

Einführung in **Python** für Geisteswissenschaftler:innen

Melanie Althage (melanie.althage@hu-berlin.de)

4.-8. März 2024,
Campus Essen, Universität Duisburg-Essen

Tag 1



Ziele des Workshops

- Grundprinzipien und -konzepte von Python verstehen und anwenden können
- Verständnis für die Bedeutung von Python für die Anwendung in den Geisteswissenschaften
 - Erste exemplarische Einblicke in Datensammlung, -verarbeitung, -analyse und -visualisierung
- Kenntnisse von Ressourcen für die Hilfe zur Selbsthilfe
- Überblick zum Jupyter-Project

Overall:
Erste Programmierkenntnisse als Basis
für eigenständiges Arbeiten mit Python!

Selbsteinschätzung

- Wie schätzen Sie Ihre Programmierkenntnisse auf der inoffiziellen Python-Skala von 1-10 ein?



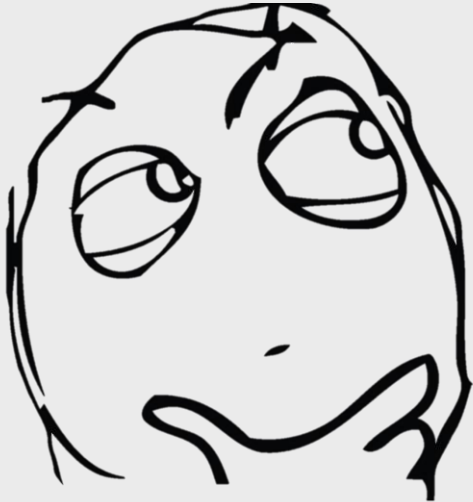
Ich habe noch
keine Programmier-
erfahrung

Ich kann bedarfsorientiert (mit
Recherche) Skripte/Programme in
Python schreiben

Ich bin Expert:in
und Top-
Kommentator:in
auf Stack
Overflow

Bitte sortieren Sie sich in Absprache
miteinander im Raum, anschließend folgt die
Vorstellungsrunde 😊

Vorstellungsrunde



- Wie heißen Sie?
- Welchen fachlichen Hintergrund haben Sie?
- Was sind Ihre Erwartungen und Wünsche an den Workshop?

Grober Plan für diese Woche

MONTAG

Python
Basics

DIENSTAG

Python
Basics

MITTWOCH

Data
Mining &
Visuali-
sierung
mit
Pandas

DONNERSTAG

Textverar-
beitung mit
Regulären
Ausdrücken
und spaCy

FREITAG

Text
Mining
&
Abschluss

Arbeitsweise

- kurze theoretische Inputs
- Livecoding-Einheiten im Wechsel mit der eigenständigen Bearbeitung von Übungsaufgaben
- Zentral: **Pairprogramming**
 - Gemeinsam coden macht mehr Spaß und wir lernen voneinander!
 - Explizit erwünscht: Tauschen Sie sich bei der Bearbeitung der Übungsaufgaben aus und unterstützen Sie sich gegenseitig.
- Immer erlaubt:
 - Fragen stellen
 - Online-Recherche

Lerntagebuch:

Am Ende des Workshoptages reflektieren, was gelernt wurde & was noch vertieft werden darf.

Ressourcen zum Workshop

- [GitHub-Repository](#)
 - Notebooks
 - Exemplarische Datensätze
 - Präsentation
 - Vorlage Lerntagebuch
- Jupyter Book
[Python für Historiker:innen](#)



Lernziele des heutigen Workshop-Tages

- Überblick über das Python-Ökosystem
- Grundverständnis für die Arbeit mit Jupyter Notebooks
- Grundverständnis für die Grundlagen der Programmierung in Python
 - Atomare Datentypen: String und Integer
 - Kontrollstrukturen:
 - Bedingte Anweisungen
 - Schleifen

Einführung

Einsatzgebiete und Anwendungsszenarien
Python-Ökosystem
Jupyter Notebooks





Warum Python?

- einfach zu lernen
- einfach zu lesen
- riesige Community
- Open Source
- viele Programm-Bibliotheken zur Erweiterung verfügbar

Python als Skriptsprache

- interpretiert
- höhere Programmiersprache
- Python-Skripte mit der Endung:
`skript_name.py`
- Jupyter Notebooks mit der Endung:
`notebook_name.ipynb`





Einsatzbereiche

- Universitäten und Forschungseinrichtungen
- Technologie-Branche
- Industrie
- Data Science

Einsatzgebiete

- Data and Text Mining
- Datenanalyse
- Visualisierung
- Web Entwicklung
- System-Administration
- Rapid Prototyping
- Machine Learning



Anwendungsbeispiele für Historiker:innen

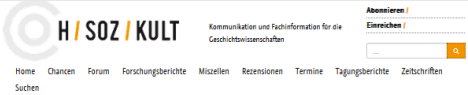
Web Scraping

```
import csv
import requests
from bs4 import BeautifulSoup
import re
import time

"""Basic functions to fetch data"""

def get_info(variable, string):
    """variable = css-selector-object (list)
    string = search term
    returns one string object"""
    meta_list = []
    for each in variable:
        variable = each.text.strip()
        variable = re.sub("\s+", " ", variable)
        meta_list.append(variable)
    if str(string) in meta_list:
        info_index = meta_list.index(str(string)) + 1
        return meta_list[info_index]
    else:
        return "not found"

def get_keywords(text):
    """searches the HTML document for the information specified by the parameter (text)
    returns two lists (text content and link collection)"""
    if soup.find("div", string=text):
        text = soup.find("div", string=text)
        text_text = []
        text_link = []
```



Vertreibung und Erinnerung.
Forschungsstand und Geschichtspolitik im östlichen Europa

Info
Drucken
PDF

Vertreibung und Erinnerung, Forschungsstand und Geschichtspolitik im östlichen Europa

Organisatoren: Katrin Bonck, Forschungsstelle Kultur und Erinnerung, Heimatvertriebene und Aussiedler in Bayern, Leibniz-Institut für Ost- und Südosteuropaforschung, Regensburg (Leibniz-Institut für Ost- und Südosteuropaforschung)

Ausrichter: 9504

PLZ: Regensburg

Land: Deutschland

Fand statt: In Präsenz

Vom - Bis: 06.10.2023 - 07.10.2023

Von: Maximilian Sommer, Forschungsstelle Kultur und Erinnerung, Heimatvertriebene und Aussiedler in Bayern, Leibniz-Institut für Ost- und Südosteuropaforschung, Regensburg

Die 2022 von der bayerischen Landesregierung ins Leben gerufene Forschungsstelle „Kultur und Erinnerung. Heimatvertriebene und Aussiedler in Bayern“ hat eine internationale Konferenz zum Thema „Vertreibung und Erinnerung. Forschungsstand und Geschichtspolitik im östlichen Europa“ organisiert, zu der viele Vortragende, vor allem aus dem östlichen Europa nach Regensburg kamen. Die Tagung hatte zum Ziel, die historische Forschung über dieses Thema in der Öffentlichkeit präsent zu machen und wissenschaftliches Arbeiten durch Vernetzung und Austausch zu fördern. Im Vordergrund stand der Forschungsstand in den jeweiligen Ländern. Hierbei war vor allem durch den internationalen und interdisziplinären Vergleich ein Mehrwert zu erwarten, bei dem nicht nur die materiellen Güter, welche die

| year | report_issued | report_ID | report_first_creator | report_second_creator | report_third_creator | url_report_creator | raw_report_creator | event_title | event_organizer | event_date_start | event_date_end | event_city | event_country |
|------|---------------|-----------|-----------------------|-----------------------|----------------------|-------------------------|--|---|--|------------------|----------------|------------|---------------|
| 1996 | 1996-12-18 | 1930 | Ame Dells | | | [person/betrager-5170] | Ame Dells, John F. Kennedy Institut | Self and Community | interdisziplinäres Graduiertenkolleg "Democracy in the U.S." | 1996-12-12 | 1996-12-14 | Berlin | Deutschland |
| 1997 | 1997-06-13 | 1942 | Marco Bellabarba | | | [person/betrager-5287] | Marco Bellabarba, Reinhard Stauber | Traditionsbildung und Veränderung politischer Identitäten im alpinen Raum | Istituto Storico Italiano-Germanico in Trento; Institut für Neuere Geschichte der Universität München; Fritz-Thyssen-Stiftung, Köln; autonome Region Trentino-Südtirol, Trient/Bolzen, Verein für italienisch-deutsche Geschichtsforschung, Trient | 1997-04-10 | 1997-04-12 | Trient | Italy |
| 1997 | 1997-06-16 | 1939 | Arpad von Klimó | | | [person/betrager-6319] | Arpad von Klimó, Zentrum für Zeithistorische Forschung | Die Volksgeschichte der NS-Zeit. Vorläufen der Sozialgeschichte der Bundesrepublik/ Wiener Conze und Theodor Schieder in der Diskussion | Freie Universität Berlin, Arbeitsstelle für Vergleichende Gesellschaftsgeschichte | 1997-06-09 | 1997-06-09 | Berlin | Deutschland |
| 1997 | 1997-07-24 | 1941 | Katja Schlichtenbrede | | | [person/betrager-39871] | Katja Schlichtenbrede | Theorie und Praxis des Diktaturvergleichs | Günther Heydemann, Eckhard Jesse, Universität Leipzig, (V. Symposium der J'achgruppe Geschichte der Gesellschaft für Deutschlänrforschung (GfD) | 1997-05-09 | 1997-05-10 | Leipzig | Deutschland |

Topic Modeling

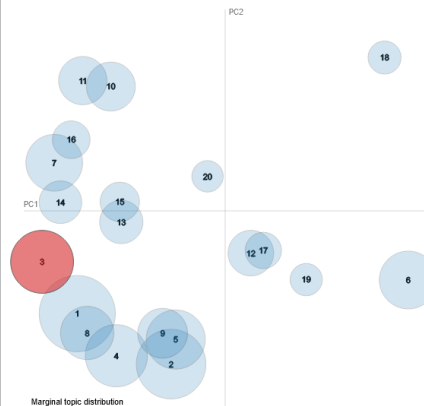
Selected Topic: 3 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric: (2)

$\lambda = 1$

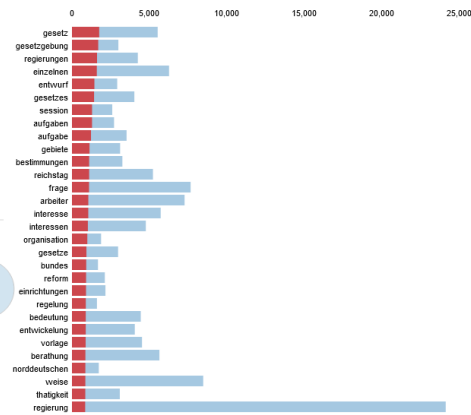
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Marginal topic distribution

Top-30 Most Relevant Terms for Topic 3 (7.7% of tokens)



Overall term frequency

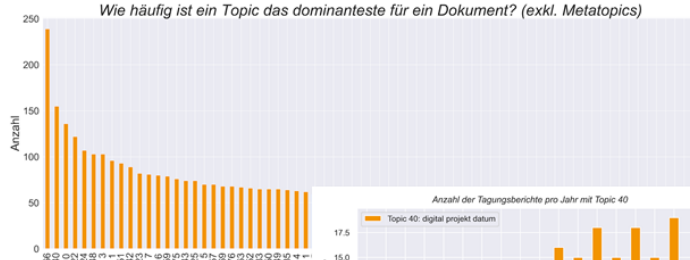
Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * (sum_i p(i | w) * log(p(i | w)/p(i))) for topics i; see Chuang et al. (2012)
2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | \text{top}(w))$; see Sievert & Shirley (2014)

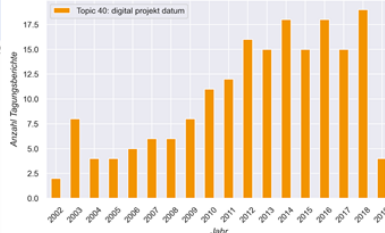
Christliche Religionsgeschichte



Wie häufig ist ein Topic das dominanteste für ein Dokument? (exkl. Metatopics)



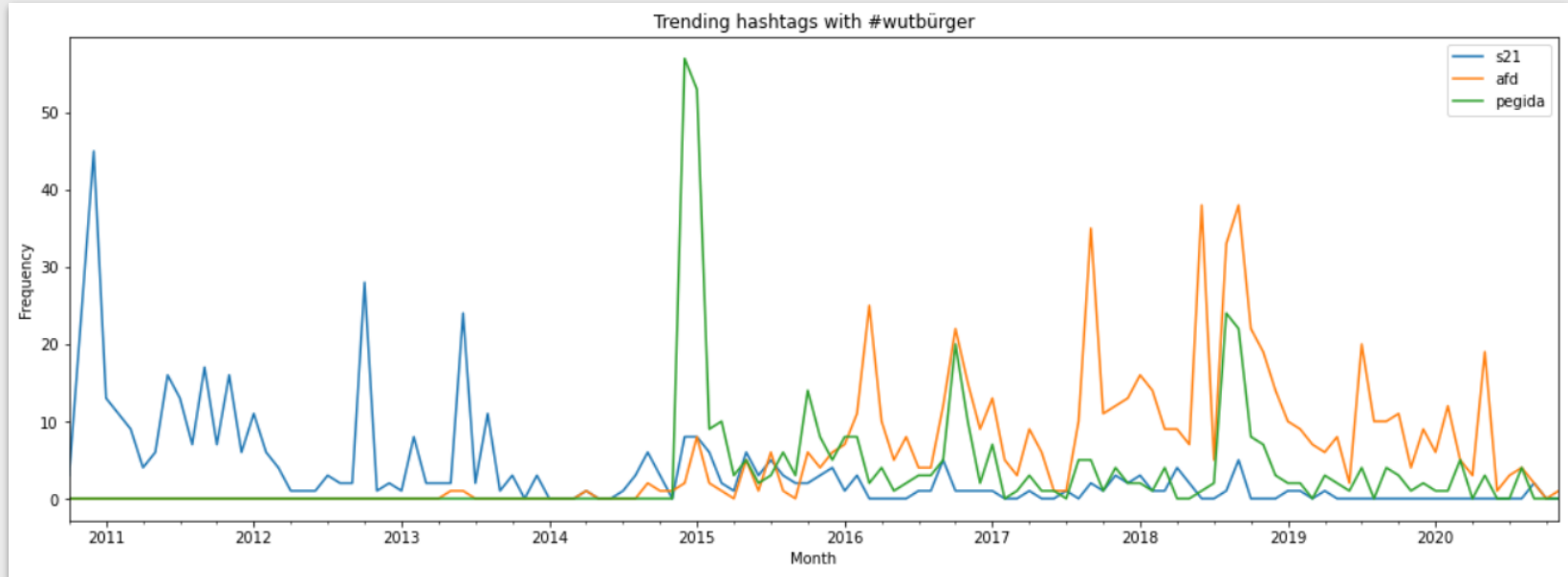
Anzahl der Tagungsberichte pro Jahr mit Topic 40



Martin Dröge

Melanie Althage

Twitter Mining anhand von Hashtags



Martin Dröge

Stilometrie bzw. Authorship Attribution

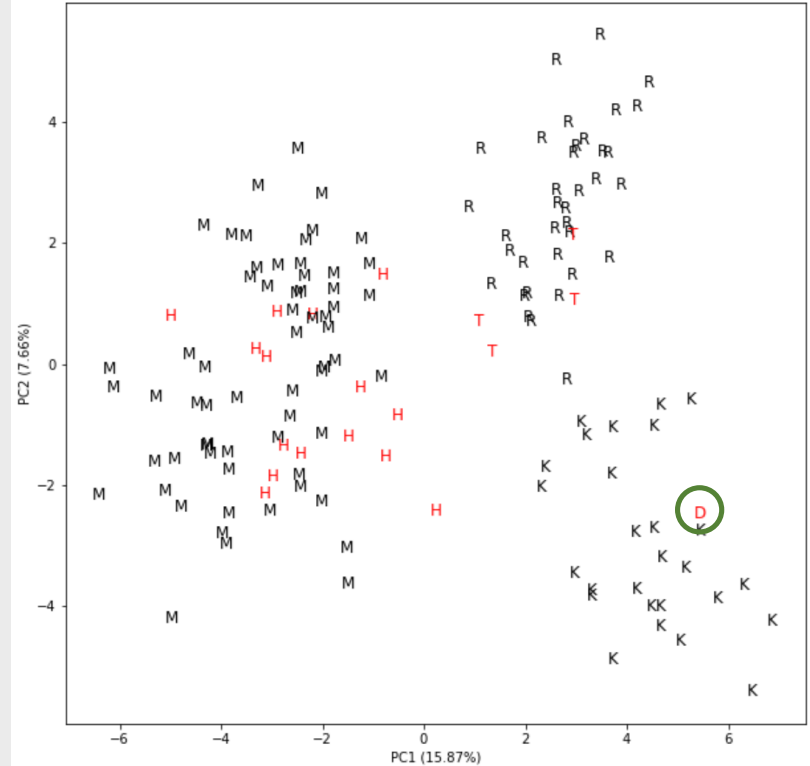
Fragestellung: Wer hat die 1923 erschienene Biografie „Adolf Hitler: Sein Leben und seine Reden“ geschrieben?

Vermeintlicher Autor der Biografie: Baron Adolf Victor von Koerber, der dem konservativen Lager nahestand.

Kontroverse:

- Der Historiker Thomas Weber (Universität in Aberdeen) vertrat hingegen die Meinung, Adolf Hitler habe diese Biografie selbst geschrieben, noch vor „Mein Kampf“, das er zwischen 1925 und 1926 verfasste.
- Winfried Meyer (TU Berlin) wiederum suchte durch klassische, analoge Quellenarbeit diese Position zu widerlegen.

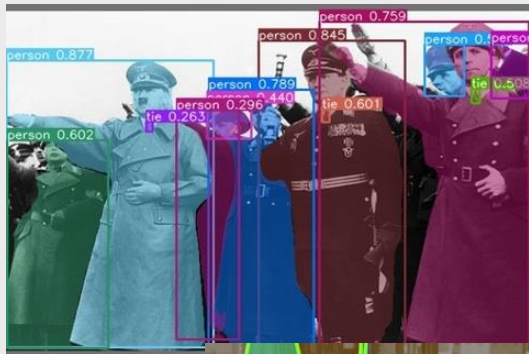
Martin Dröge, Torsten Hiltmann



Visualisierung der Ergebnisse mittels PCA:

- D ist das zu bestimmende Dokument
- T sind Testdokumente, um die Methode zu prüfen (anhand von R = Alfred Rosenberg)
- K ist von Koerber, der offiziell die Biografie geschrieben hat
- H steht für Vergleichsdokumente von Adolf Hitler aus der Zeit
- M steht für Auszüge „Mein Kampf“, das als Probe genutzt wurde

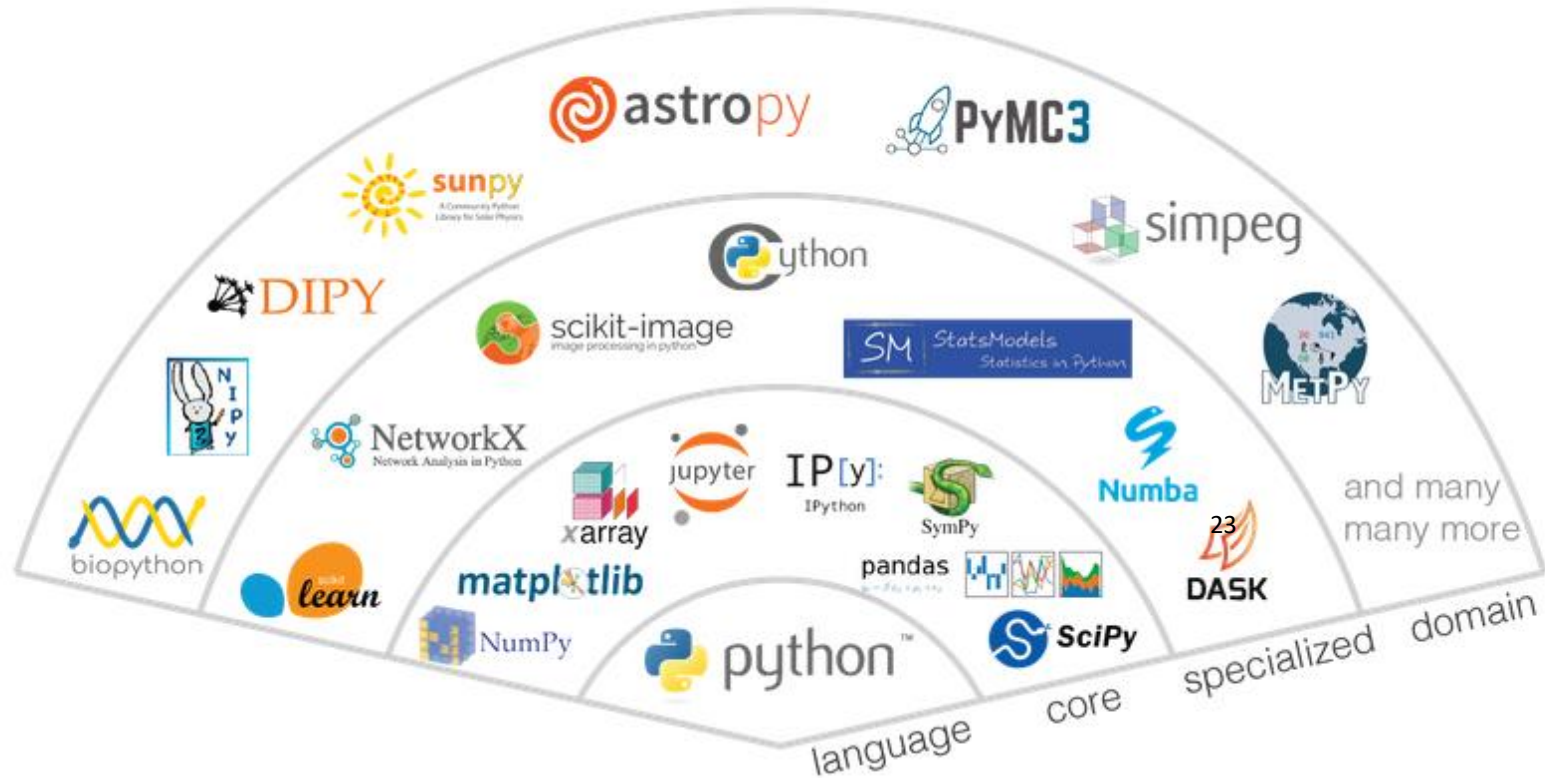
Computer Vision





Haben Sie Fragen?

Python Ökosystem



✂ The Classical Language Toolkit



NLTK

install

os

csv



json

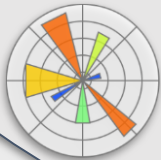
import

collections



string

regex



spaCy

pandas

NumPy

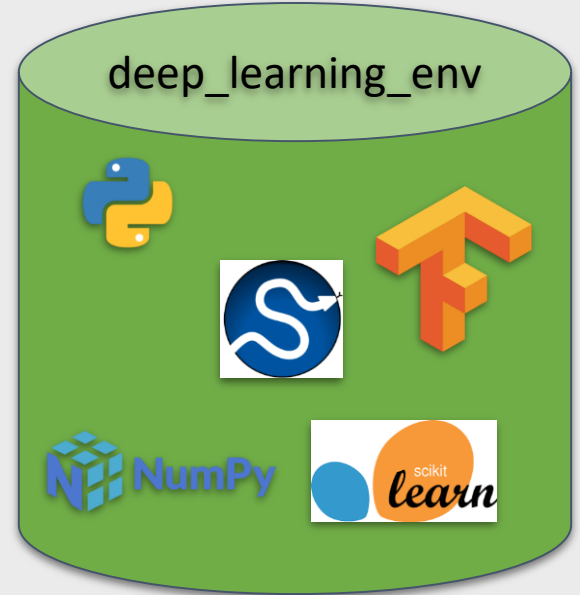
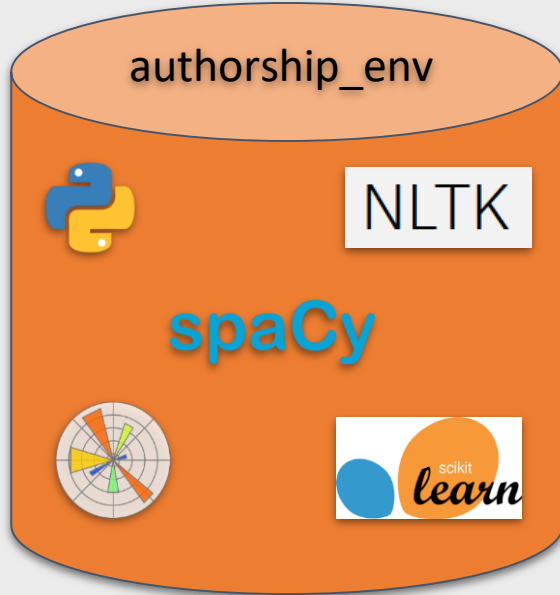
Package Manager conda und pip

CONDA



Real Python

Environments



Python Dokumentation

<https://docs.python.org/3/>

```
print(*objects, sep=' ', end='\n', file=None, flush=False)
```

Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file*, and *flush*, if present, must be given as keyword arguments.

All non-keyword arguments are converted to strings like `str()` does and written to the stream, separated by *sep* and followed by *end*. Both *sep* and *end* must be strings; they can also be `None`, which means to use the default values. If no *objects* are given, `print()` will just write *end*.

The *file* argument must be an object with a `write(string)` method; if it is not present or `None`, `sys.stdout` will be used. Since printed arguments are converted to text strings, `print()` cannot be used with binary mode file objects. For these, use `file.write(...)` instead.

Output buffering is usually determined by *file*. However, if *flush* is true, the stream is forcibly flushed.

Changed in version 3.3: Added the *flush* keyword argument.

The screenshot shows the Python 3.11.8 documentation page. At the top, there's a navigation bar with 'Python', 'English', '3.11.8', and '3.11.8 Documentation'. On the left, a sidebar contains links for 'Download', 'Docs by version' (listing Python 3.13 to 3.4), 'Tutorial', and 'What's new in Python 3.11?'. The main content area is titled 'Python 3.11.8 documentation' and includes a welcome message, a 'Parts of the documentation' section with links to 'What's new in Python 3.11?', 'Tutorial', 'Installing Python Modules', 'Distributing Python Modules', 'Extending and Embedding', 'Python/C API', and 'FAQs', and a 'Search' section with a 'Complete Table of Contents'.

Stackoverflow

<https://stackoverflow.com/>

How to print without a newline or space

Asked 15 years, 1 month ago Modified 6 months ago Viewed 2.5m times

27 Answers

Sorted by: Highest score (default)



In Python 3, you can use the `sep=` and `end=` parameters of the `print` function:

3283

To not add a newline to the end of the string:

```
print('.', end='')
```



To not add a space between all the function arguments you want to print:

```
print('a', 'b', 'c', sep='')
```



You can pass any string to either parameter, and you can use both parameters at the same time.

If you are having trouble with buffering, you can flush the output by adding `flush=True` keyword argument:

```
print('.', end='', flush=True)
```



stackoverflow

About

Products

For Teams

Search...

Log in

Sign up



Find the best answer to your technical



Want a secure, private space for your technical knowledge?

Get started

Organizations

For small teams

perhas a
overflow

5,000+

Stack Overflow for Teams
Instances active every day



dreftymac

31.9k • 26 • 122 • 185



Andrea Ambu

38.6k • 14 • 55 • 77

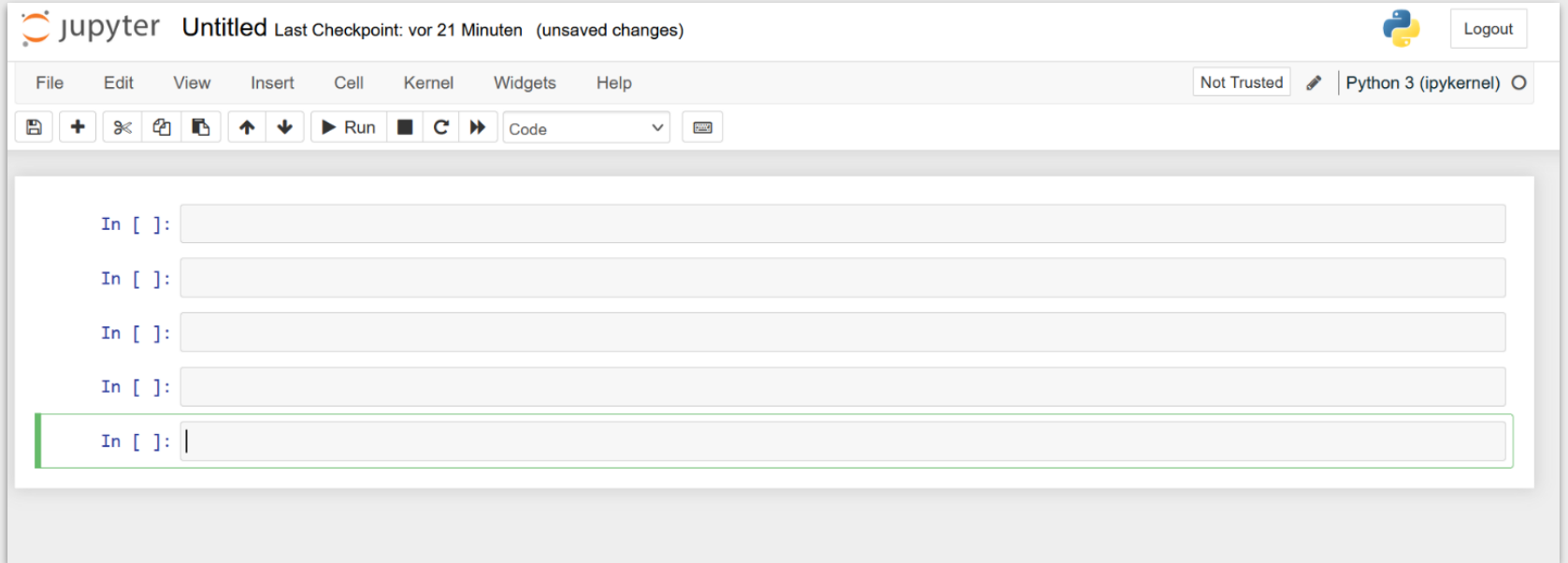
edited Apr 28, 2023 at 19:31

asked Jan 29, 2009 at 20:58

Project Jupyter



Jupyter Notebooks



The image shows the Jupyter Notebook web interface. At the top, the header bar includes the Jupyter logo, the text "jupyter", and "Untitled" followed by "Last Checkpoint: vor 21 Minuten (unsaved changes)". On the right side of the header, there is a Python logo and a "Logout" button. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu bar, there is a "Not Trusted" status indicator, a pencil icon, and "Python 3 (ipykernel)" with a refresh icon. Below the menu bar is a toolbar with icons for saving, adding a new file, undo, redo, copy, paste, scroll up, scroll down, run, interrupt kernel, restart kernel, and a dropdown menu currently set to "Code". The main area of the notebook contains five code cells, each starting with "In []:". The bottom-most cell is currently selected, indicated by a green border and a green vertical bar on the left.

jupyter Untitled Last Checkpoint: vor 21 Minuten (unsaved changes) Python 3 (ipykernel) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Save Add Undo Redo Copy Paste Scroll Up Scroll Down Run Interrupt Restart Code

In []:

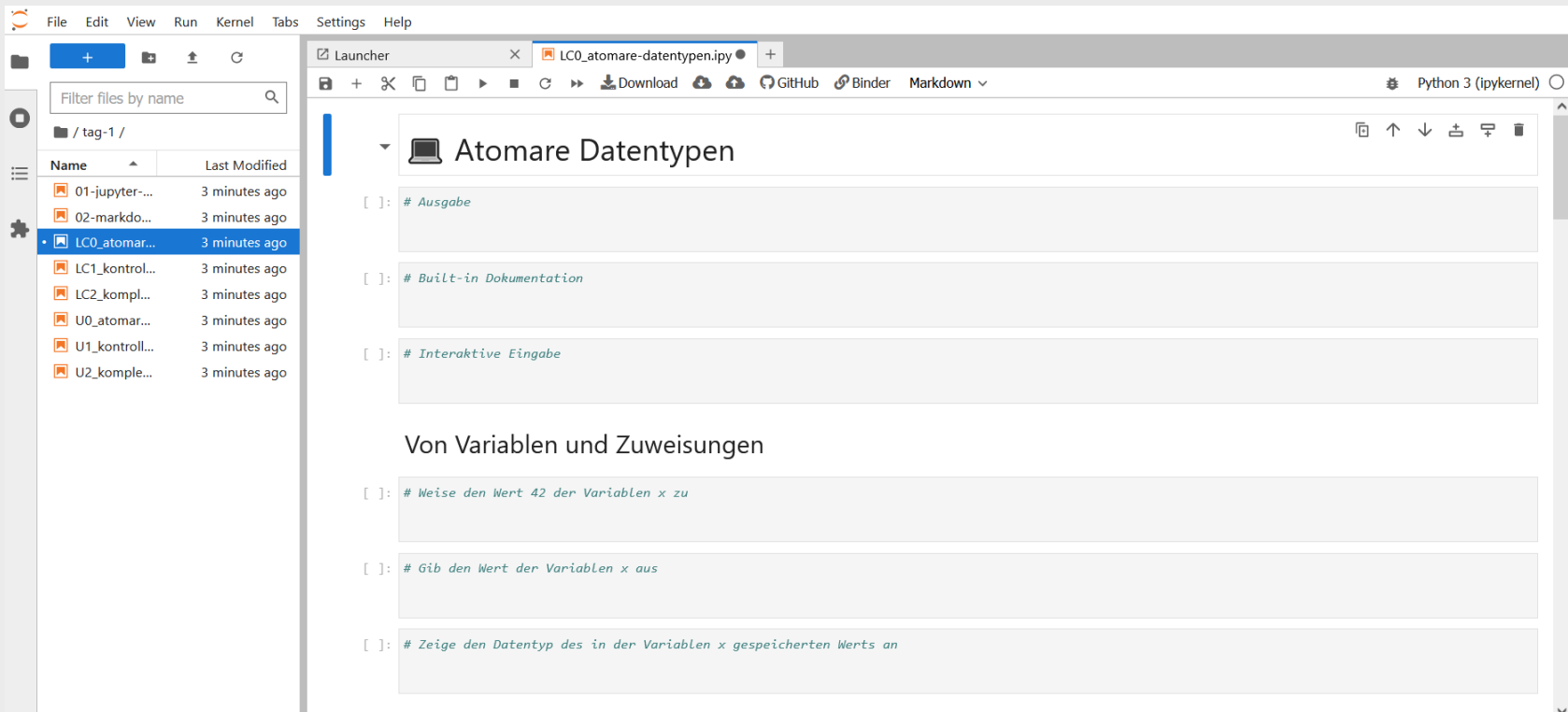
In []:

In []:

In []:

In []:

Jupyter Lab in Binder



The screenshot shows the Jupyter Lab interface in a Binder environment. The left sidebar displays a file explorer with a list of files, including 'LC0_atomar...', which is selected. The main area shows the 'Atomare Datentypen' notebook, which contains several code cells with comments in German.

File Explorer (Left Sidebar):

| Name | Last Modified |
|----------------|---------------|
| 01-jupyter... | 3 minutes ago |
| 02-markdo... | 3 minutes ago |
| LC0_atomar... | 3 minutes ago |
| LC1_kontrol... | 3 minutes ago |
| LC2_kompl... | 3 minutes ago |
| U0_atomar... | 3 minutes ago |
| U1_kontroll... | 3 minutes ago |
| U2_komple... | 3 minutes ago |

Notebook Content (Main Area):

Atomare Datentypen

```
[ ]: # Ausgabe
```

```
[ ]: # Built-in Dokumentation
```

```
[ ]: # Interaktive Eingabe
```

Von Variablen und Zuweisungen

```
[ ]: # Weise den Wert 42 der Variablen x zu
```

```
[ ]: # Gib den Wert der Variablen x aus
```

```
[ ]: # Zeige den Datentyp des in der Variablen x gespeicherten Werts an
```

Wichtiger Hinweis!



[Binder Usage Guidelines](#)

Laden Sie regelmäßig die von Ihnen bearbeiteten **Notebooks (und Dateien) aus der Binder-Umgebung herunter!** Diese werden nicht dauerhaft gespeichert.

Wenn Ihre Binderinstanz längere Zeit inaktiv war (mehr als **10 Minuten!**), dann wird ihre Session terminiert, alle nicht gesicherten Daten und Notebooks sind dann verloren.

Heruntergeladene Notebooks können hochgeladen und weiter bearbeitet werden.

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ tag-1 /

| Name | Last Modified |
|---------------|---------------|
| 01-jupyter... | 4 minutes ago |
| 02-markdo... | 4 minutes ago |
| LC0_atomar... | 4 minutes ago |

- Open
- Open With
- Open in New Browser Tab
- Rename
- Delete
- Cut
- Copy
- Paste
- Duplicate
- Download
- Shut Down Kernel
- Copy Download Link
- Copy Path
- Copy Shareable Link
- New File
- New Notebook
- New Folder

Atomare Datentypen

Ausgabe

lt-in Dokumentation

eraktive Eingabe

n Variablen und Zuweisungen

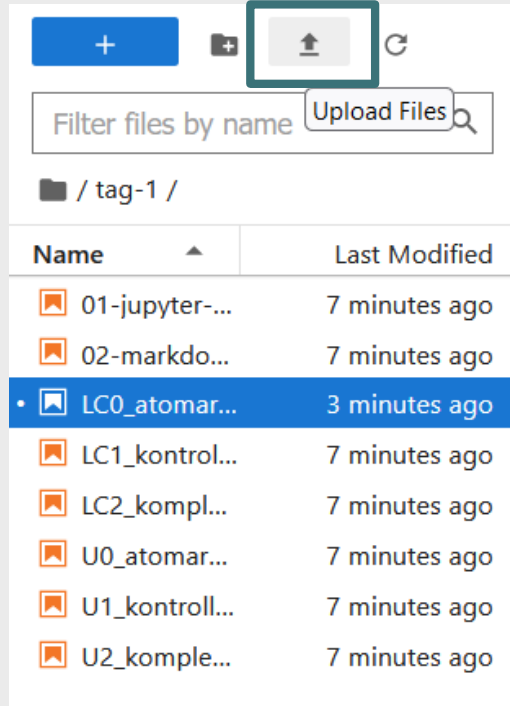
den Wert 42 der Variablen x zu

den Wert der Variablen x aus

ge den Datentyp des in der Variablen x gespeicherten Werts an

Simple Shift+Right Click for Browser Menu 138.59 / 2048.00 MB Mode: Command Ln 1, Col 1 LC0_atomare-datentypen.ipynb 1

Arbeitsstände regelmäßig lokal speichern!



Filter files by name

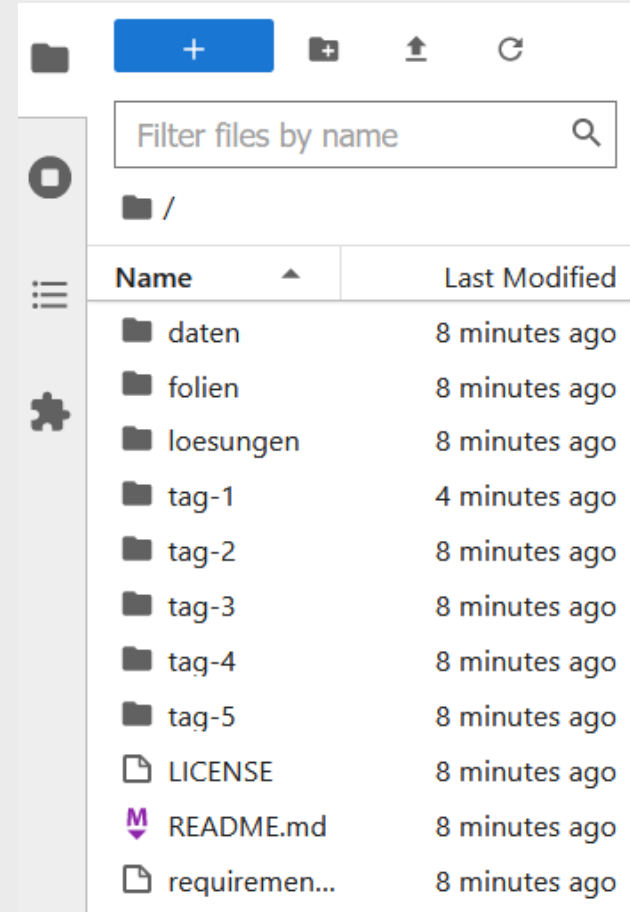
/ tag-1 /

| Name | Last Modified |
|-----------------|---------------|
| 01-jupyter-... | 7 minutes ago |
| 02-markdo... | 7 minutes ago |
| • LC0_atomar... | 3 minutes ago |
| LC1_kontrol... | 7 minutes ago |
| LC2_kompl... | 7 minutes ago |
| U0_atomar... | 7 minutes ago |
| U1_kontroll... | 7 minutes ago |
| U2_komple... | 7 minutes ago |

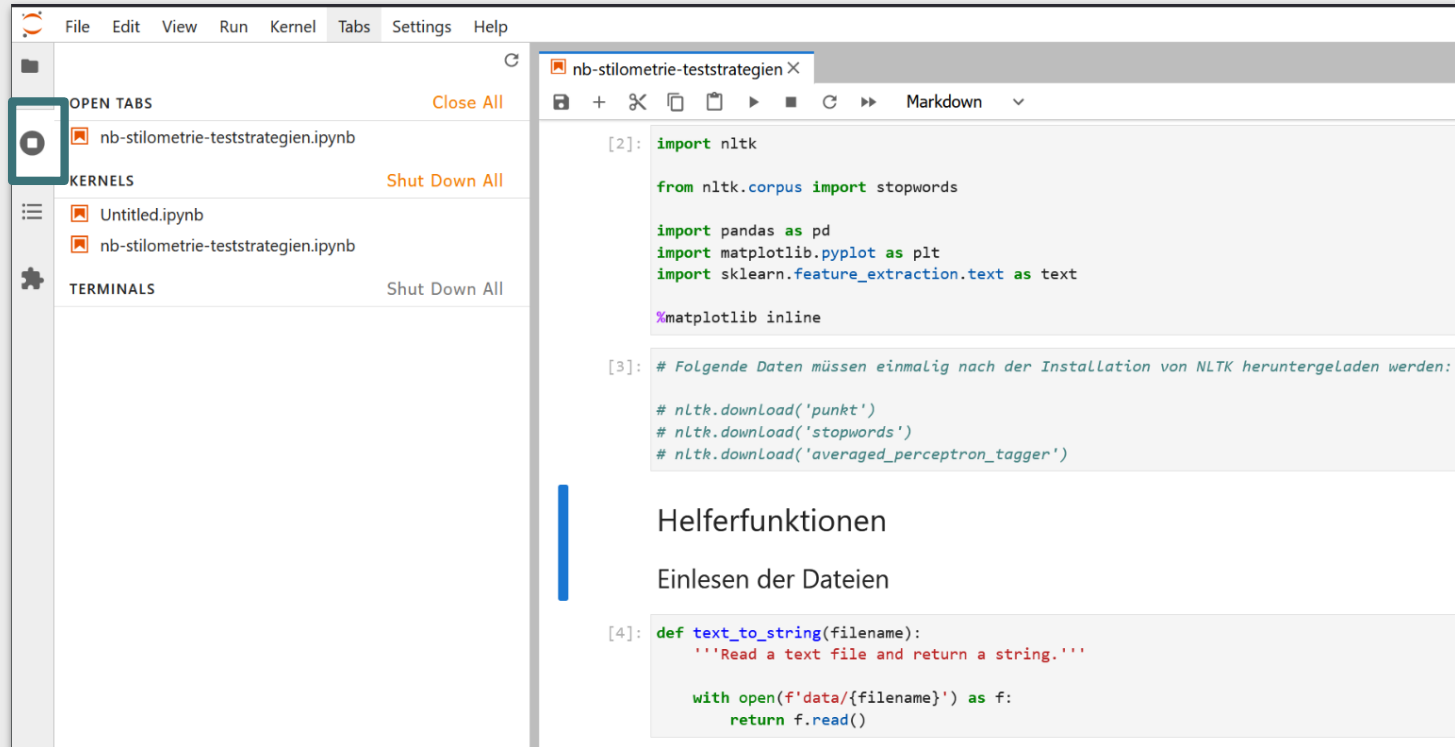
Heruntergeladene Arbeitsstände können wieder hochgeladen und weiter bearbeitet werden!

Dateibrowser

- Workshop-Materialien lassen sich über diese Struktur ansteuern
- Empfehlenswert: Für Arbeit mit Dateien im Laufe der Woche, mit Ordnerstruktur (insb. Hierarchie) vertraut machen



Tabs und Kernels: Sinnvoll – zwischendurch aufräumen, um wieder Arbeitsspeicher freizugeben



The screenshot shows the JupyterLab interface. The left sidebar contains three panels: 'OPEN TABS', 'Kernels', and 'TERMINALS'. The 'Kernels' panel is highlighted with a red box. It shows a list of kernels, including 'nb-stilometrie-teststrategien.ipynb'. The main area displays the code editor for the selected kernel, showing Python code for importing NLTK, pandas, matplotlib, and sklearn, and a function definition for reading text files.

File Edit View Run Kernel Tabs Settings Help

OPEN TABS Close All

- nb-stilometrie-teststrategien.ipynb

KERNELS Shut Down All

- Untitled.ipynb
- nb-stilometrie-teststrategien.ipynb

TERMINALS Shut Down All

nb-stilometrie-teststrategien X

Markdown

```
[2]: import nltk

from nltk.corpus import stopwords

import pandas as pd
import matplotlib.pyplot as plt
import sklearn.feature_extraction.text as text

%matplotlib inline

[3]: # Folgende Daten müssen einmalig nach der Installation von NLTK heruntergeladen werden:

# nltk.download('punkt')
# nltk.download('stopwords')
# nltk.download('averaged_perceptron_tagger')

Helferfunktionen

Einlesen der Dateien

[4]: def text_to_string(filename):
    '''Read a text file and return a string.'''

    with open(f'data/{filename}') as f:
        return f.read()
```

Gliederungsansicht

The screenshot displays a Jupyter Notebook interface. On the left, the 'Gliederungsansicht' (Table of Contents) is visible, listing the notebook's structure. The main area on the right shows the notebook content, which includes code cells and a section titled 'Helferfunktionen'.

Table of Contents (Left Panel):

- Stilometrie: basale Teststrategien und grundlegende Ansätze
- Importe
- Helferfunktionen
- Einlesen der Dateien
 - Erstellen eines Dictionaries mit tokenisierten Wörtern
 - Kürzesten Corpus finden
 - Laden der Texte in ein Dictionary
 - Tokenisierung und Bestimmung des kürzesten Corpus
 - Anwenden der Teststrategien
 - Häufigkeiten der Wortlängen
 - Häufigkeiten der Stoppwörter
 - Häufigkeiten der Wortarten (Part-of-Speech)
 - Vergleich der am häufigsten verwendeten Wörter
 - Berechnung der Jaccard-Metrik

Notebook Content (Right Panel):

The notebook is titled 'nb-stilometrie-teststrategien'. It contains the following code cells:

```
[2]: import nltk

from nltk.corpus import stopwords

import pandas as pd
import matplotlib.pyplot as plt
import sklearn.feature_extraction.text as text

%matplotlib inline
```

[3]: # Folgende Daten müssen einmalig nach der Installation von NLTK heruntergeladen werden:

```
# nltk.download('punkt')
# nltk.download('stopwords')
# nltk.download('averaged_perceptron_tagger')
```

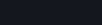
Helferfunktionen

Einlesen der Dateien

```
[4]: def text_to_string(filename):
      '''Read a text file and return a string.'''

      with open(f'data/{filename}') as f:
          return f.read()
```

JupyterLab Documentation

 jupyterlab
stable ▾
Get Started
User Guide > The...
Develop Extensions
Contribute
Privacy policies
🔍 Search

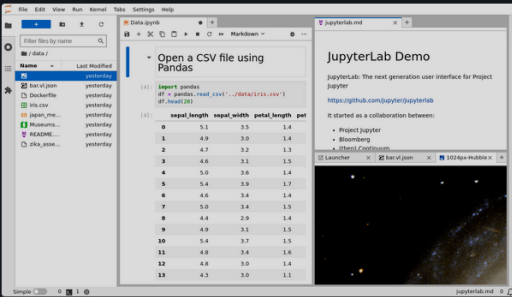
Section Navigation

- Announcements
- The JupyterLab Interface
- JupyterLab URLs
- Working with Files
- Text Editor
- Notebooks
- Code Consoles
- Completer
- Terminals
- Managing Codes and Terminals
- Commands
- Documents and Kernels
- File and Output Formats
- Debugger
- Table Of Contents
- Extensions
- JupyterLab on JupyterHub
- Exporting Notebooks
- Localization and language
- Real Time Collaboration
- Language Server Protocol support
- Interface Customization
- Advanced Usage
- JupyterLab on Binder

The JupyterLab Interface

JupyterLab provides flexible building blocks for interactive, exploratory computing. While JupyterLab has many features found in traditional integrated development environments (IDEs), it remains focused on interactive, exploratory computing.

The JupyterLab interface consists of a [main work area](#) containing tabs of documents and activities, a collapsible [left sidebar](#), and a [menu bar](#). The left sidebar contains a [file browser](#), the [list of running kernels and terminals](#), the [command palette](#), the [notebook cell tools inspector](#), and the [tabs list](#).



☰ On this page

- Menu Bar
- Left and Right Sidebar
- Main Work Area
- Tabs and Simple Interface Mode
- Searching
- Context Menus
- Keyboard Shortcuts

[Edit on GitHub](#)

[Show Source](#)

<https://jupyterlab.readthedocs.io/en/stable/user/interface.html>



Haben Sie Fragen?

Let's code!

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

<https://github.com/Digital-History-Berlin/python-workshop-2024>



Lerntagebuch



5-15 Minuten

Workshop-Lerntagebuch: „Python für Geisteswissenschaftler:innen“

Name:

Datum:

Was habe ich heute gelernt?

Welchen Aha-Moment hatte ich heute?