Debugging & Matplotlib

Jannusch Bigge

28.11.2023

Matplotlib

Matplotlib

Matplotlib is a plotting library.

Matplotlib

Matplotlib is a plotting library.

Many different plots are possible.

Matplotlib

Matplotlib is a plotting library.

Many different plots are possible.

- Line plots
- Scatter plots
- Bar plots
- 3D plots
- ...

Matplotlib

Works great together with other libraries (e.g. numpy).

Matplotlib

Works great together with other libraries (e.g. numpy).

Allows you to customize your plots.

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode
  - Global state-based interface mode

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode
  - Global state-based interface mode
  - like MATLAB

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode
  - Global state-based interface mode
  - like MATLAB
  - not recommended (but always used on the internet)

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode
- Explicit mode

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode
- Explicit mode
  - Object-oriented (OO) interface

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode
- Explicit mode
  - Object-oriented (OO) interface
  - Build up the visualization in an instance of *figure.Figure*

Matplotlib - HowTo

Attention
Matplotlib has two modes:

- Implicit mode
- Explicit mode
  - Object-oriented (OO) interface
  - Build up the visualization in an instance of *figure.Figure*
  - **Recommended to use!**

Implicit mode

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4], [0, 0.5, 1, 0.2])
```

Matplotlib - HowTo

Implicit mode

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4], [0, 0.5, 1, 0.2])
```

Explicit mode

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.subplots()
ax.plot([1, 2, 3, 4], [0, 0.5, 1, 0.2])
```

Matplotlib - Why explicit?

- More control over your plots

Matplotlib - Why explicit?

- More control over your plots
- If you have to work on an old unreferenced axes

Matplotlib - Why explicit?

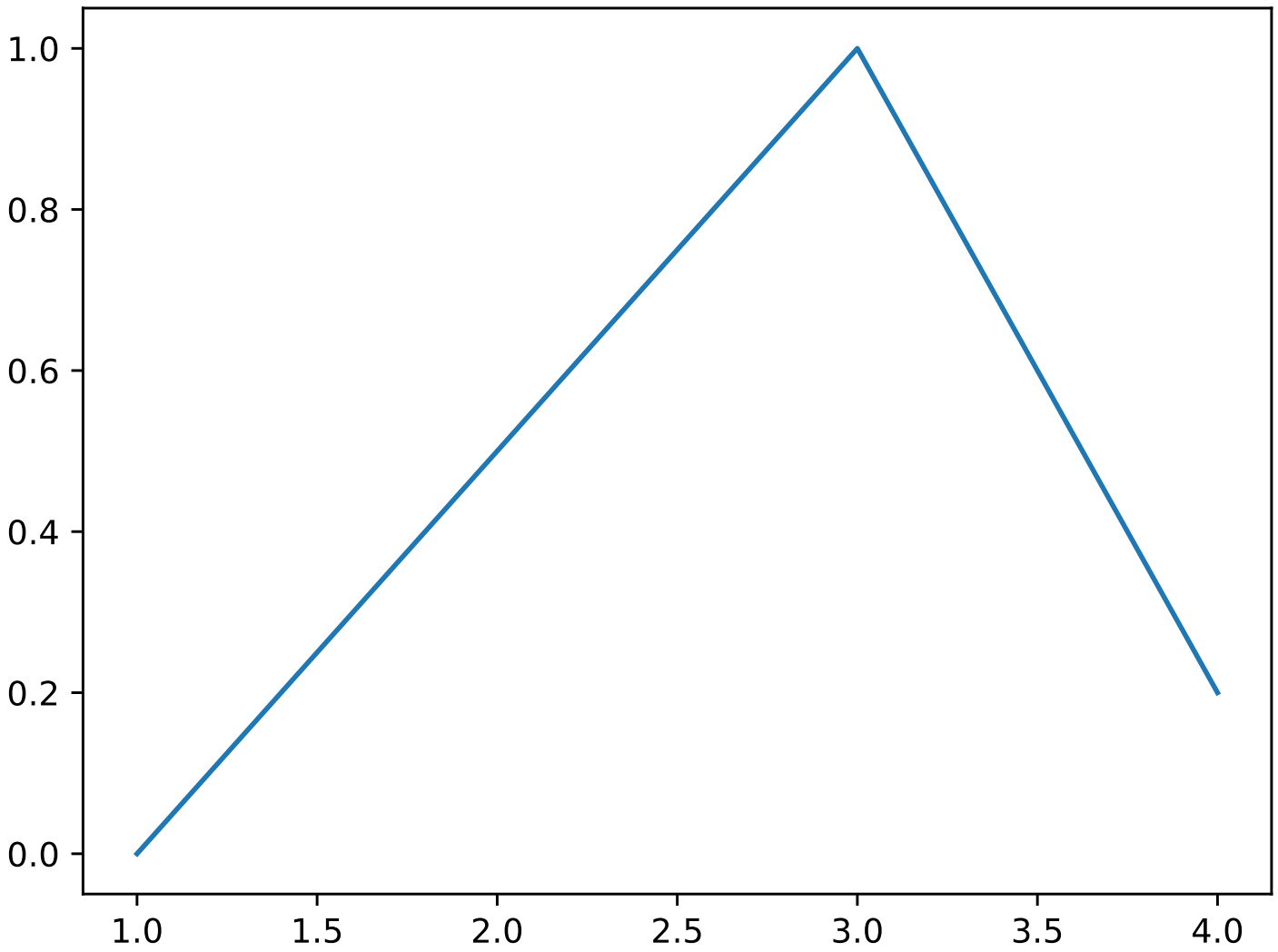
- More control over your plots
- If you have to work on an old unreferenced axes
- Third party often uses explicit mode

Matplotlib - Example

```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.subplots()
ax.plot([1, 2, 3, 4], [0, 0.5, 1, 0.2])
```

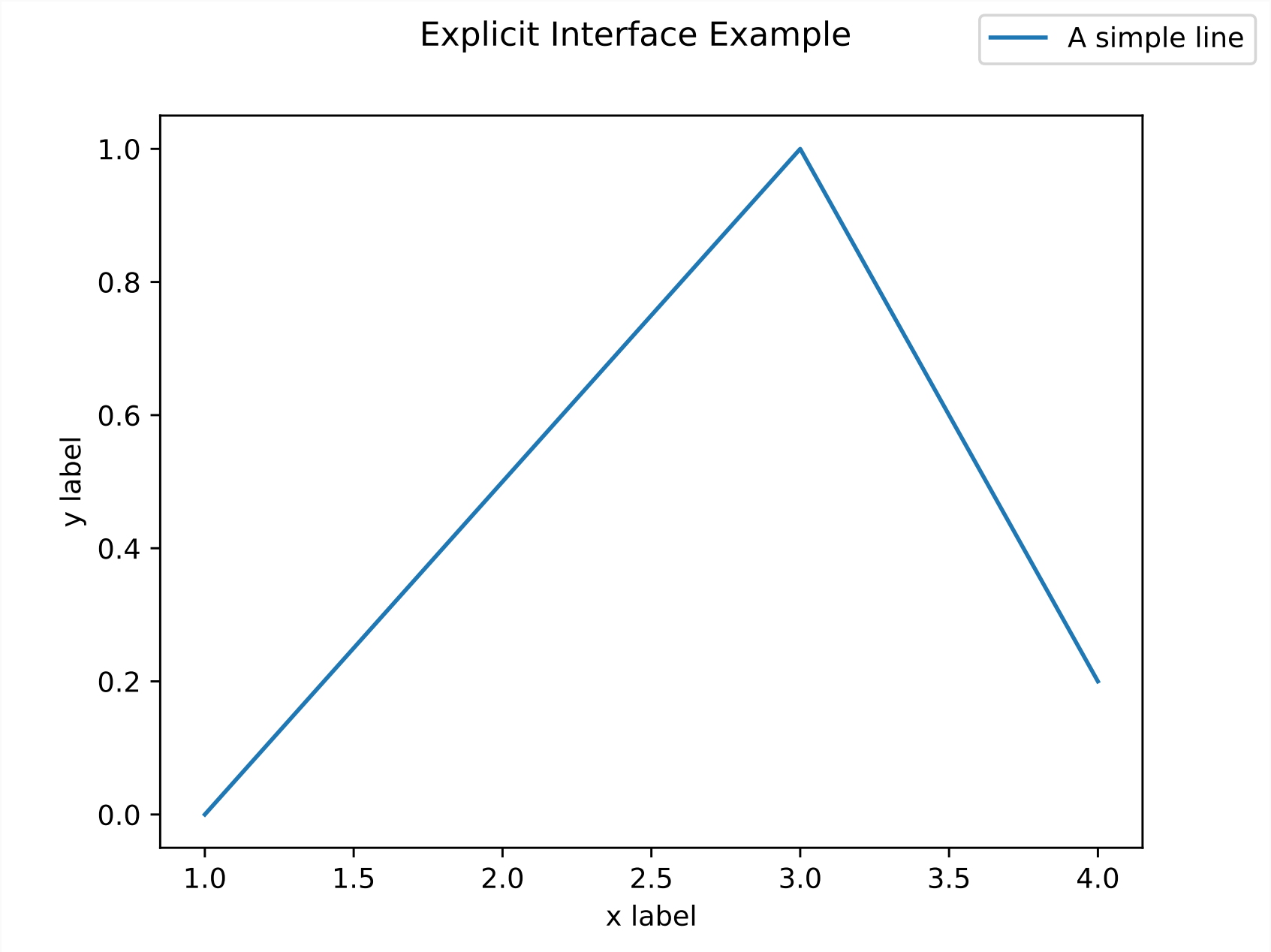
Matplotlib - Example



Matplotlib - Example

```
# Add some text to the figure
fig.suptitle('Explicit Interface Example')
ax.set_xlabel('x label')
ax.set_ylabel('y label')
```

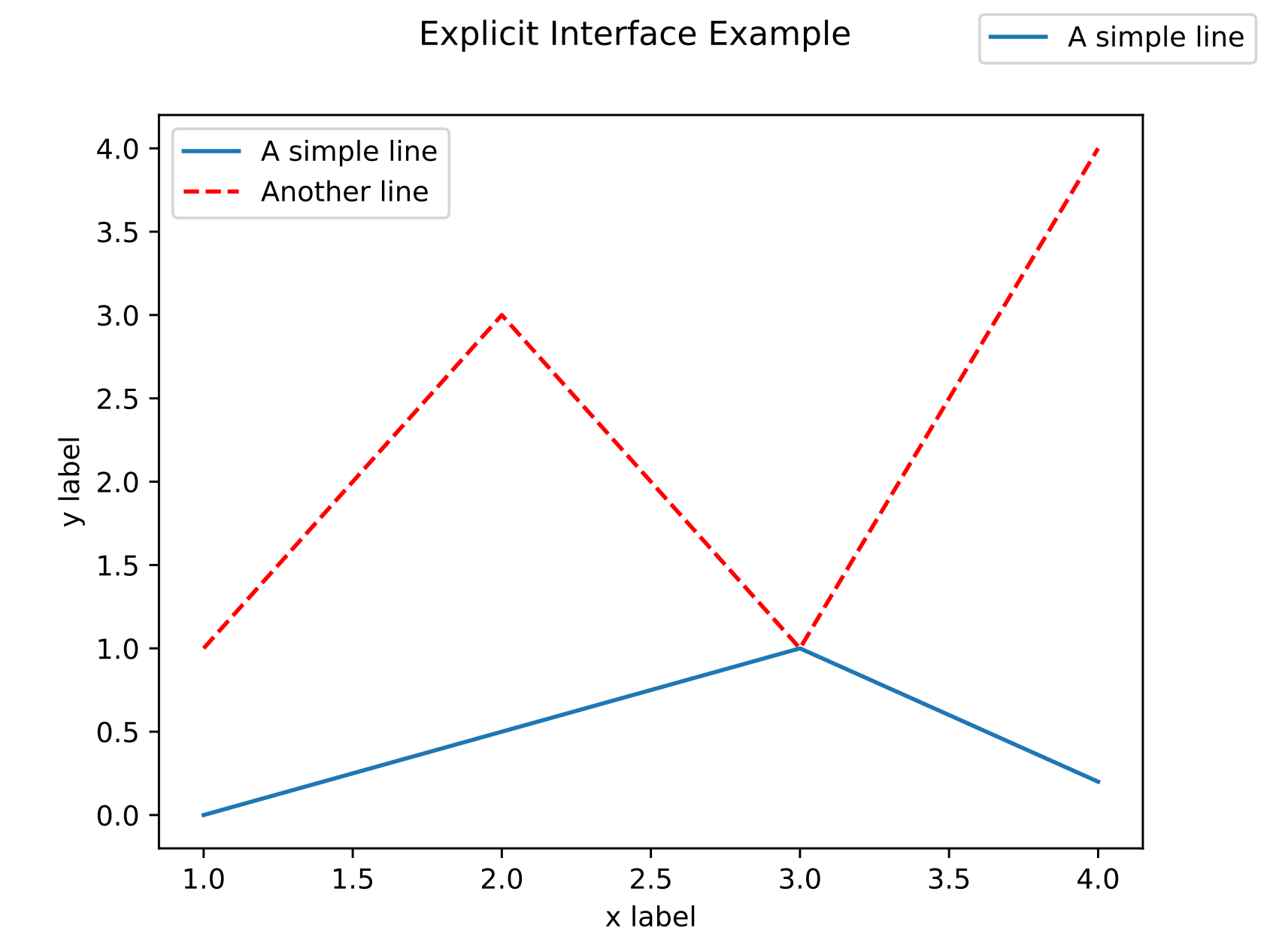
Matplotlib - Example



Matplotlib - Example

```
# Add another line and a second legend
ax.plot([1, 2, 3, 4], [1, 3, 1, 4], 'r--')
ax.legend(['A simple line', 'Another line'])
# And save the figure
fig.savefig("/workspaces/python_course/slides/five/figures/myplot.
    pdf")
```

Matplotlib - Example



Debugging

Debugging

- Debugging is the process of finding and fixing errors in your code

Debugging

- Debugging is the process of finding and fixing errors in your code
- Special program attached to the running program

Debugging

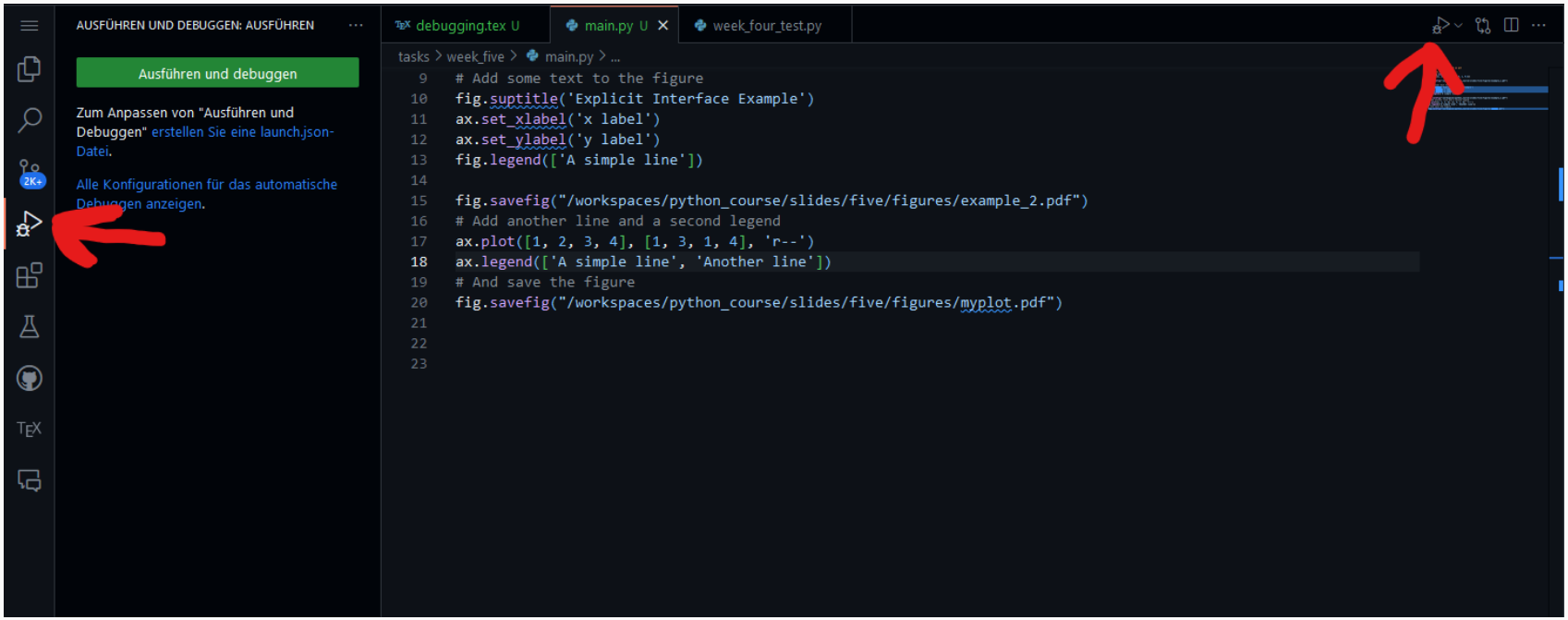
- Debugging is the process of finding and fixing errors in your code
- Special program attached to the running program
- Allows you to inspect the program at runtime

Debugging

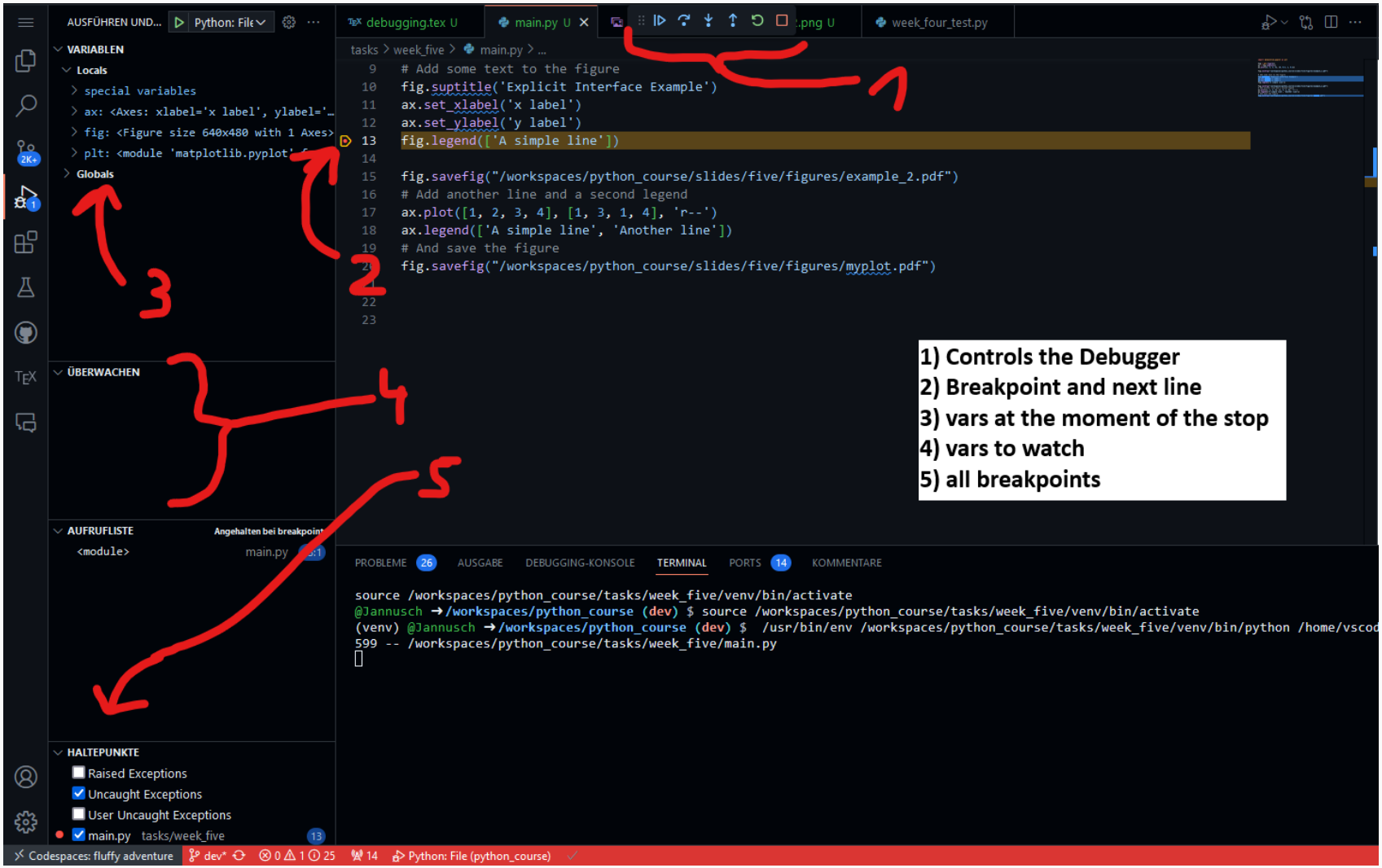
Usecases:

- cath errors and show you the code and the state of the code
- set breakpoints and hold the program at a certain point

Debugging - HowTo



Debugging - HowTo



Debugging - Example

Task

Task