

# The ABC of Computational Text Analysis

## #9 INTRODUCTION TO PYTHON

Alex Flückiger

Faculty of Humanities and Social Sciences  
University of Lucerne

5 May 2022

# Recap last Lecture

- from words to embeddings
  - recontextualized word meaning
- data-driven NLP is both powerful and biased
- data is never raw but depends on many decisions

# Outline

- enter the shiny world of Python 😎
  - programming basics
  - development editor
- think about mini-project



Python

# Python is ...

# a programming language that is ...

- general-purpose
    - not specific to any domain
  - interpreted
    - no compiling
  - standard language in data science



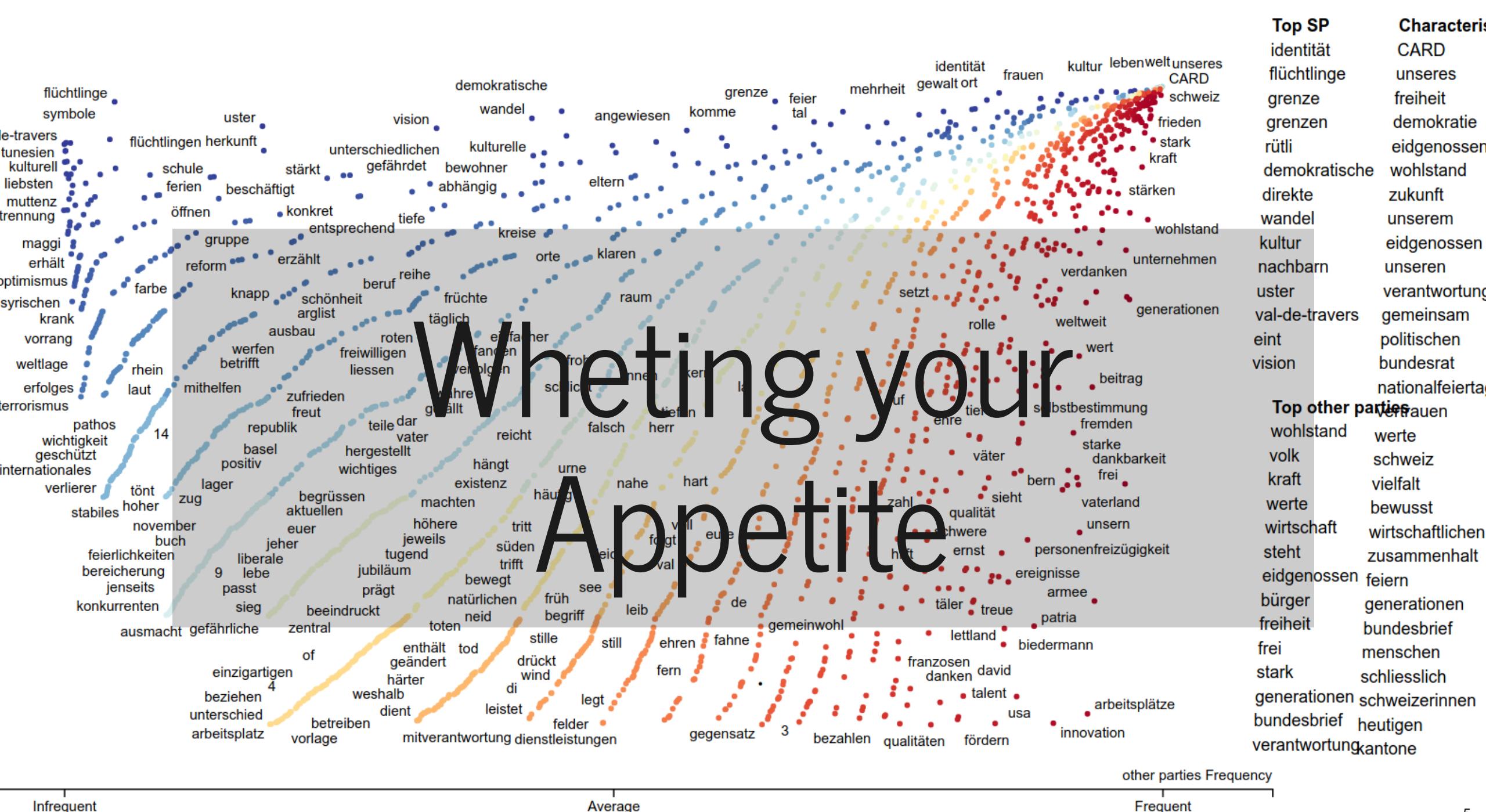
# Popular programming languages [src](#)

# How to learn programming?

## Three inconvenient truths

- programming cannot be learnt in a course
  - I try to make the start as easy as possible!
- frustration is part of learning
  - fight your way!
- the Python ecosystem is huge
  - grow skills by step-by-step

Programming can be absolutely captivating! 



# Programming Concepts and Python Syntax

# Variables

Variables are kind of storage boxes

```
# define variables
x = "at your service"
y = 2
z = ", most of the time."

# combine variables
int_combo = y * y      # for numbers any mathematical operation
str_combo = x + z       # for text only concatenation with +
                        # show content of variable
print(str_combo)
```

# Data Types

The type defines the object's properties

Name	What for?	Type	Examples
String	Text	str	"Hi!"
Integer, Float	Numbers	int, float	20, 4.5
Boolean	Truth values	bool	True, False
:	:	:	:
List	List of items (ordered, mutable)	list	["Good", "Afternoon", "Everybody"]
Tuple	List of items (ordered, immutable)	tuple	(1, 2)
Dictionary	Relations of items (unordered, mutable)	dict	{"a":1, "b": 2, "c": 3}

# Data Type Conversion

Combine variables of the same type only

```
# check the type
type(YOUR_VARIABLE)

# convert types (similar for other types)
int('100') # convert to integer
str(100)   # convert to string

# easiest way to use a number in a text
x = 3
mixed = f"x has the value: {x}"
print(mixed)
```

# Confusing Equal-Sign

= vs. == contradicts the intuition

```
# assign a value to a variable
x = 1
word = "Test"

# compare two values if they are identical
1 == 2           # False
word == "Test"    # True
```

# Comments

- lines ignored by Python
- write comments, it helps you ...
  - to learn initially
  - to understand later

```
# single line comment

"""

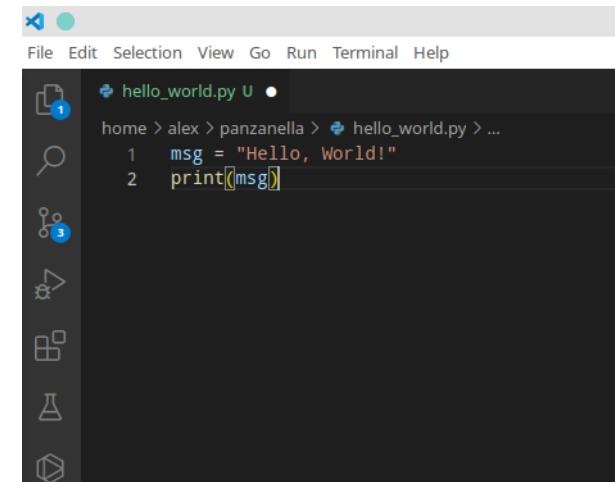
comment across
multiple
lines
"""

```

# Visual Studio Code

The (best) editor to program in Python

- integrated development environment (IDE)
  - interactive development
  - similar to RStudio
- 3 views in editor
  - programming (left)
  - output (right)
  - additional information (bottom)
- use `tab` for autocomplete



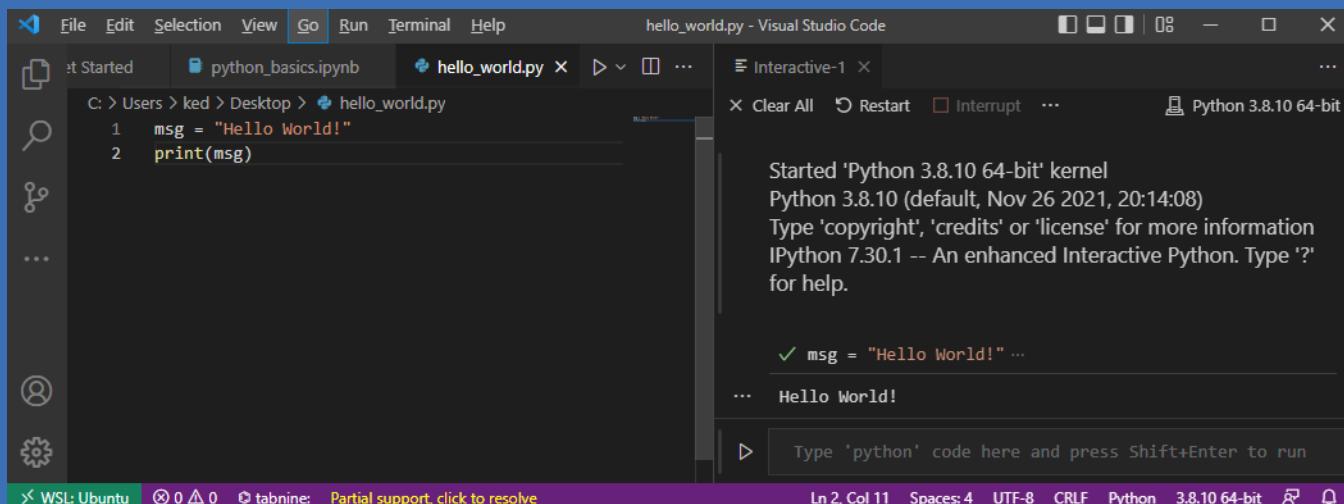
# In-class: Run your first Python Program I

1. Make sure that your local copy of the Github repository KED2022 is up-to-date with `git pull`.
2. Open the Visual Studio Editor.
3. Windows User only: Make sure that you are connected to `WSL: Ubuntu` (green badge lower-left corner, see image on the next slide). If not, click on the badge and select `New WSL Window`.
4. Create a new file with the following content, save it as `hello_world.py`. Then, execute it by a right click on the code and select `Run current file in interactive window`.

```
# print out a message
msg = "Hello World!"
print(msg)
```

5. Does the output looks like the screenshot on the next slide? If the execution doesn't work as expected, ask me or your neighbour. There might be a technical issue.

# In-class: Run your first Python Program!



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** Shows two tabs: "pythonBasics.ipynb" and "hello\_world.py". The "hello\_world.py" tab contains the following code:

```
C: > Users > ked > Desktop > hello_world.py
1 msg = "Hello World!"
2 print(msg)
```
- Terminal:** An "Interactive-1" terminal window is open, showing the output of the Python code:

```
Started 'Python 3.8.10 64-bit' kernel
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.30.1 -- An enhanced Interactive Python. Type "?" for help.

✓ msg = "Hello World!" ...
...
Hello World!
```
- Status Bar:** WSL: Ubuntu, 0 ▲ 0, tabnine: Partial support, click to resolve, Ln 2, Col 11, Spaces: 4, UTF-8, CRLF, Python 3.8.10 64-bit.

# Iterations

## for-loop

do something with each element of a collection

```
sentence = ['This', 'is', 'a', 'sentence']

# iterate over each element
for token in sentence:

    # do something with the element
    print(token)
```

# Conditionals

## if-else statement

condition action on variable content

```
sentence = ['This', 'is', 'a', 'sentence']

if len(sentence) < 3:
    print('This sentence is shorter than 3 tokens')
elif len(sentence) == 3:
    print('This sentence has exactly 3 tokens')
else:
    print('This sentence is longer than 3 tokens')
```

# Indentation matters!

- intend code within code blocks  
loops, if-statements etc.
- press `tab` to intend



```
if 5 > 2:  
    print('5 is greater than 2')
```



```
if 5 > 2:  
    print('5 is greater than 2')
```

# Methods

Do somethin with an object

```
sentence = 'This is a sentence'

# split at whitespace
tokens = sentence.split(' ')

# check the variables
print(sentence, type(sentence), tokens, type(tokens))

# add something to a list
tokens.append('.')

# concatenate elements to string
tokens = ' '.join(tokens)
print(tokens, type(tokens))
```

# Functions and Arguments

## DRY: Don't Repeat Yourself

- functions have a name and optional arguments

```
function_name(arg1, ..., argn)
```

```
# define a new function
def get_word_properties(word):
    """
    My first function to print word properties.
    It takes any string as argument (variable: word).
    """

# print(), len() and sorted() work also as functions
length = len(word)
sorted_letters = sorted(word, reverse=True)
print(word, 'length:', length, 'letters:', sorted_letters)
```

# Indexing

Computers start counting from zero! 😱

```
sentence = ['This', 'is', 'a', 'sentence']

# element at position X
first_tok = sentence[0]      # 'This'

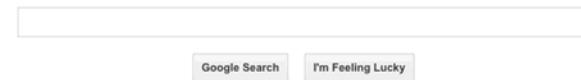
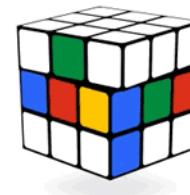
# elements of subsequence [start:end]
sub_seq = sentence[0:3]       # ['This', 'is', 'a']

# elements of subsequence backwards
sub_seq_back = sentence[-2:]    # ['a', 'sentence']
```

# Errors

A myriad of things can go wrong

1. read the message
2. find the source of the error  
script name + line number
3. paste message into Google



Play with more cubes at [Chrome Cube Lab](#)

*Learning by doing, doing by googling*

# Modules/Packages

No programming from scratch 

- packages provide specific functionalities
- packages need to be installed first

# NLP Packages

- **spaCy**

industrial-strength Natural Language Processing (NLP)

- **textaCy**

NLP, before and after spaCy

- **scattertext**

beautiful visualizations of how language differs across corpora

# Mini-Project

present project on 2 June 2022

- analyze any collection of documents
- apply quantitative measures + interpretation
  - compare historically
  - compare between actors
- form groups of 2-4 people

# In-class: Exercises I

1. Open the script with the basics of Python in your Visual Studio Editor:

`materials/code/python_basics.ipynb`

2. Try to understand the code in each cell and run them by clicking the play symbol left to them. Check the output. Modify some code as well as data and see how the output changes. Initially, this try-and-error is good strategy to learn. Some ideas:

Combine a string and an integer variable without converting it. What error do you get? How can you avoid it?

Select `is_a` from the list using the right index.

# In-class: Exercises II

## 1. Write a Python script that

takes text (a string)

splits it into words (a list)

iterates over all the tokens and print all tokens that are longer than 5 characters

Bonus: wrap your code in a function.

## 2. Go to the next slide. Start with some of the great interactive exercises out there in the web.

# Resources

learn basics interactively

- Python Principles
- LearnPython

official Python introduction

- Python introduction



Questions?

