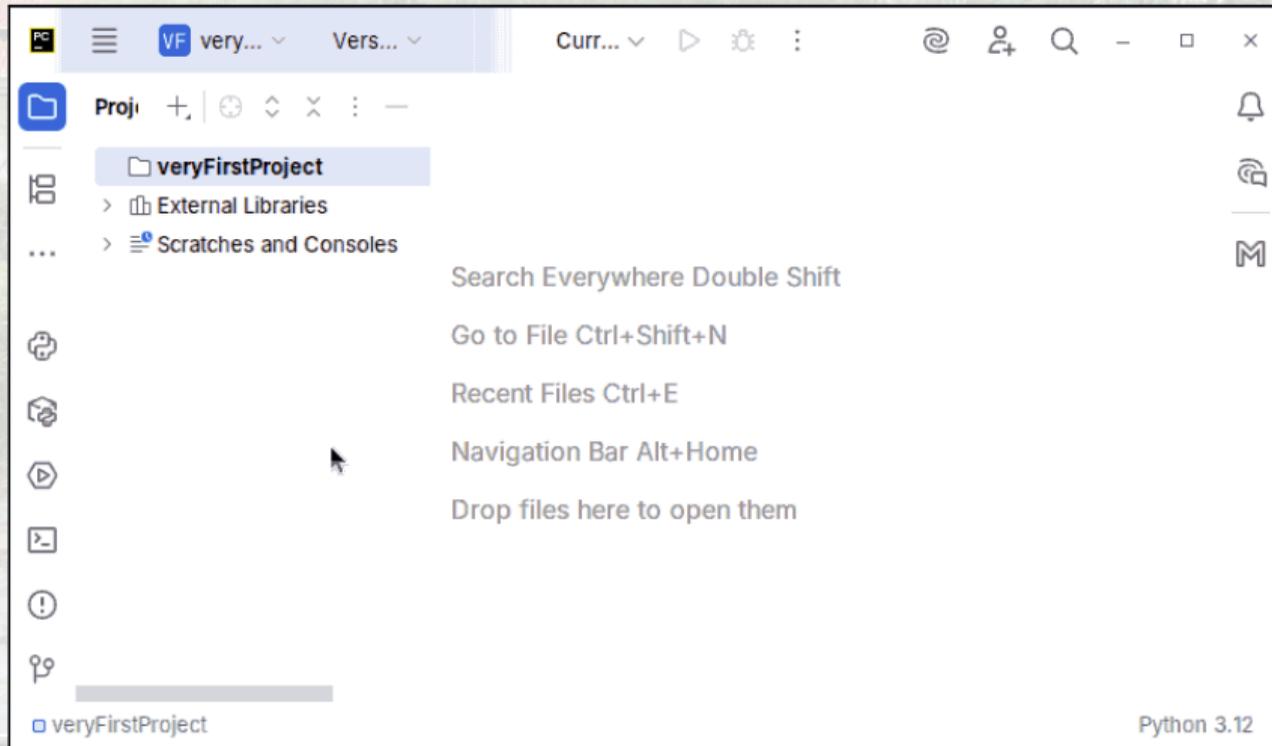
- Dann klicken wir auf `Create`.



Das Erste PyCharm Projekt und Programm

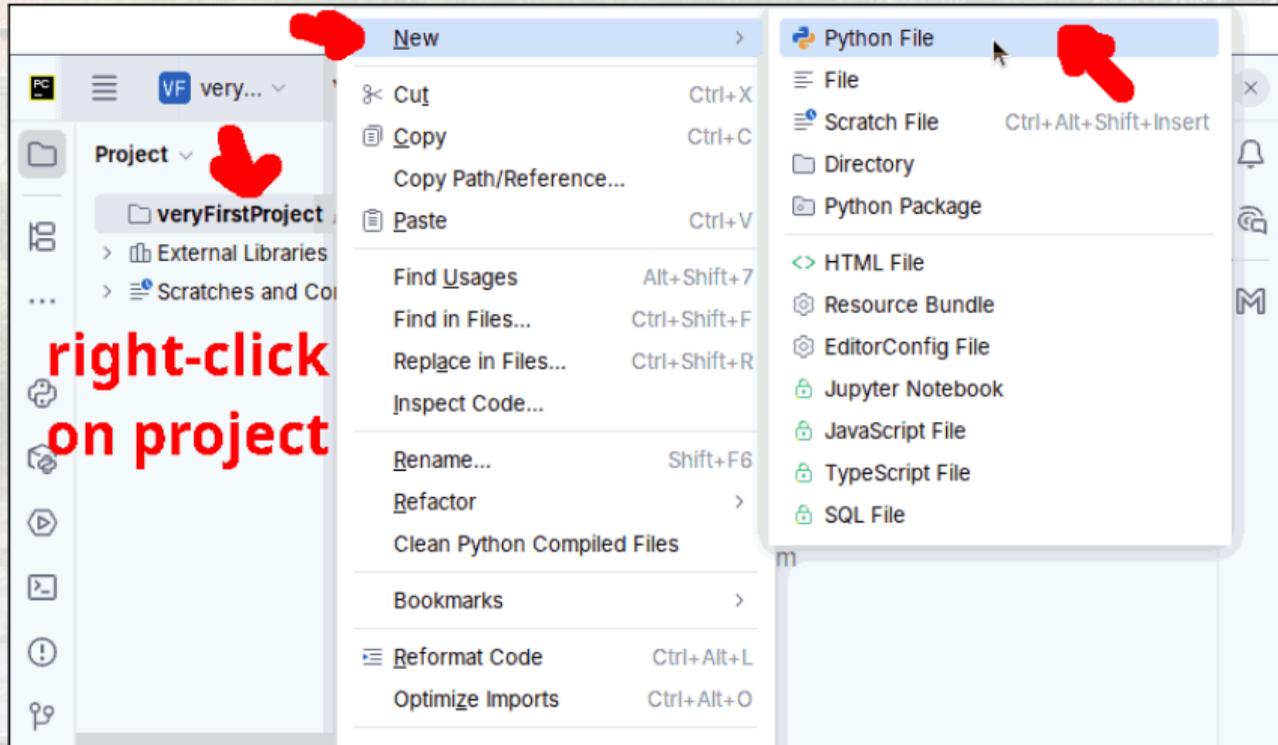


- Ein neues, leeres Projekt wurde erstellt.



Das Erste PyCharm Projekt und Programm

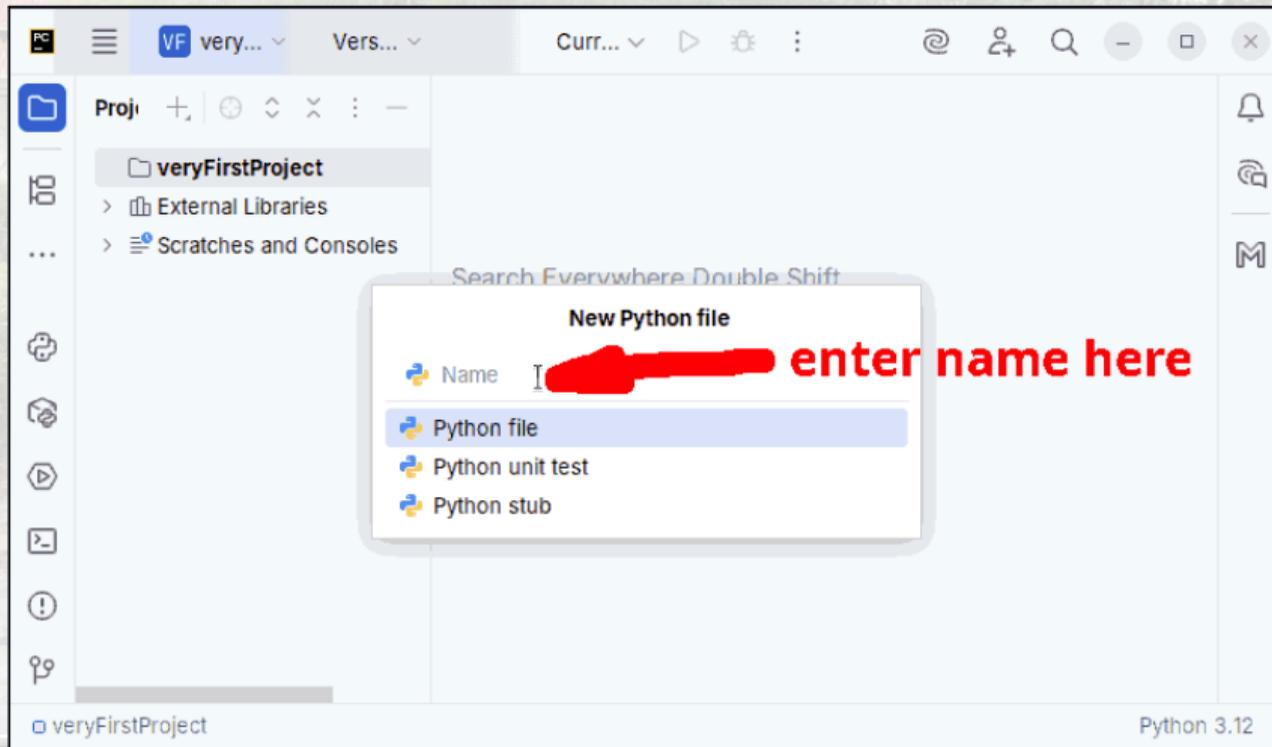
- Wir erstellen eine Python-Datei in diesem Projekt durch Rechtsklick auf den Projektorder `veryFirstProject` und dann durch auswählen von `New` `Python File`.



Das Erste PyCharm Projekt und Programm



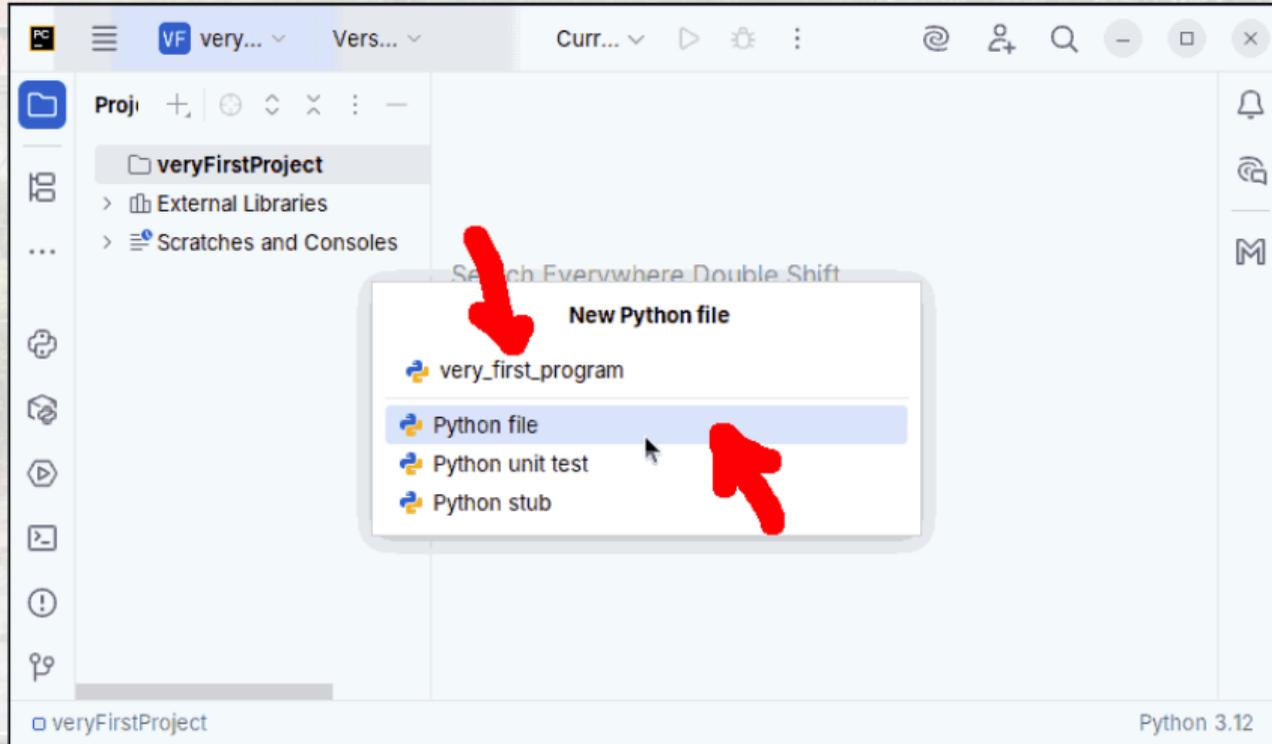
- In dem sich öffnenden Dialog können wir den Dateiname eingeben.



Das Erste PyCharm Projekt und Programm



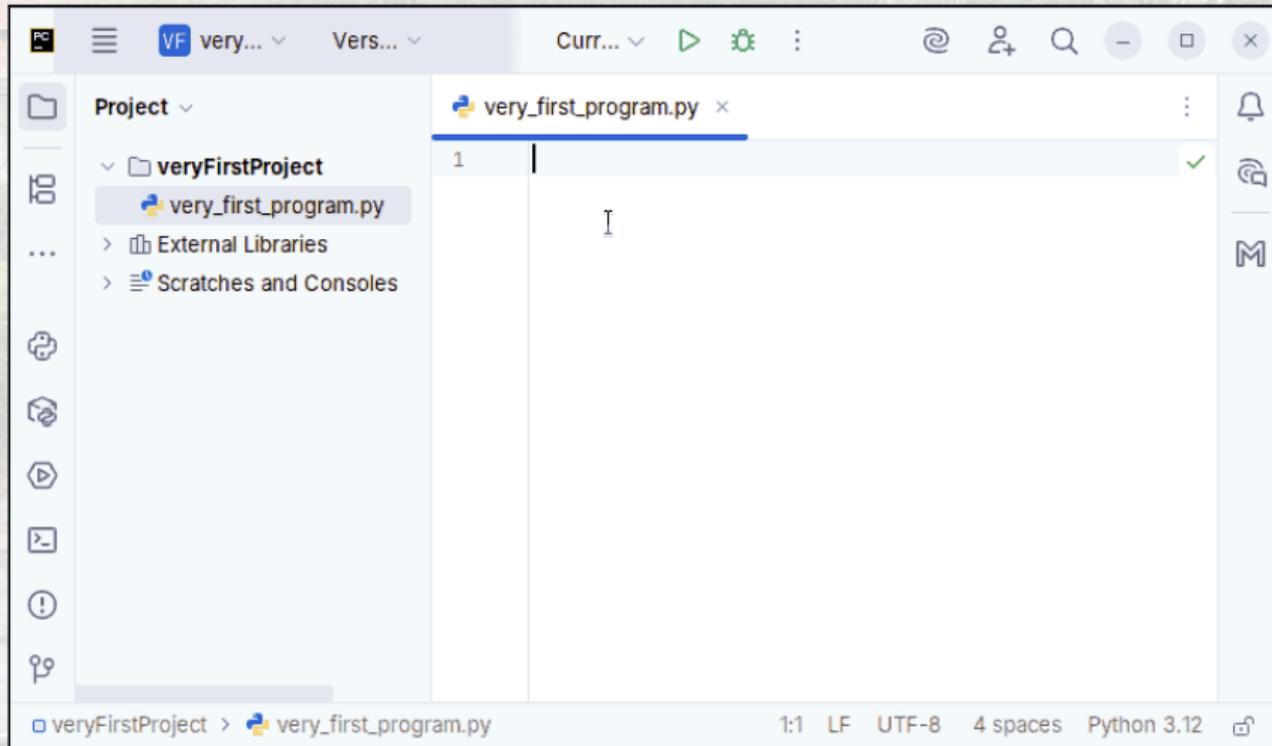
- Wir nennen unsere Datei `very_first_program` und drücken `Enter`.



Das Erste PyCharm Projekt und Programm



- Die neue, leere Datei `very_first_program.py` wurde im Projektordner `veryFirstProject` erstellt.



Das Erste PyCharm Projekt und Programm



- Nun tippen wir das Programm `print("Hello World!")` ab.

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and debugging. The left sidebar shows the project structure with a folder named 'veryFirstProject' containing the file 'very_first_program.py'. The main editor window displays the code for 'very_first_program.py' with the following content:

```
1 print("Hello World!")  
2 |
```

The status bar at the bottom indicates the current file is 'veryFirstProject > very_first_program.py' with settings for 2:1 LF, UTF-8 encoding, 4 spaces, and Python 3.12.

Das Erste PyCharm Projekt und Programm



- Nun tippen wir das Programm `print("Hello World!")` ab.
- PyCharm speichert die Datei automatisch für uns.

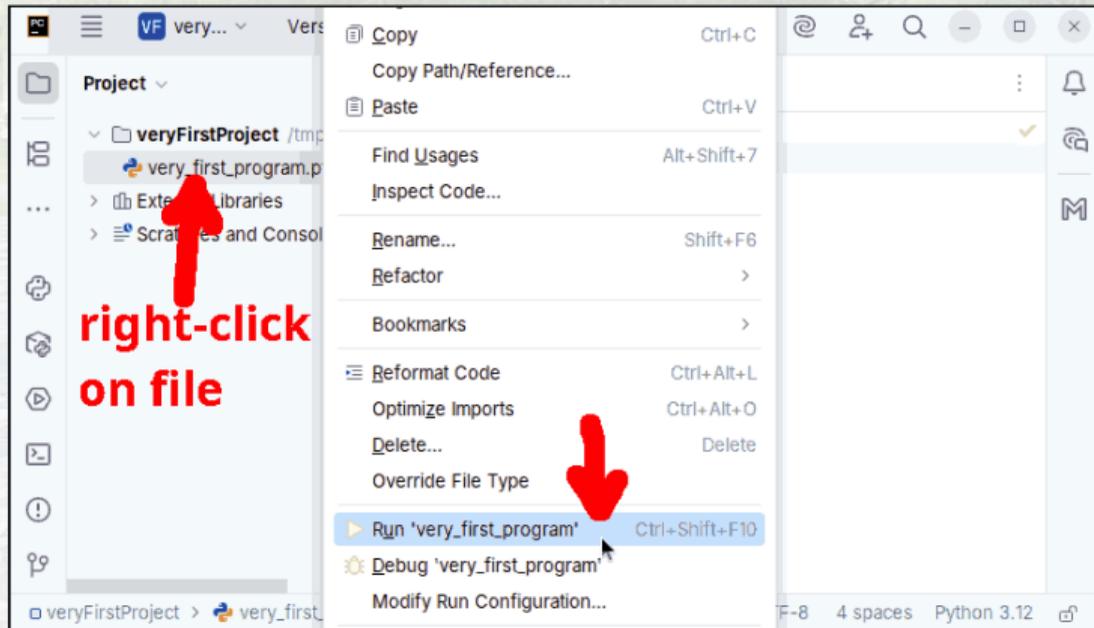
A screenshot of the PyCharm IDE interface. The top toolbar shows various icons for file operations and execution. The left sidebar displays the project structure with a folder named 'veryFirstProject' containing a file 'very_first_program.py'. The main editor window shows the code for 'very_first_program.py' with two lines: '1 print("Hello World!")' and '2 |'. A green checkmark is visible at the end of the first line. The bottom status bar indicates the current file path, encoding (UTF-8), indentation (4 spaces), and Python version (3.12).

```
PC  VF very...  Vers...  Curr...  ▶  ⚙  ⋮  @  👤  🔍  -  □  ×  
Project  ▾  very_first_program.py  ×  ⋮  🔔  
  ▾  veryFirstProject  
    very_first_program.py  
  >  External Libraries  
  >  Scratches and Consoles  
  
1  print("Hello World!")  ✓  
2  |  
  
veryFirstProject > very_first_program.py  2:1  LF  UTF-8  4 spaces  Python 3.12  🔗
```

Das Erste PyCharm Projekt und Programm



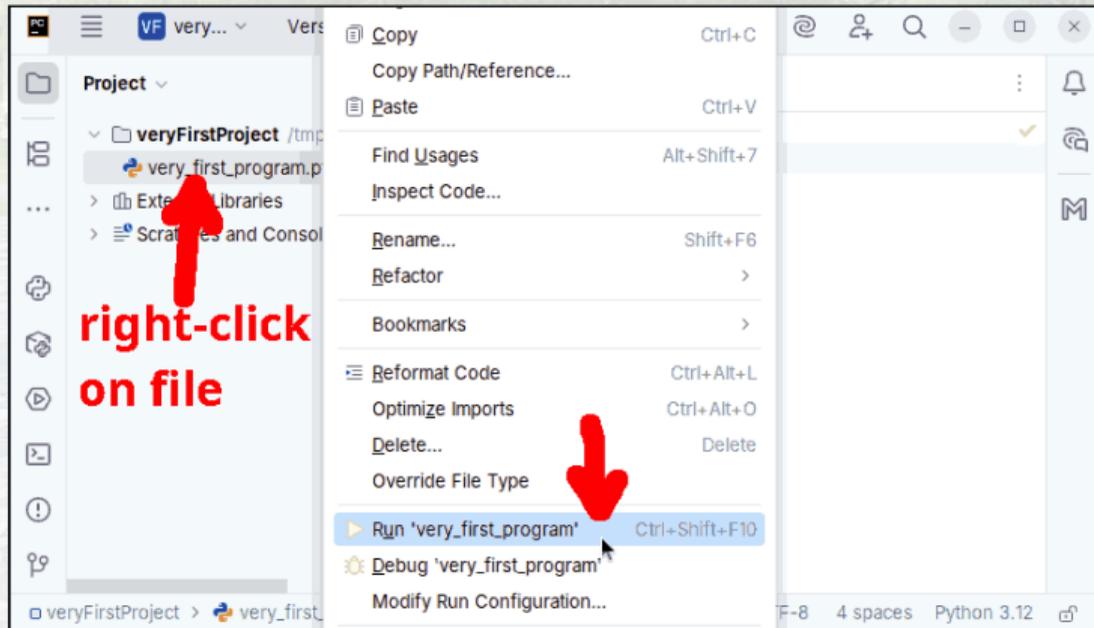
- Um das Programm auszuführen, rechtsklicken wir auf die Programmdatei und wählen `Run 'very_first_program'` us.



Das Erste PyCharm Projekt und Programm

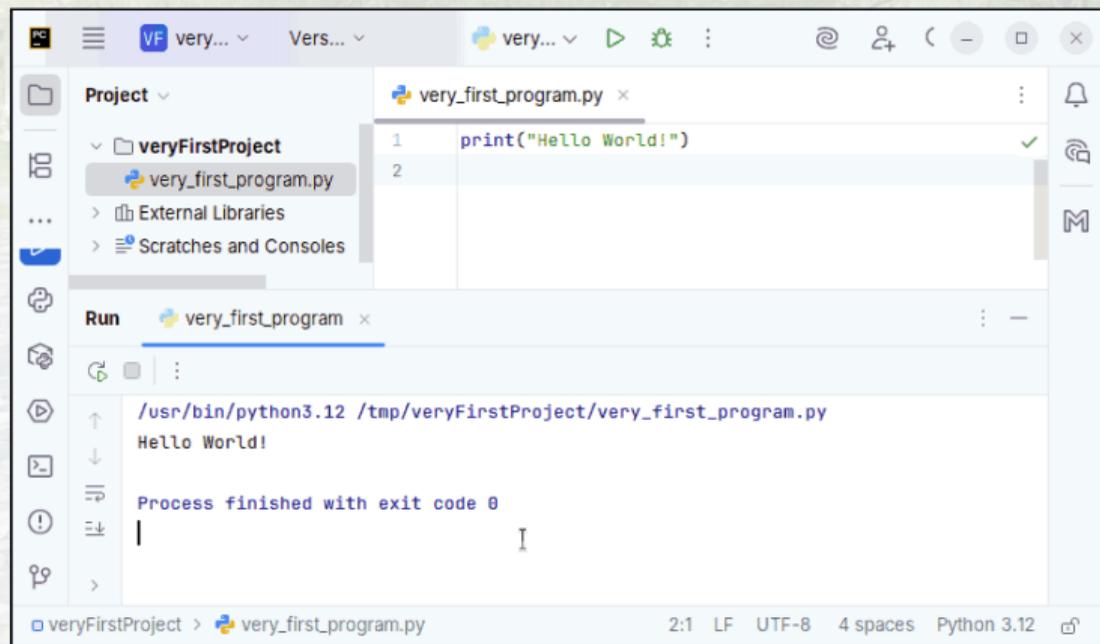


- Um das Programm auszuführen, rechtsklicken wir auf die Programmdatei und wählen `Run 'very_first_program'` us.
- Alternativ könnten wir auch `Ctrl` + `↑` + `F10` drücken.



Das Erste PyCharm Projekt und Programm

- Tatsächlich: In der Konsolenfläche im PyCharm-Fenster erscheint der Text "Hello World!".



The screenshot displays the PyCharm IDE interface. The top toolbar shows the 'Run' button (a green play icon) and other standard window controls. The left sidebar contains the 'Project' view, showing a folder named 'veryFirstProject' with a file 'very_first_program.py' selected. The main editor window shows the code for 'very_first_program.py' with two lines: '1 print("Hello World!")' and '2'. The 'Run' console at the bottom shows the execution path: '/usr/bin/python3.12 /tmp/veryFirstProject/very_first_program.py', followed by the output 'Hello World!' and the message 'Process finished with exit code 0'. The status bar at the bottom indicates the current file is 'veryFirstProject > very_first_program.py' with settings for '2:1 LF UTF-8 4 spaces Python 3.12'.

```
very...  very...  very...  ▶  ⚙  ⋮  @  👤  ◀  -  □  ×
```

Project ▾

- veryFirstProject
 - very_first_program.py
 - External Libraries
 - Scratches and Consoles

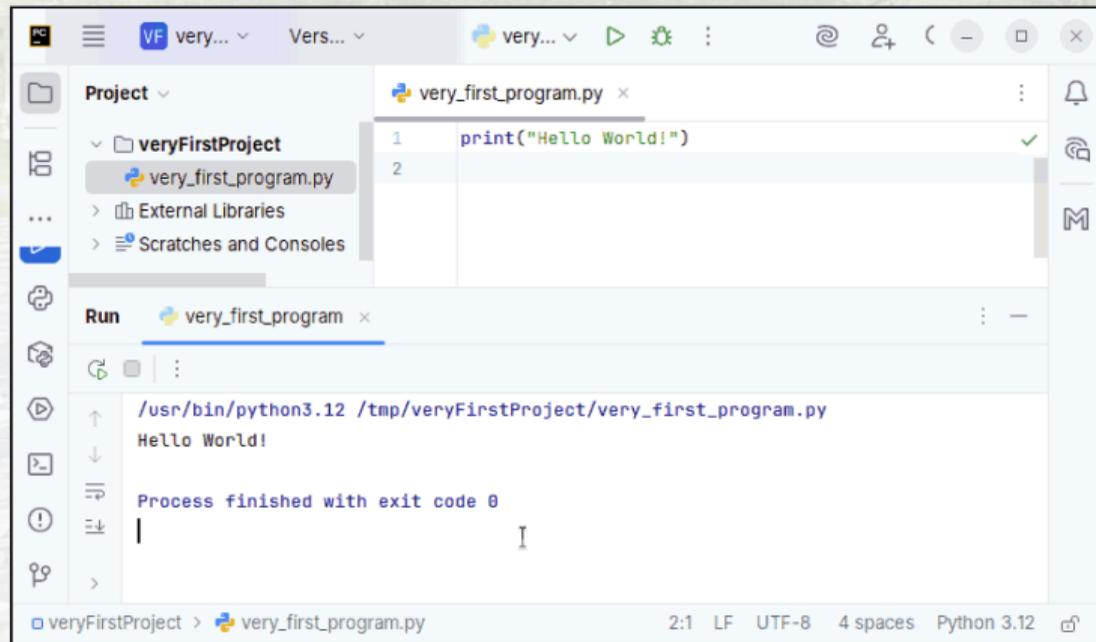
Run very_first_program ×

```
/usr/bin/python3.12 /tmp/veryFirstProject/very_first_program.py  
Hello World!  
  
Process finished with exit code 0
```

veryFirstProject > very_first_program.py 2:1 LF UTF-8 4 spaces Python 3.12

Das Erste PyCharm Projekt und Programm

- Tatsächlich: In der Konsolenfläche im PyCharm-Fenster erscheint der Text "Hello World!".
- Zusätzlich sehen wir auch, wie das Programm ausgeführt wurde, nämlich den Python-Interpreter mit dem Pfad zu unserer Datei als Parameter.



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, a search icon, and a run button (a green play icon). The left sidebar shows a project tree with a folder named 'veryFirstProject' containing a file 'very_first_program.py'. The main editor window displays the code for 'very_first_program.py' with two lines: '1 print("Hello World!")' and '2'. A green checkmark is visible at the end of the first line. Below the editor is the 'Run' console, which shows the command executed: '/usr/bin/python3.12 /tmp/veryFirstProject/very_first_program.py'. The output of the command is 'Hello World!'. Below the output, it states 'Process finished with exit code 0'. The bottom status bar indicates the current file is 'veryFirstProject > very_first_program.py' with settings for '2:1 LF UTF-8 4 spaces Python 3.12'.

```
very...  very...  very...  ▶  ⚙  ⋮  @  👤  ◀  -  □  ×  
Project  very_first_program.py  ×  
  1  print("Hello World!")  ✓  
  2  
Run  very_first_program  ×  
  ↻  |  ⋮  
  ↑  /usr/bin/python3.12 /tmp/veryFirstProject/very_first_program.py  
  ↓  Hello World!  
  ≡  Process finished with exit code 0  
  ⚠  |  
  >  
veryFirstProject > very_first_program.py  2:1 LF UTF-8 4 spaces Python 3.12  🗑
```

Das Erste PyCharm Projekt und Programm



- Wir bekommen auch "Process finished with exit code 0" angezeigt: Unser Programm ist erfolgreich und ohne Fehler abgelaufen.

The screenshot displays the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and debugging. The left sidebar shows the project structure with a folder named 'veryFirstProject' containing a file 'very_first_program.py'. The main editor window shows the code for 'very_first_program.py' with two lines: '1 print("Hello World!")' and '2'. A green checkmark is visible at the end of the first line. Below the editor, the 'Run' console shows the execution command: '/usr/bin/python3.12 /tmp/veryFirstProject/very_first_program.py'. The output of the program is 'Hello World!'. At the bottom of the console, the message 'Process finished with exit code 0' is displayed, indicating a successful execution. The status bar at the bottom shows the current file path, line and column numbers (2:1), encoding (LF), character set (UTF-8), indentation (4 spaces), and the Python version (Python 3.12).

```
1 print("Hello World!")
2
```

Run very_first_program x

/usr/bin/python3.12 /tmp/veryFirstProject/very_first_program.py
Hello World!
Process finished with exit code 0

veryFirstProject > very_first_program.py 2:1 LF UTF-8 4 spaces Python 3.12



Program im Terminal ausführen



Program im Terminal ausführen

- Führen wir nun das selbe Program im normalen Terminal aus.



Program im Terminal ausführen



- Wir öffnen ein Terminal.

A terminal window with a dark background and light text. The title bar shows 'tweise@weise-laptop: ~'. The command prompt is 'tweise@weise-laptop:~\$' and the command entered is 'cd /tmp/veryFirstProject/'.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/
```

Program im Terminal ausführen



- Wir öffnen ein Terminal. (Unter Ubuntu Linux durch Drücken von `Ctrl` + `Alt` + `T`, unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`.)

```
tweise@weise-laptop: ~
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/
```

Program im Terminal ausführen



- Wir öffnen ein Terminal. (Unter Ubuntu Linux durch Drücken von `Ctrl` + `Alt` + `T`, unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`.)
- Wir wechseln in das Projektverzeichnis, wo sich die auszuführende Python-Datei befindet.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/
```

Program im Terminal ausführen



- Wir öffnen ein Terminal. (Unter Ubuntu Linux durch Drücken von `Ctrl` + `Alt` + `T`, unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`.)
- Wir wechseln in das Projektverzeichnis, wo sich die auszuführende Python-Datei befindet.
- Das Kommando dafür ist `cd directory` gefolgt von `Enter`.

```
tweise@weise-laptop: ~
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/
```

Program im Terminal ausführen



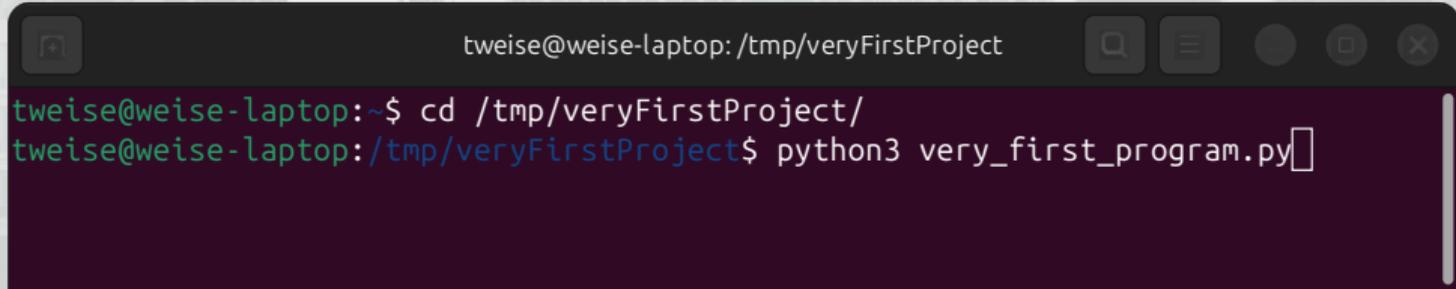
- Wir sind nun in dem Projektverzeichnis.

```
tweise@weise-laptop: /tmp/veryFirstProject  
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/  
tweise@weise-laptop:/tmp/veryFirstProject$
```

Program im Terminal ausführen



- Wir führen ein Programm “program.py” mit dem Befehl `python3 program.py` (gefolgt von `Enter`) aus.

A terminal window with a dark purple background. The title bar shows the user 'tweise@weise-laptop' and the current directory '/tmp/veryFirstProject'. The terminal contains two lines of text: the first line shows the user changing the directory to '/tmp/veryFirstProject/' using the 'cd' command, and the second line shows the user running 'python3 very_first_program.py' with a cursor at the end of the command.

```
tweise@weise-laptop: /tmp/veryFirstProject  
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/  
tweise@weise-laptop:/tmp/veryFirstProject$ python3 very_first_program.py
```

Program im Terminal ausführen



- Wir führen ein Programm “program.py” mit dem Befehl `python3 program.py` (gefolgt von `Enter`) aus. In unserem Fall ist der Dateiname “very_first_program.py”.

```
tweise@weise-laptop: /tmp/veryFirstProject
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/
tweise@weise-laptop:/tmp/veryFirstProject$ python3 very_first_program.py
```

Program im Terminal ausführen



- Wie erwartet wird erscheint “Hello World!” im Terminal.

```
tweise@weise-laptop: /tmp/veryFirstProject  
tweise@weise-laptop:~$ cd /tmp/veryFirstProject/  
tweise@weise-laptop:/tmp/veryFirstProject$ python3 very_first_program.py  
Hello World!  
tweise@weise-laptop:/tmp/veryFirstProject$
```

Program im Terminal ausführen



- Führen wir nun das selbe Program im normalen Terminal aus.

Gute Praxis

Die einzig **richtige** Art, Python Programme im Produktiveinsatz auszuführen, ist sie im Terminal mit dem Python Interpreter als Programmdatei zu starten.

Program im Terminal ausführen



- Führen wir nun das selbe Program im normalen Terminal aus.

Gute Praxis

Die einzig **richtige** Art, Python Programme im Produktiveinsatz auszuführen, ist sie im Terminal mit dem Python Interpreter als Programmdatei zu starten.

- Alle anderen Arten sind vielleicht während der Entwicklung nützlich, haben aber nichts im Produktiveinsatz verloren.

Program im Terminal ausführen



- Führen wir nun das selbe Program im normalen Terminal aus.

Gute Praxis

Die einzig **richtige** Art, Python Programme im Produktiveinsatz auszuführen, ist sie im Terminal mit dem Python Interpreter als Programmdatei zu starten.

- Alle anderen Arten sind vielleicht während der Entwicklung nützlich, haben aber nichts im Produktiveinsatz verloren.
- Das gilt ganz besonders für das Ausführen mit Hilfe von PyCharm.

Program im Terminal ausführen



- Führen wir nun das selbe Program im normalen Terminal aus.

Gute Praxis

Die einzig **richtige** Art, Python Programme im Produktiveinsatz auszuführen, ist sie im Terminal mit dem Python Interpreter als Programmdatei zu starten.

- Alle anderen Arten sind vielleicht während der Entwicklung nützlich, haben aber nichts im Produktiveinsatz verloren.
- Das gilt ganz besonders für das Ausführen mit Hilfe von PyCharm. Machen Sie das niemals im Produktiveinsatz.



Programm in Python Interpreter in PyCharm Eingeben



Programm in Python Interpreter in PyCharm Eingeben

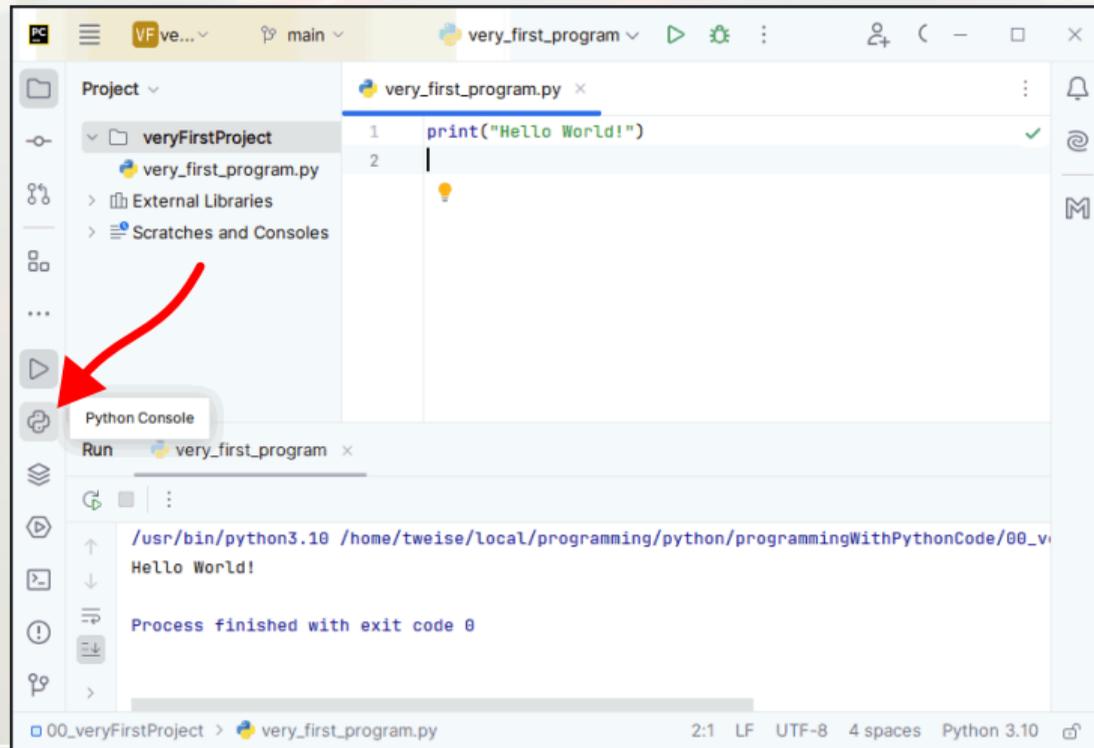


- Nun wollen wir ein Programm Schritt-für-Schritt in den Python-Interpreter in PyCharm eingeben und ausführen.

Programm in Python Interpreter in PyCharm Eingeben



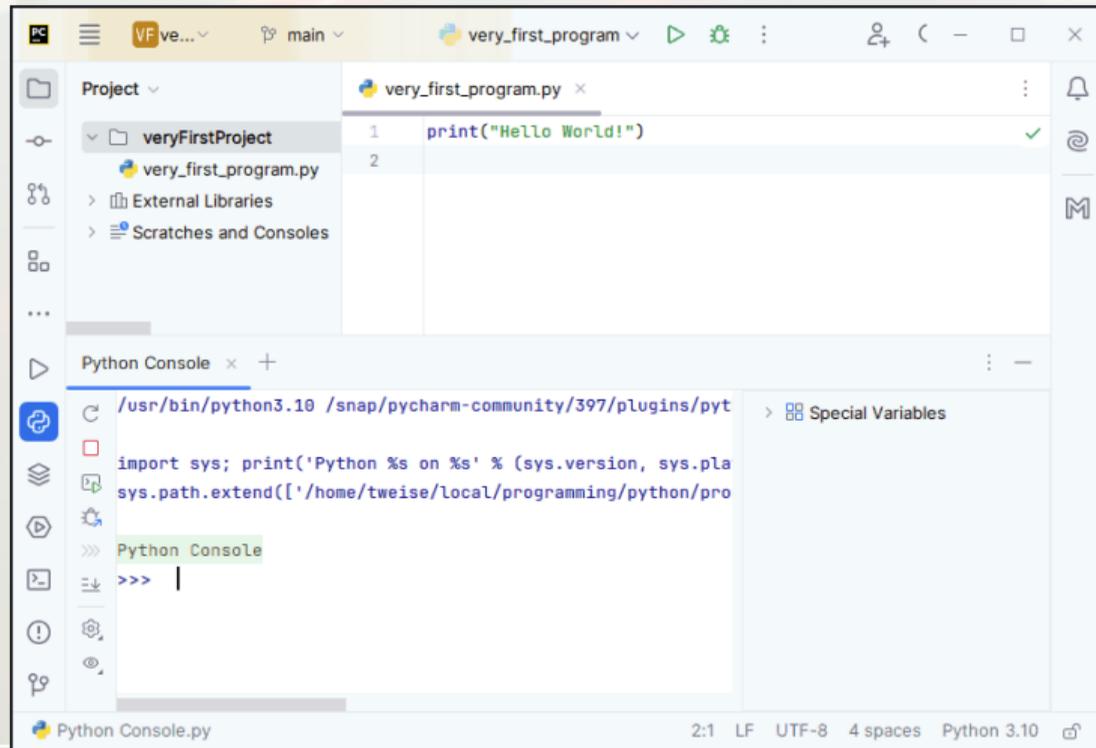
- Wir drücken den -Button auf der vertikalen Knopfliste auf der linken Seite des PyCharm-Fensters.



Programm in Python Interpreter in PyCharm Eingeben



- Die PyCharm Python-Interpreter-Konsole öffnet sich.



Programm in Python Interpreter in PyCharm Eingeben



- Wir tippen das “Hello World!”-Programm ein, i.e., `print("Hello World!")`, und drücken `Enter`.

The screenshot displays the PyCharm IDE interface. The top toolbar shows the 'Run' button (a green play icon). The main editor window shows a file named `very_first_program.py` with the following code:

```
1 print("Hello World!")
```

The Python Console at the bottom shows the execution of the code:

```
/usr/bin/python3.10 /snap/pycharm-community/397/plugins/pyt
import sys; print('Python %s on %s' % (sys.version, sys.pla
sys.path.extend(['/home/tweise/local/programming/python/pro
>>> Python Console
>>> print("Hello World!")
```

The status bar at the bottom indicates the current settings: `2:1 LF UTF-8 4 spaces Python 3.10`.

Programm in Python Interpreter in PyCharm Eingeben



- Die Ausgabe "Hello World!" erscheint.

The screenshot displays the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and debugging. The main editor window shows a file named `very_first_program.py` with the following code:

```
1 print("Hello World!")
```

The Python Console at the bottom shows the execution process:

```
/usr/bin/python3.10 /snap/pycharm-community/397/plugins/pyt  
import sys; print('Python %s on %s' % (sys.version, sys.pla  
sys.path.extend(['/home/tweise/local/programming/python/pro  
Python Console>>> print("Hello World!")  
Hello World!  
>>>
```

The status bar at the bottom indicates the current settings: `Python Console.py`, `2:1`, `LF`, `UTF-8`, `4 spaces`, and `Python 3.10`.



Programm in Python Interpreter in Terminal Eingeben



Programm in Python Interpreter in Terminal Eingeben



- Jetzt werden wir das Programm in den Python-Interpreter im Terminal eingeben.

Programm in Python Interpreter in Terminal Eingeben



- Wir öffnen ein Terminal.

A terminal window with a dark background. The title bar shows 'tweise@weise-laptop: ~'. The prompt 'tweise@weise-laptop:~\$' is followed by the command 'python3' and a cursor. The window has standard Linux window controls (minimize, maximize, close) and a search icon.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ python3
```

Programm in Python Interpreter in Terminal Eingeben



- Wir öffnen ein Terminal. (Unter Ubuntu Linux durch Drücken von `Ctrl` + `Alt` + `T`, unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`.)

```
tweise@weise-laptop: ~
tweise@weise-laptop:~$ python3
```

Programm in Python Interpreter in Terminal Eingeben



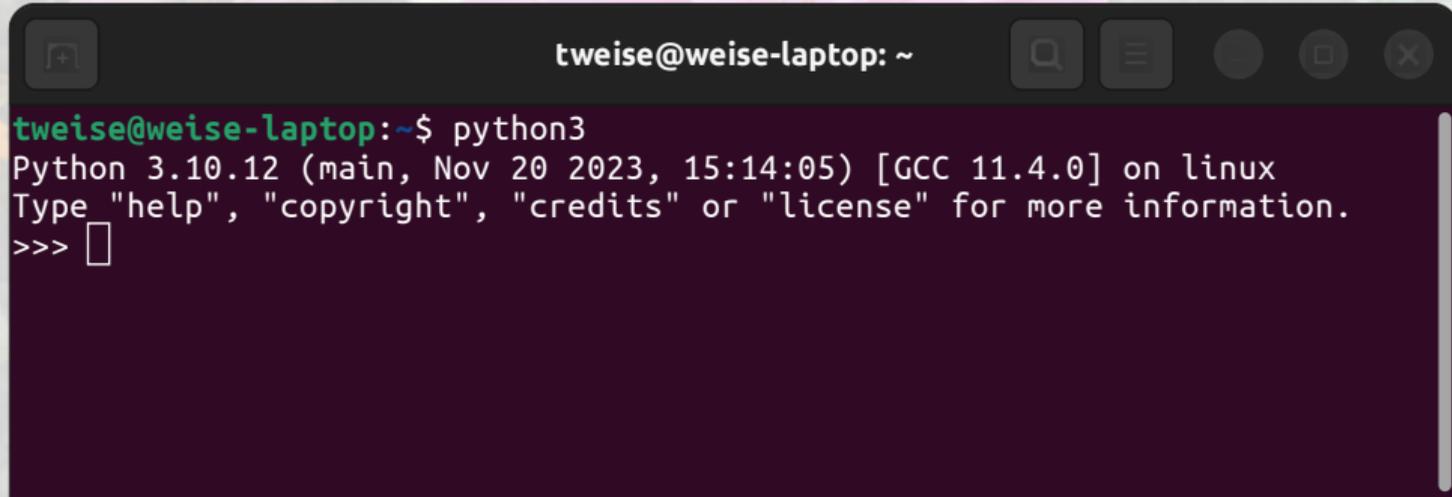
- Wir öffnen ein Terminal. (Unter Ubuntu Linux durch Drücken von `Ctrl` + `Alt` + `T`, unter Microsoft Windows durch Druck auf `Windows` + `R`, dann Schreiben von `cmd`, dann Druck auf `↵`.)
- Wir geben `python3` ein und drücken `Enter`.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ python3
```

Programm in Python Interpreter in Terminal Eingeben



- Die Python-Interpreter-Konsole öffnet sich im Terminal.

A terminal window with a dark background and light text. The window title is 'tweise@weise-laptop: ~'. The terminal output shows the command 'python3' being executed, resulting in the Python 3.10.12 startup message. The prompt '>>>' is followed by a cursor.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ python3  
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

Programm in Python Interpreter in Terminal Eingeben



- Wir tippen das "Hello World!"-Programm ein, i.e., `print("Hello World!")`, und drücken .

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ python3  
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello World!")
```

Programm in Python Interpreter in Terminal Eingeben



- Die Ausgabe “Hello World!” erscheint.

```
tweise@weise-laptop: ~  
twaise@weise-laptop:~$ python3  
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello World!")  
Hello World!  
>>> 
```

Programm in Python Interpreter in Terminal Eingeben



- Um den interaktiven Python-Interpreter wieder zu verlassen, tippen wir `exit()` ein und drücken `Enter`.

```
tweise@weise-laptop: ~  
twiese@weise-laptop:~$ python3  
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello World!")  
Hello World!  
>>> exit()
```

Programm in Python Interpreter in Terminal Eingeben



- Wir sind zurück im normalen Terminal.

```
tweise@weise-laptop: ~  
tweise@weise-laptop:~$ python3  
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello World!")  
Hello World!  
>>> exit()  
tweise@weise-laptop:~$
```



Zusammenfassung



Zusammenfassung



- Wir haben vier Arten kennengelernt, wie wir Python-Programme ausführen können.

Zusammenfassung



- Wir haben vier Arten kennengelernt, wie wir Python-Programme ausführen können.
- Auf der einen Seite können wir Programme als Textdateien mit der Endung `.py` speichern.

Zusammenfassung



- Wir haben vier Arten kennengelernt, wie wir Python-Programme ausführen können.
- Auf der einen Seite können wir Programme als Textdateien mit der Endung `.py` speichern. Diese können wir dann entweder im Terminal oder in PyCharm ausführen.

Zusammenfassung



- Wir haben vier Arten kennengelernt, wie wir Python-Programme ausführen können.
- Auf der einen Seite können wir Programme als Textdateien mit der Endung `.py` speichern. Diese können wir dann entweder im Terminal oder in PyCharm ausführen.
- Auf der anderen Seite können wir Programme auch Zeile-für-Zeile in einer interaktiven Python-Interpreter-Session direkt in den Interpreter eintippen.

Zusammenfassung



- Wir haben vier Arten kennengelernt, wie wir Python-Programme ausführen können.
- Auf der einen Seite können wir Programme als Textdateien mit der Endung `.py` speichern. Diese können wir dann entweder im Terminal oder in PyCharm ausführen.
- Auf der anderen Seite können wir Programme auch Zeile-für-Zeile in einer interaktiven Python-Interpreter-Session direkt in den Interpreter eintippen. Auch das können wir entweder im Terminal oder in PyCharm machen.
- Natürlich werden wir unsere Programme in “richtigen” Projekten immer in Dateien speichern.

Zusammenfassung



- Wir haben vier Arten kennengelernt, wie wir Python-Programme ausführen können.
- Auf der einen Seite können wir Programme als Textdateien mit der Endung `.py` speichern. Diese können wir dann entweder im Terminal oder in PyCharm ausführen.
- Auf der anderen Seite können wir Programme auch Zeile-für-Zeile in einer interaktiven Python-Interpreter-Session direkt in den Interpreter eintippen. Auch das können wir entweder im Terminal oder in PyCharm machen.
- Natürlich werden wir unsere Programme in “richtigen” Projekten immer in Dateien speichern.
- Aber zum Kennenlernen von Python ist eine interaktive Nutzung des Interpreters sehr geeignet¹.



谢谢您门！
Thank you!
Vielen Dank!



References I



- [1] "An Informal Introduction to Python". In: *Python 3 Documentation. The Python Tutorial*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/tutorial/introduction.html> (besucht am 2025-07-11) (siehe S. 69–74).