



Programming with Python

Introduction



History of Python*

- Developed by Guido van Rossum in Amsterdam in the early 1990s.
- Object-oriented and modular from the start
- 1994: Supplemented by functional concepts (lambda)
- 2000: Complemented by garbage collection and Unicode
- Current version: Python 3 (Python 3.10...)

* Tribute to the British comedy group *Monty Python*



What can Python do?

Python is used

- to create prototypes or scripts
- on the Server (server side of the web application)
- to read or write data (Files or Databases)
- to solve mathematical problems (statistics, data analysis, visualization, games,...)
- ...



Benefits of Python

- Python works on various platforms (Windows, Mac, Linus, Raspberry Pi, ...)
- Python has a simple syntax and is therefore easy to learn
- Python programs come with little overhead
- Python can be used procedurally, functionally or object-oriented



Your First Python Program

Python can be run directly from the console

```
C:\>"Program Files\Python\python.exe"
```

```
>>> print("Hello Word!")
```

```
Hello World!
```

```
>>> 3*4+16-9
```

```
19
```

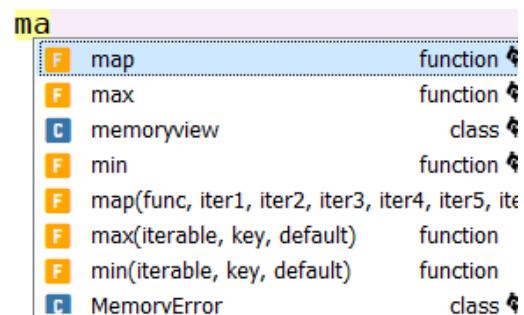
```
>>>
```

```
C:\>"Program Files\Python\python.exe"
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18)
Type "help", "copyright", "credits" or "license" for more information
>>> print("Hello World!")
Hello World!
>>> 3*4+16-9
19
>>> 
```

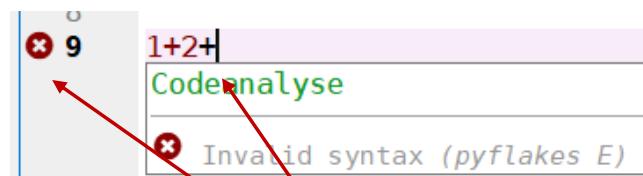


Programming environment

Programming is more convenient with support for a programming environment that knows the commands and displays typos.



All commands starting
with "ma" are displayed



Errors are displayed



Spyder Programming Environment

The screenshot shows the Spyder Python IDE interface. The menu bar includes Datei, Bearbeiten, Suchen, Quellcode, Ausführen, Debug, Konsolen, Projekte, Werkzeuge, Ansicht, and Hilfe. The toolbar has icons for file operations like Open, Save, and Run. The left sidebar shows project files: twoEqual.py and unbenannt0.py*. The main code editor window displays:

```
1 # -*- coding: utf-8 -*-
2
3 a = 1+2
4
5 b = 3+4
6
7
```

To the right is the Variable Manager, showing the state of variables:

Name	Typ	Größe	Wert
a	int	1	3
b	int	1	7

Below the Variable Manager is the IPython console output:

```
IPython 7.12.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Temp/PythonBsp/ITEinfuehrung/unbenannt0.py', wdir='C:/Temp/PythonBsp/ITEinfuehrung')

In [2]:
```

The status bar at the bottom indicates: Kite: ready (no index), conda: base (Python 3.7.6), Line 7, Col 1, UTF-8, CRLF, RW, Mem 55%.



Miniconda / Anaconda / Spyder



Data science technology for
human sensemaking.

A movement that brings together millions of data science practitioners, data-driven enterprises, and the open source community.



Installation

The screenshot shows the left sidebar of a website. At the top is a green header bar with the text "conda latest" and a search bar labeled "Search docs". Below this are two main categories: "Conda" and "Conda-build". Under "Conda", there is a collapsed section titled "Miniconda" which contains the following links: "System requirements", "Latest Miniconda Installer Links", "Windows installers", "macOS installers", "Linux installers", "Installing", "Other resources", and "Help and support".

Docs » Miniconda

Edit on GitHub

Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the `conda install` command to install 720+ additional conda packages from the Anaconda repository.

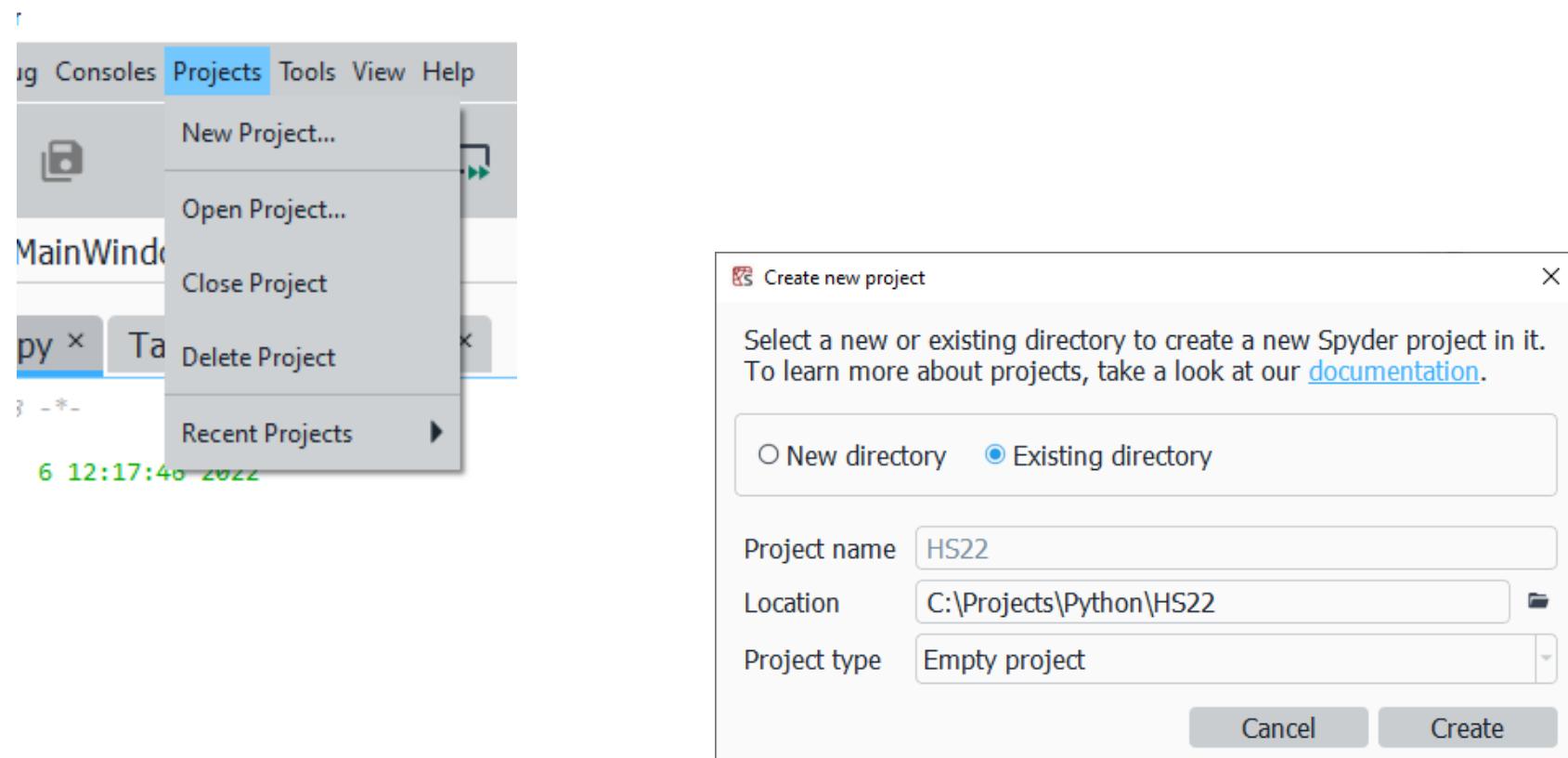
[See if Miniconda is right for you.](#)

System requirements

- License: Free use and redistribution under the terms of the [EULA for Miniconda](#).
- Operating system: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 7+, and others.
- If your operating system is older than what is currently supported, you can find older



Create a new Python project





Start Python Interpreter

The screenshot shows the Spyder IDE interface. The top menu bar includes file operations like New, Open, Save, and Run. The main area displays a code editor for a file named 'temp.py' located at 'C:\Users\aebl.spyder-py3\temp.py'. The code contains a docstring, variable assignments (x=5, y=2.3, z=[1,2,3]), and a print statement. To the right is a Variable Explorer showing the values of x, y, and z. Below it is a Console window showing the output of running the script.

```
C:\Users\aebl.spyder-py3\temp.py
```

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.

"""

x = 5
y = 2.3
z = [1,2,3]

print("z=", z)
```

Name	Type	Size	Value
x	int	1	5
y	float	1	2.3
z	list	3	[1, 2, 3]

```
In [1]: runfile('C:/Users/aebl/.spyder-py3/temp.py', wdir='C:/Users/aebl/.spyder-py3')
z= [1, 2, 3]
```



Console / Variable Explorer

The screenshot shows the Spyder IDE interface with three main panes:

- Editor Pane:** Displays the script file `temp.py` with the following code:

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.

"""

x = 5
y = 2.3
z = [1,2,3]

print("z=", z)
```
- Variable Explorer Pane:** Shows a table of variables with their types and values:

Name	Type	Size	Value
x	int	1	5
y	float	1	2.3
z	list	3	[1, 2, 3]
- Console Pane:** Displays the output of the command run in the console:

```
In [1]: runfile('C:/Users/aeb1/.spyder-py3/temp.py', wdir='C:/Users/aeb1/.spyder-py3')
z= [1, 2, 3]
```

Two blue arrows point from the text "Variable Explorer" in the slide title down to the Variable Explorer pane in the screenshot.