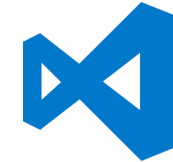


INSTALLATION ANACONDA UND VS CODE

IDE – Entwicklungsumgebung

Visual Studio Code: Erweiterungen



Über Erweiterung müssen zunächst diverse „Extension Packs“ installiert werden (falls noch nicht vorhanden):

Bitte installieren Sie dazu folgende Software Packages im Voraus auf das Notebook, diese sind beide Lizenzen frei:

- Visual Studio Code von Microsoft:
<https://code.visualstudio.com/download>
- Anaconda, beinhaltet Python Runtime und eine Python Notebook Oberfläche
<https://www.anaconda.com/download>

Oder

- Miniconda, ist eine kleine Version von Anaconda
<https://docs.conda.io/projects/miniconda/en/latest/index.html>

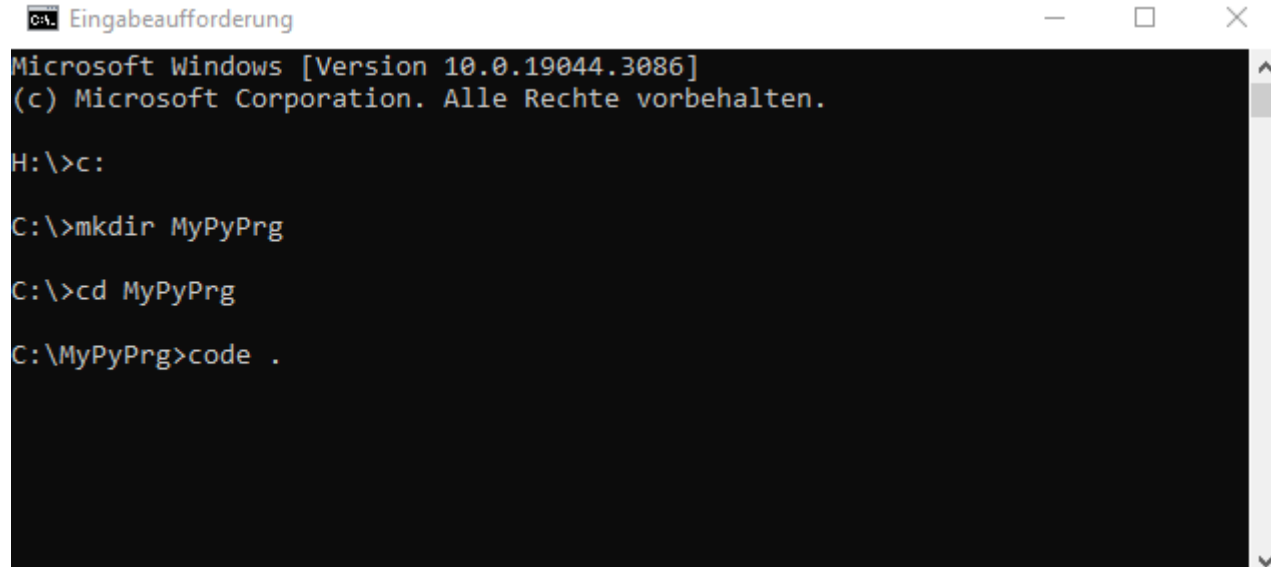
Achtung:
Anaconda oder Miniconda am besten in
ein höheres Directory installieren:
z.B. C:\temp\Miniconda

Anlegen eines Projekts

1. Öffnen Sie cmd

2. Gehe nach C:\

- ### 3. Erstellen ein dir
- C:\> mkdir myPyPrg
 - C:\> mkdir MyPyPrg
 - C:\> cd MyPyPrg
 - C:\MyPyPrg> code .



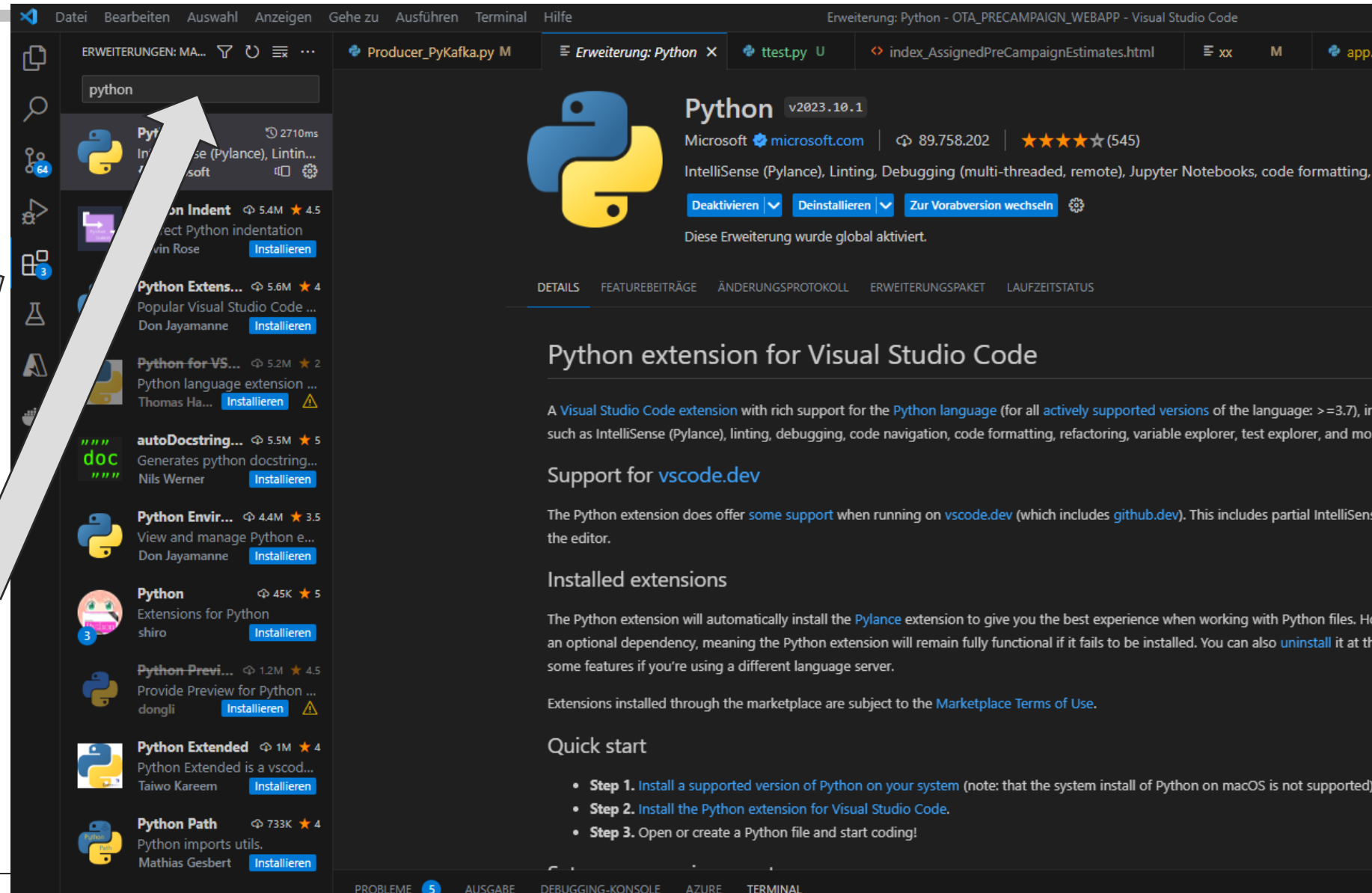
Code Setup

Install Python Extension

1. Selektiere 5 Icon

2. Filter Python

3. Installation von Python Extension

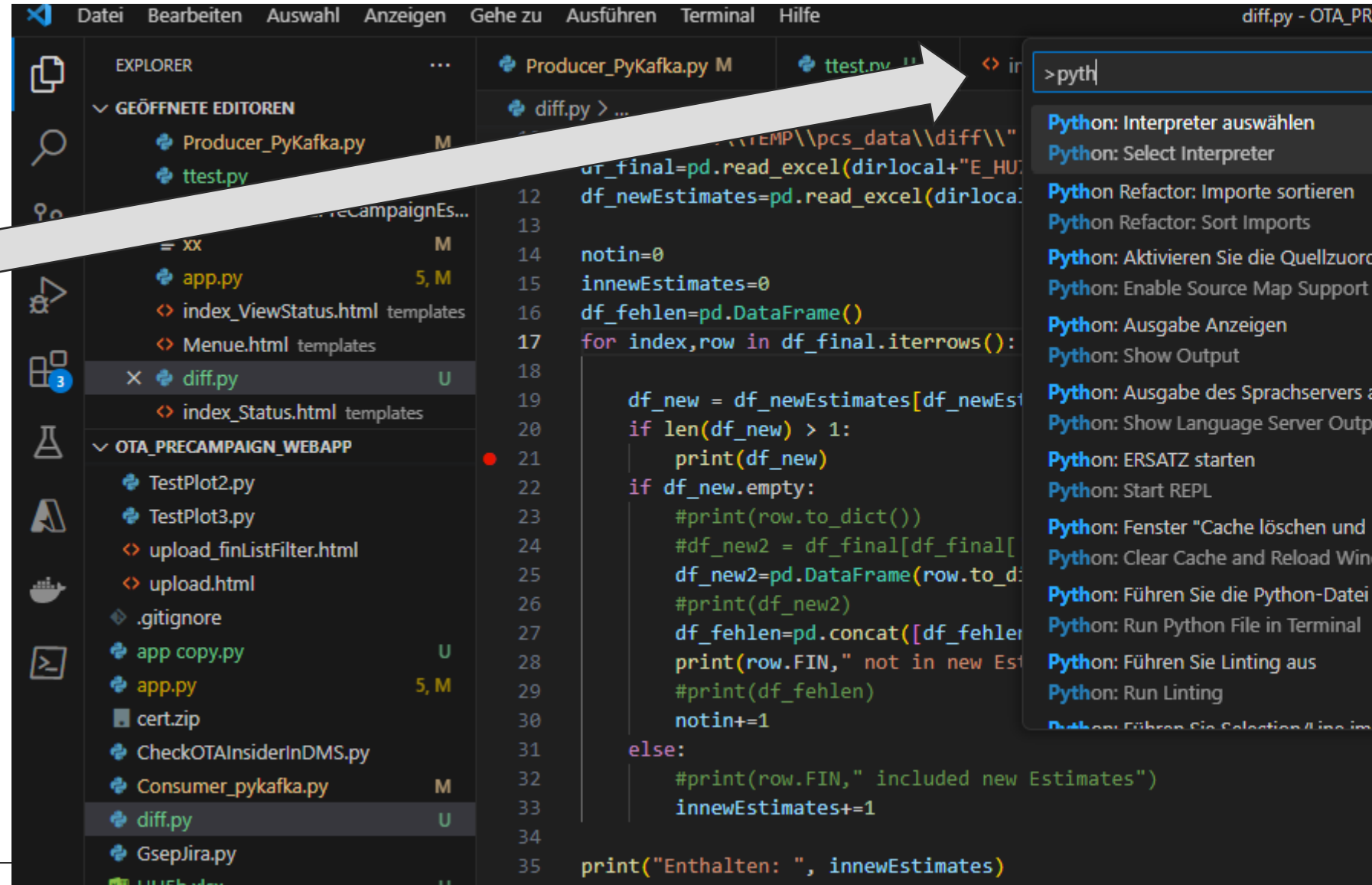


Code Setup

Select Python Interpreter

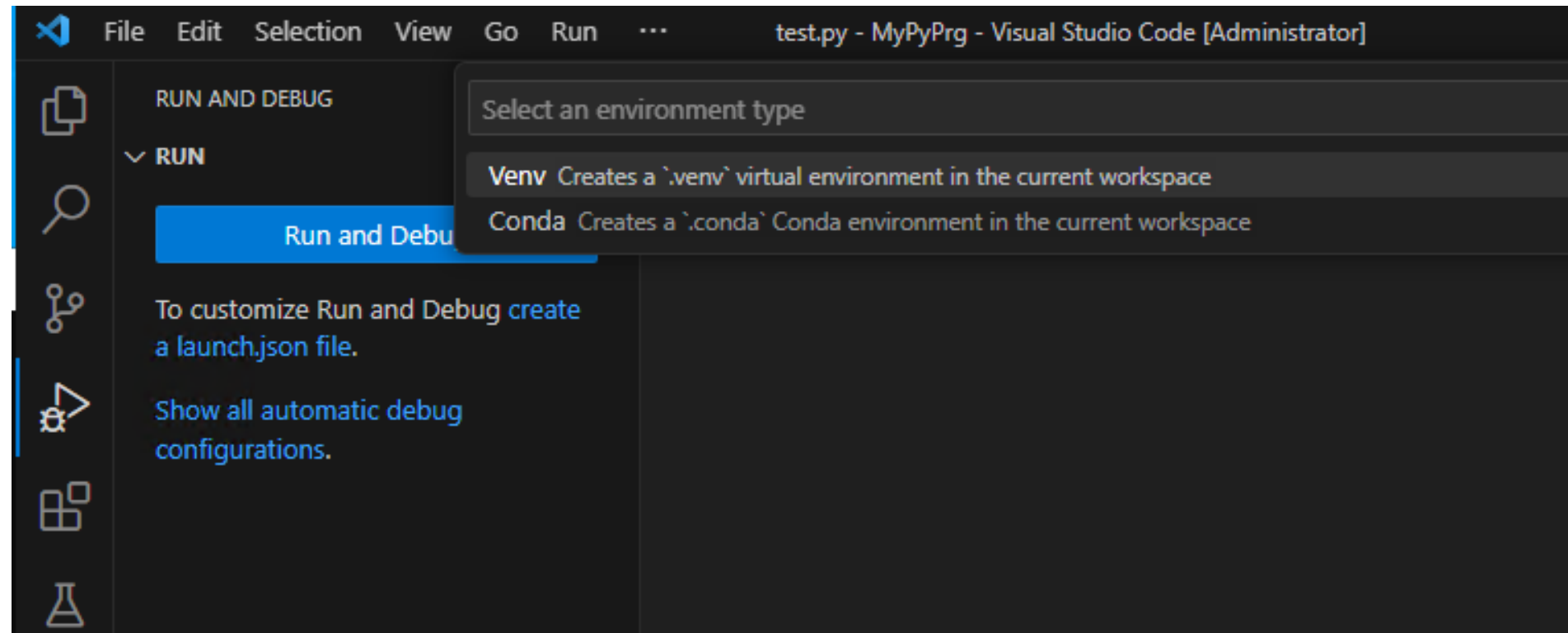
1. Press
Strg+Shift+P

2. Filter python



Create Venv

1. Press **Strg+Schift+P**
2. Create Virtuel Env
3. Select „Venv“
4. Check if python is selected with new Venv Path

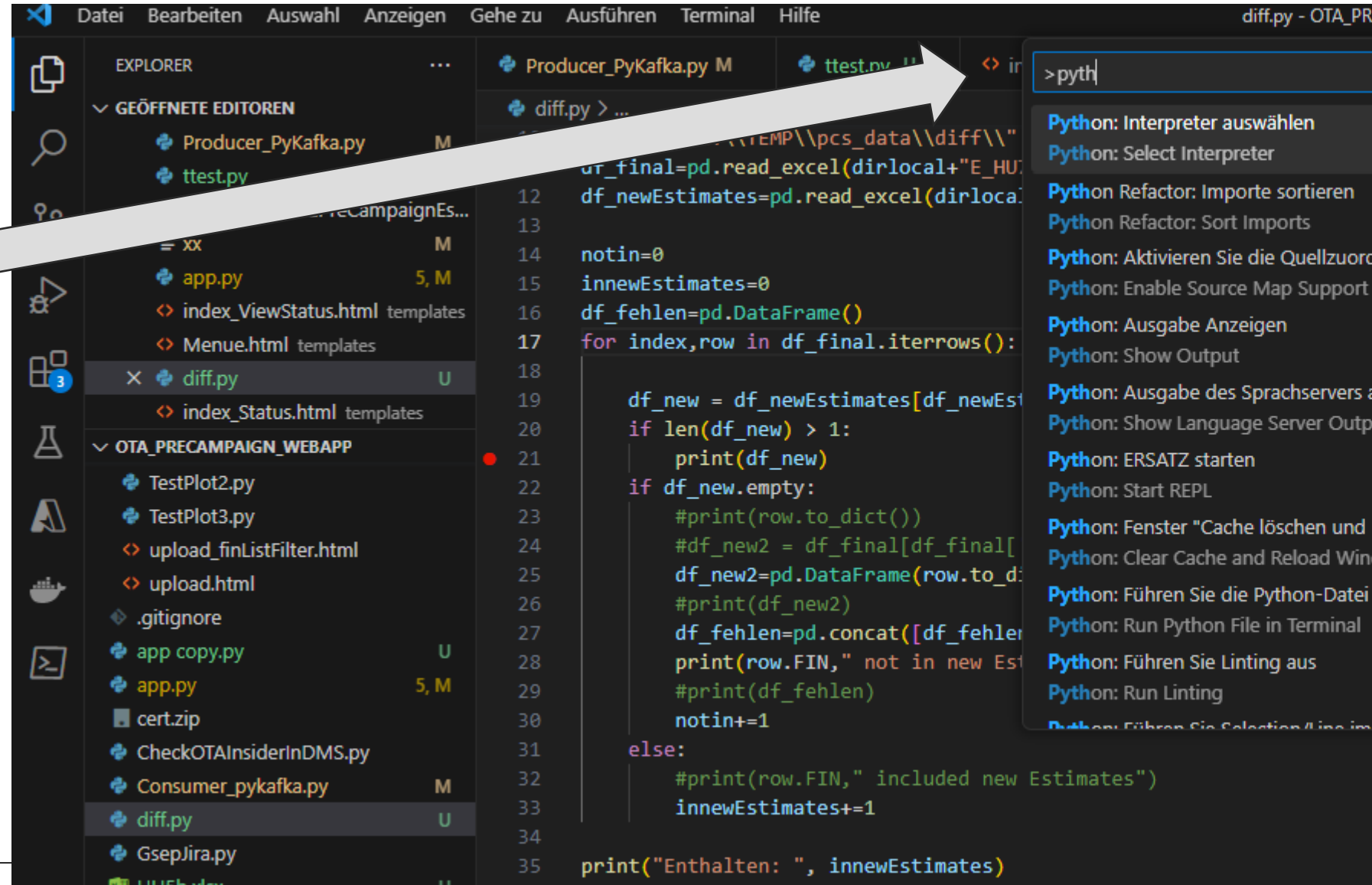


Code Setup

Select Python Interpreter

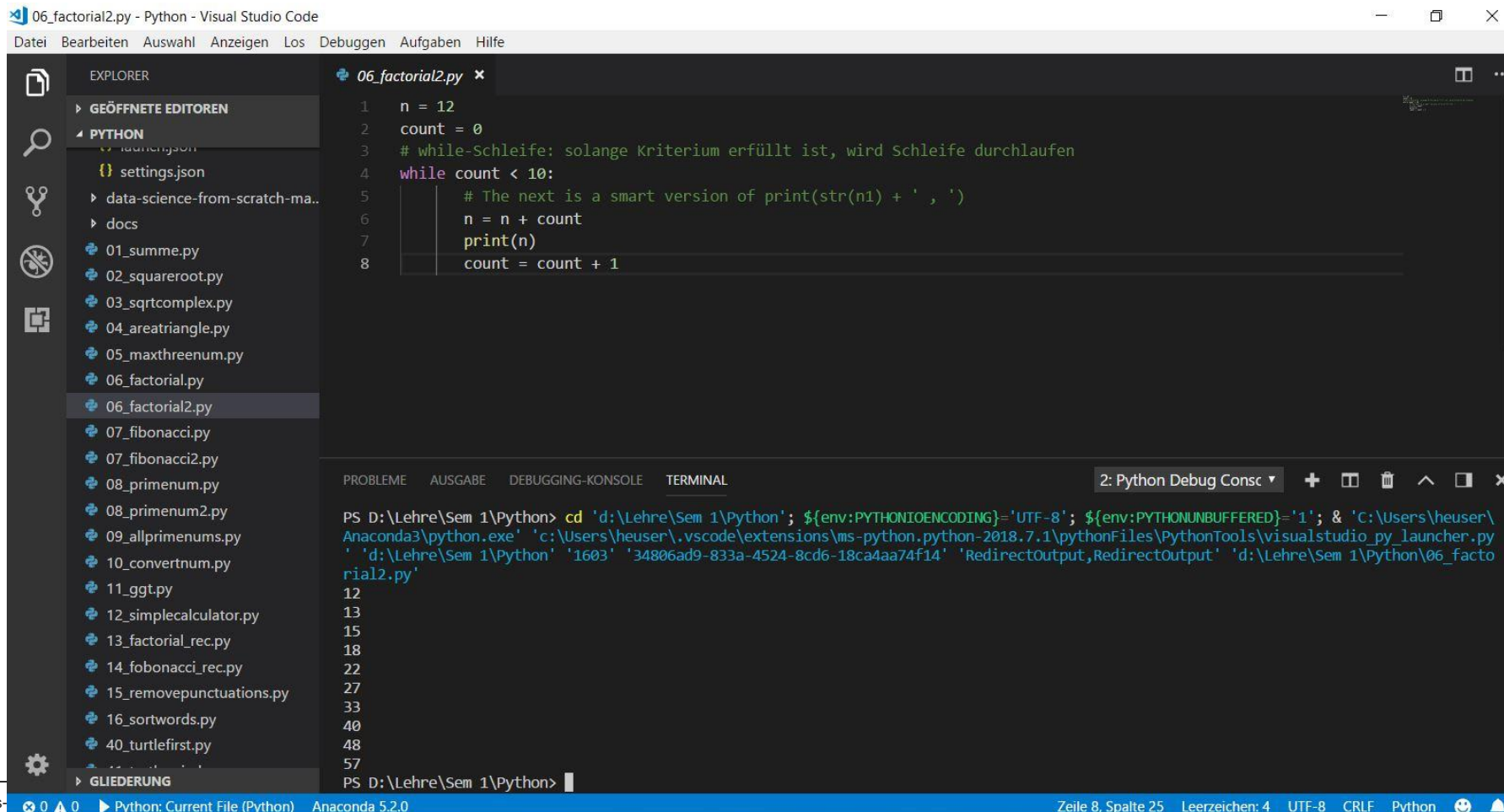
1. Press
Strg+Shift+P

2. Filter python



IDE – Entwicklungsumgebung

Visual Studio Code



IDE – Entwicklungsumgebung

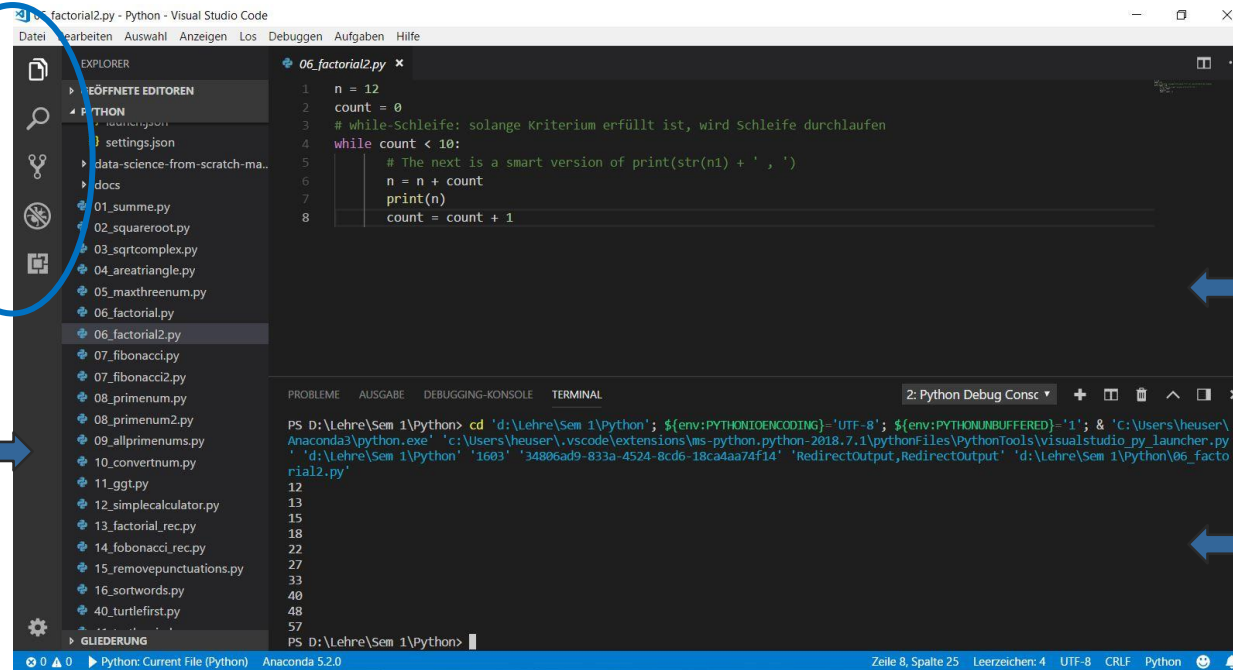
Visual Studio Code



Explorer,
Suchen,
Quellcode-
verwaltung,
Debuggen,
Erweiterungen

Datei-Explorer

Einstellungen

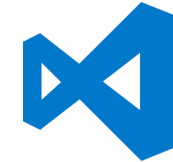


Code-Editor

Ausgaben,
Debugging-
Konsole,
Terminal,
Probleme.

IDE – Entwicklungsumgebung

Visual Studio Code: Debugging



Haltepunkt



```
06_factorial2.py x
1  n = 12
2  count = 0
3  # while-Schleife: solange Kriterium erfüllt ist, wird Schleife durchlaufen
4  while count < 10:
5      # The next is a smart version of print(str(n1) + ' ', ')
6      n = n + count
7      print(n)
8      count = count + 1
```

Im Debug-Modus lassen sich mit der Maus vor der Zeilennummerierung Breakpoints (Haltepunkte) setzen. Der Haltepunkt erscheint als roter Punkt vor der Zeile (in unserem Beispiel Zeile 6; s. oben). Der Debugger wird anschließend gestartet. Das Skript stoppt in Zeile 6. An diesem Punkt angelangt, können Variablen mit Hilfe von `Debuggen > Überwachen` überwacht werden. In unserem Beispiel sollen dies die Variablen `count` und `n` sein (s. nächste Folie).

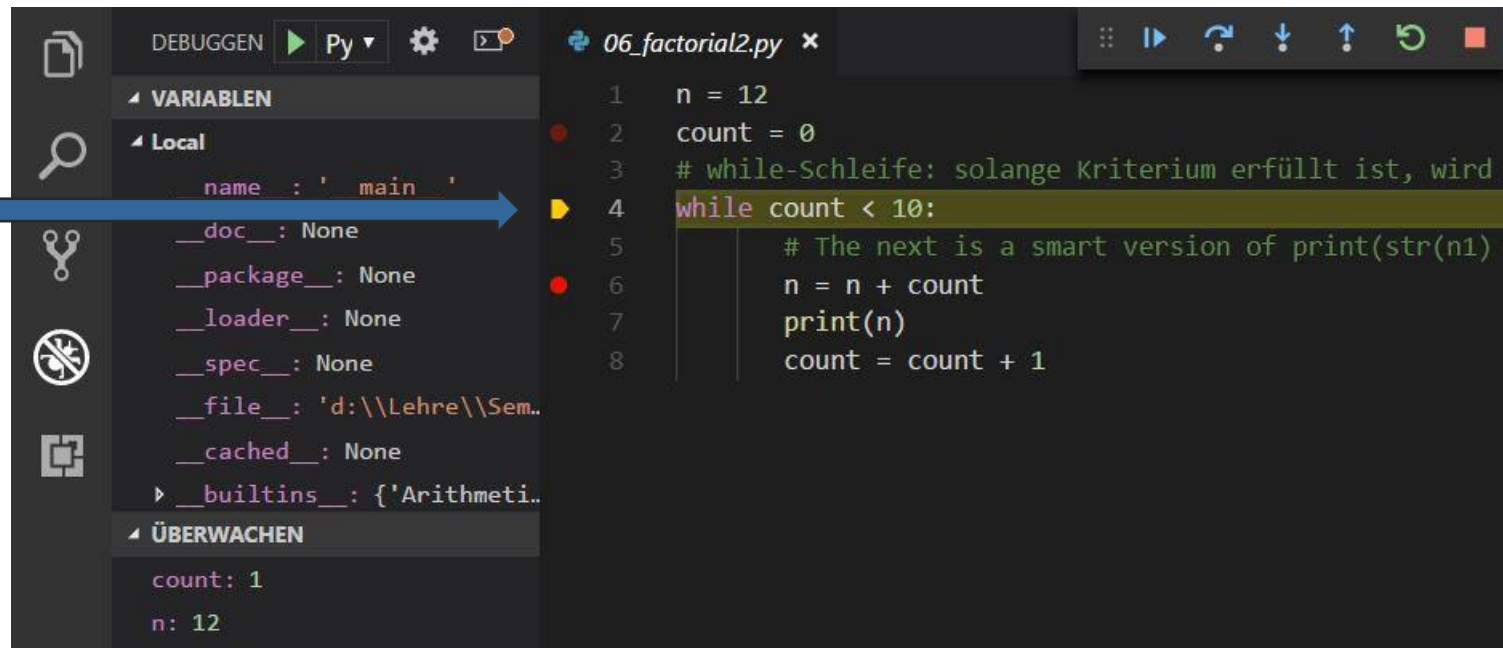
IDE – Entwicklungsumgebung

Visual Studio Code: Debugging



Aktueller Codeverlauf

Überwachte Variablen



Mit Debuggen > Prozedur-/Einzelschritt lässt sich das Programm sukzessive weiter bewegen. Dabei lassen sich die Inhalte der überwachten Variablen zum aktuellen Skriptverlauf (gelber Pfeil; in unserem Beispiel Codezeile 4) auf der linken Seite überwachen.

IDE – Entwicklungsumgebung

Alternative IDE: Visual Studio Enterprise

