

Python Programming for Linguists

Ingo Kleiber, 2020

Learning Objectives

After completing this workshop, you will be able to ...

1. describe what programming essentially is about.
2. name and describe some basic programming terminology.
3. model simple problems in terms of data structures and basic algorithms.
4. write basic scripts in Python in order to solve specific problems.
5. utilize third-party libraries such as *NLTK*, *spaCy*, and *TextDirectory*.
6. construct and apply basic regular expressions.
7. utilize Python for text manipulation.
8. utilize Python to perform concordance and frequency analysis.
9. automatically annotate texts (PoS, Universal Dependencies, NER) using spaCy.
10. scrape web data in order to build corpora (Web as Corpus) using Python.
11. compute basic statistics using Python.

Workshop Outline

Video Python Programming for Absolute Beginners

Exercises 1 – 3

Video The Pizza Problem

Exercises 4 – 5

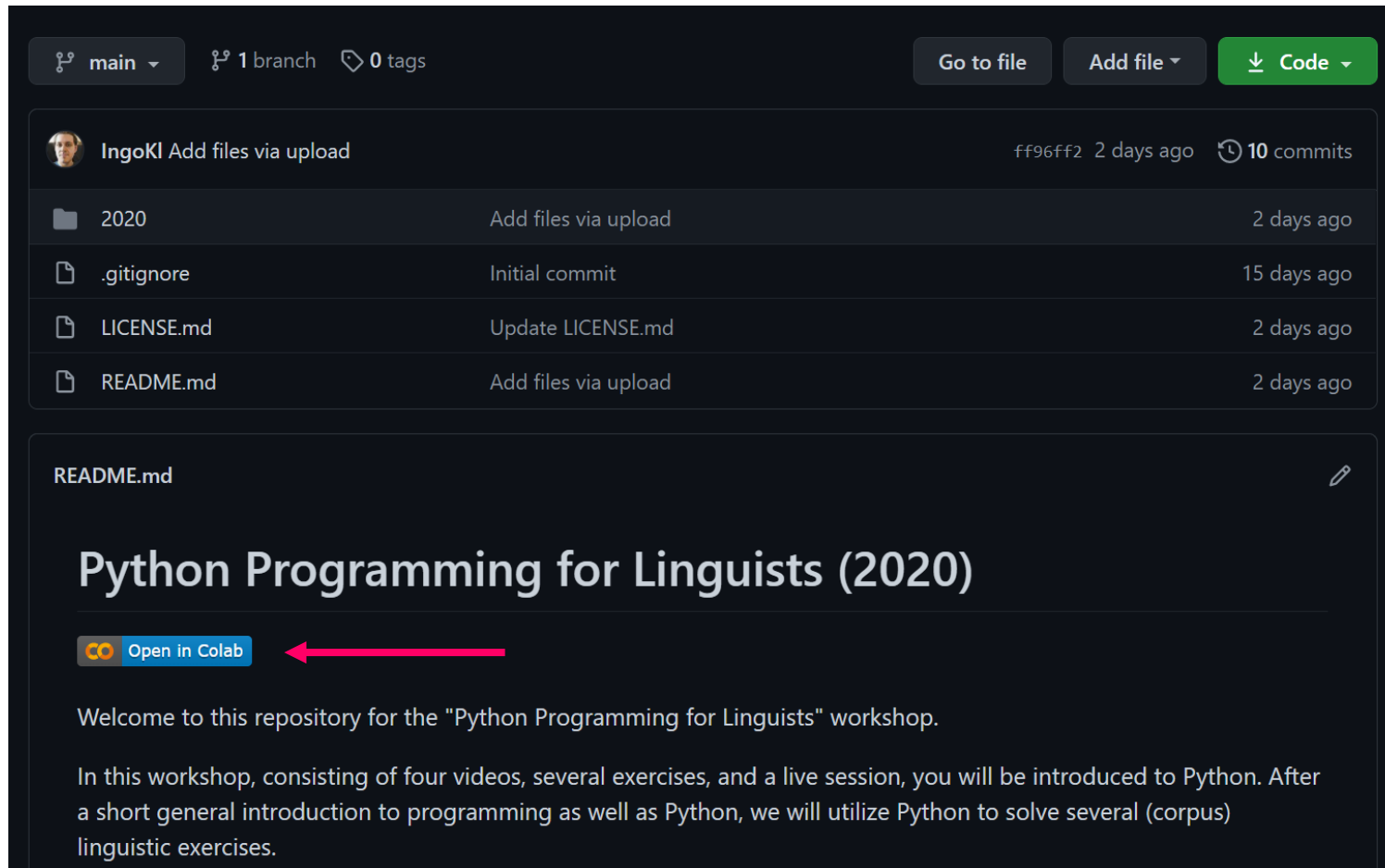
Video Working with Files, Texts, and Regular Expressions

Exercises 6 – 7

Live Session (*or Recording*) – Exercises 8 – 16

Video Summary and Resources

Coding Along



The screenshot shows a GitHub repository interface. At the top, there are navigation elements: a dropdown for the 'main' branch, a link to '1 branch', and a link to '0 tags'. To the right are buttons for 'Go to file', 'Add file', and a green 'Code' button with a download icon. Below this is a commit history table. The table has three columns: a profile picture and username, a commit message, and a timestamp. The first commit is by 'IngoKI' with the message 'Add files via upload' and a timestamp of '2 days ago'. Below the commit history is a section for the 'README.md' file. The file content starts with the title 'Python Programming for Linguists (2020)' in a large font. Below the title is a button that says 'Open in Colab' with a red arrow pointing to it. The text below the button reads: 'Welcome to this repository for the "Python Programming for Linguists" workshop. In this workshop, consisting of four videos, several exercises, and a live session, you will be introduced to Python. After a short general introduction to programming as well as Python, we will utilize Python to solve several (corpus) linguistic exercises.'

Profile Picture	Commit Message	Timestamp
IngoKI	Add files via upload	2 days ago
	Initial commit	15 days ago
	Update LICENSE.md	2 days ago
	Add files via upload	2 days ago

Python Programming for Linguists (2020)

[Open in Colab](#)

Welcome to this repository for the "Python Programming for Linguists" workshop.

In this workshop, consisting of four videos, several exercises, and a live session, you will be introduced to Python. After a short general introduction to programming as well as Python, we will utilize Python to solve several (corpus) linguistic exercises.

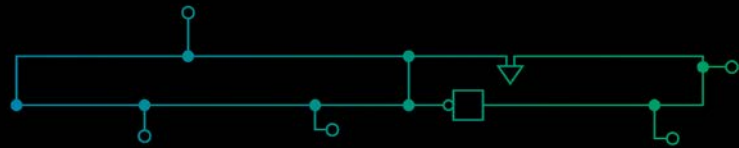
You can find all videos, materials, and exercises on ***GitHub***.

Disclaimer

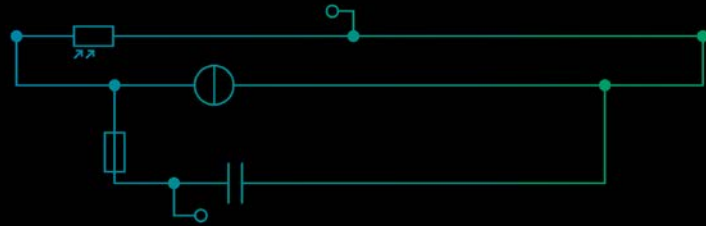
Everything that follows should be considered a
(gross) oversimplification of reality!

This Workshop

This workshop is heavily inspired by a number of workshops I held at 35c3 and 36c3.



(PYTHON) PROGRAMMING FOR
ABSOLUTE BEGINNERS



PYTHON PROGRAMMING FOR
ABSOLUTE BEGINNERS

36c3

00

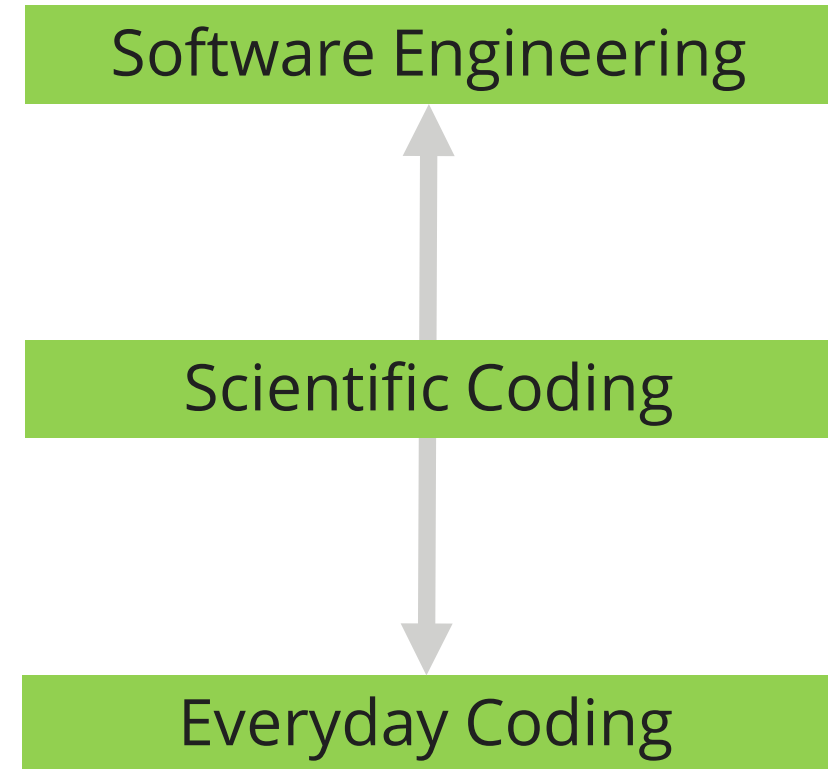
Python Programming for Absolute Beginners

Python Programming for Linguists

Ingo Kleiber, 2020

Programming is ...

- instructing machines and computers
- problem solving
- thinking differently (computationally)
- modeling problems and other things
- an art
- science
- fun
- ...

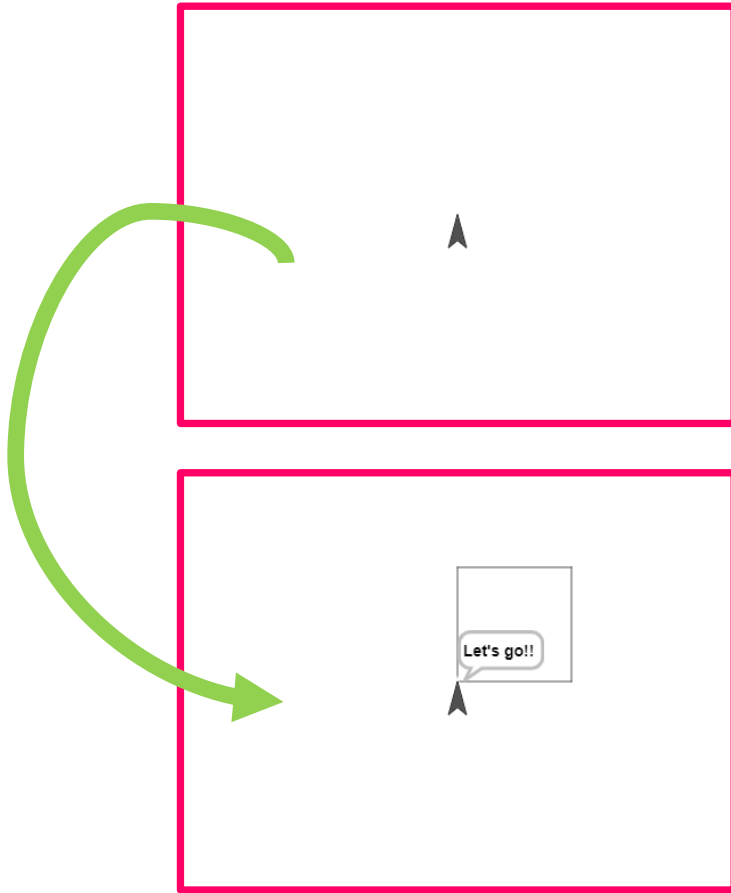


Programming

“It’s difficult not to have a love/hate relationship with computer programming if you have any relationship with it at all.”

(Rosenberg 2006)

What Does Code Look Like? *Snap!*



Commands

Control Structure (*repeat*)

Python

Python is one of hundreds of programming languages.

- free, open, and available on almost any platform
- modern and widely used; there is a great community
- relatively easy to learn; hard to master
- legacy Python (2.x) vs. **modern Python** (3.x)


R



“R is a language and environment for
statistical computing and **graphics.**”
(r-project.org)

What Does Code Look Like? *Python*

Lines and Blocks



```
1  print('Counting up to ten!')
2
3  for i in range(10):
4      |    print(f'{i + 1} out of 10')
5
6  print('Finished!')
```

Command

calling a function print with an argument

= One command

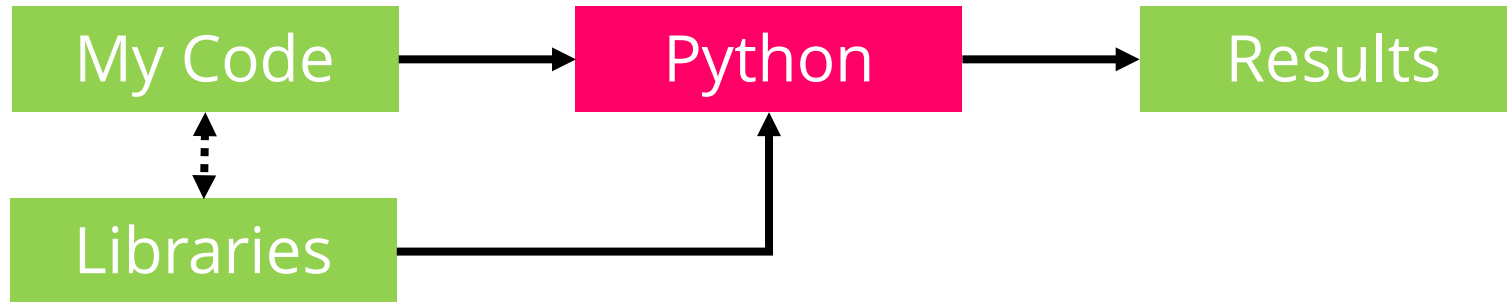
Block / Control Structure

for loop

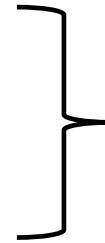
= Unit of functionality

Executed from top to bottom – line by line

Executing Python Code



1. Script (e.g., helloworld.py)
2. Interactive (*IPython*)
3. Notebook
4. ...

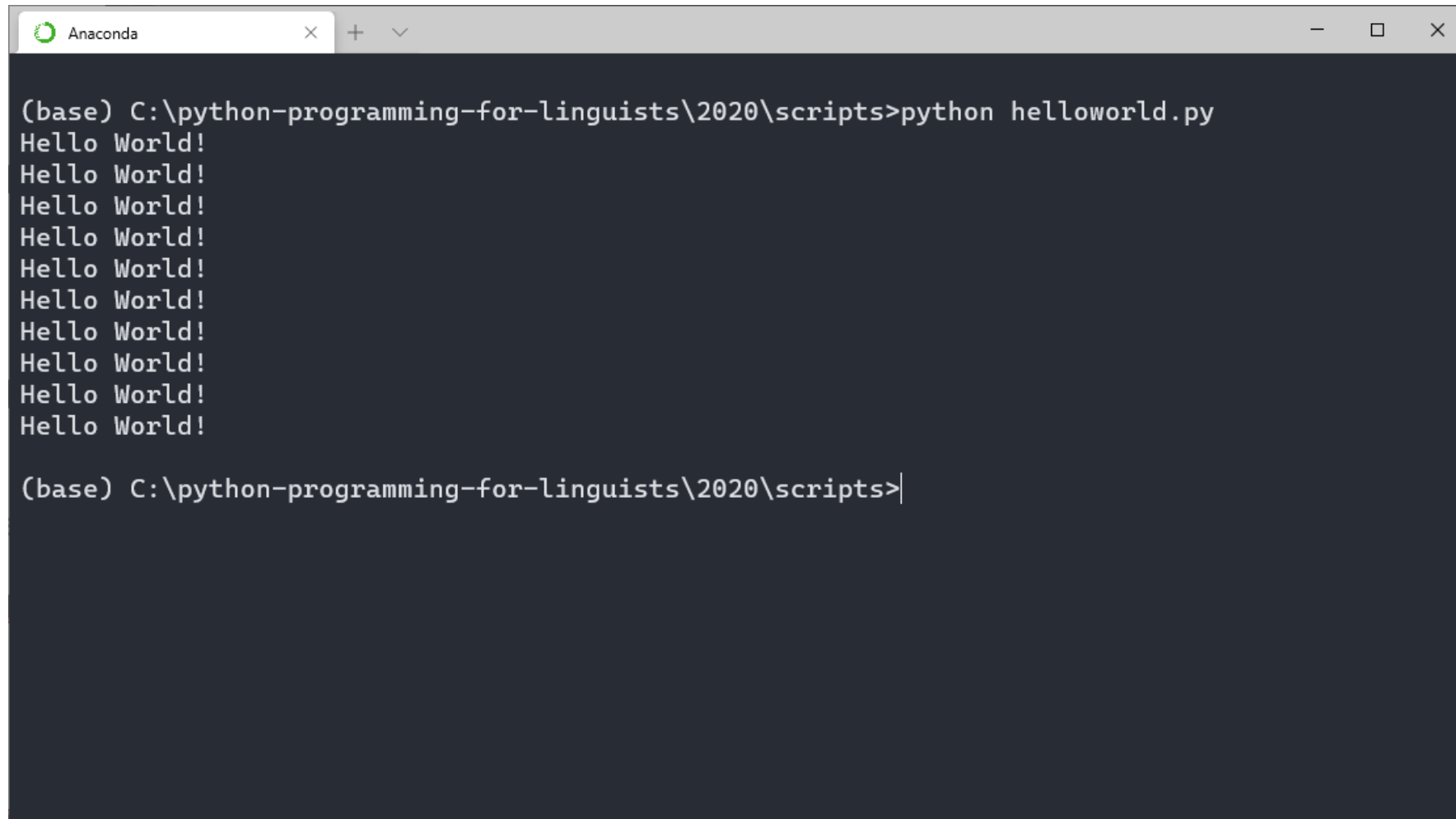


You will need Python on your system. There are various versions and distributions. For science, I would strongly recommend *Anaconda*.

Script

Run the script *helloworld.py* using Python.

python *heLLoworld.py*

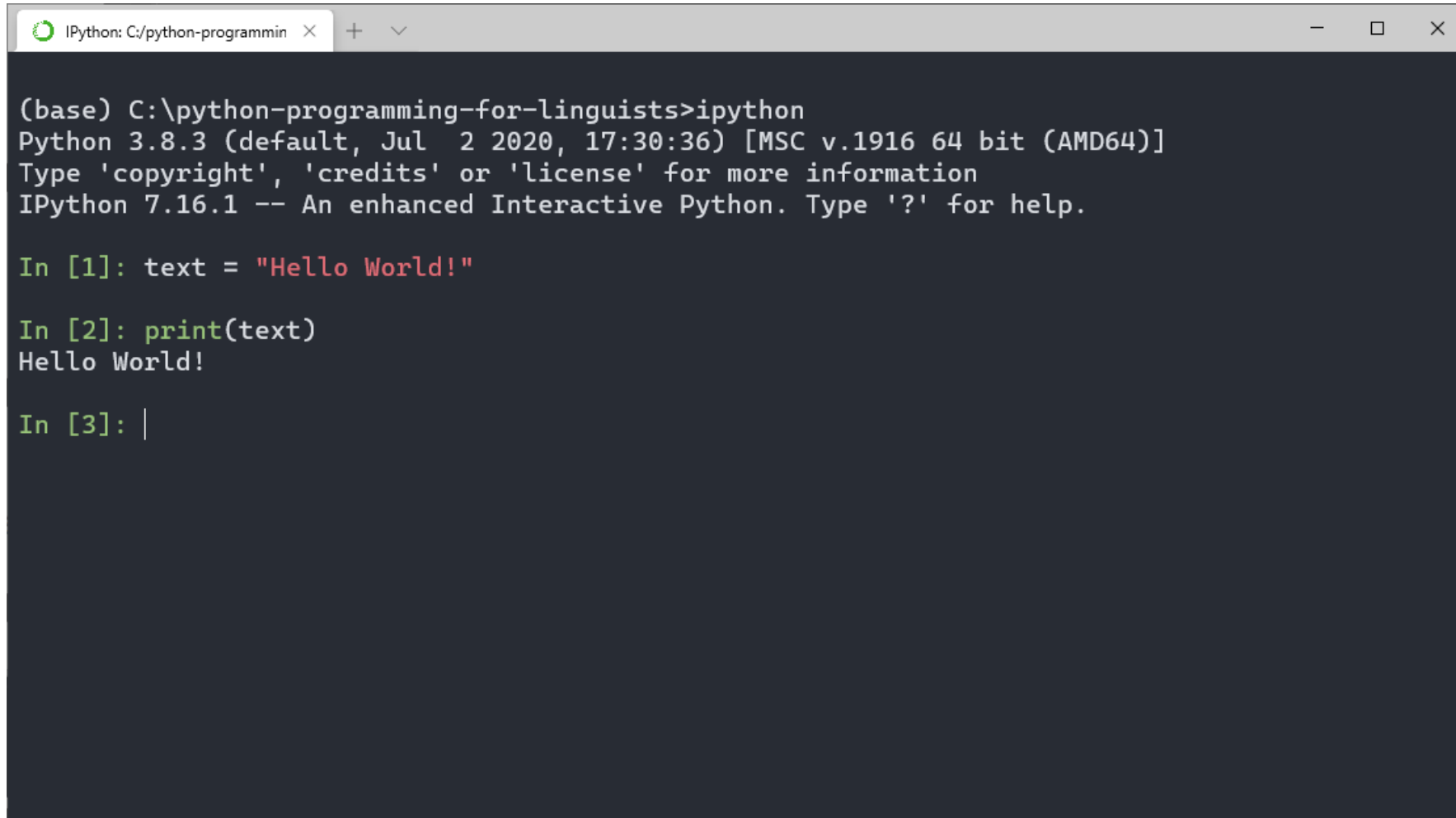
A screenshot of an Anaconda terminal window. The window has a title bar with the Anaconda logo and the text "Anaconda". The terminal content shows a command prompt where the user has entered the command to run a Python script. The output of the script is displayed as ten lines of "Hello World!".

```
(base) C:\python-programming-for-linguists\2020\scripts>python helloworld.py
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!

(base) C:\python-programming-for-linguists\2020\scripts>
```

Interactive (using *IPython*)

ipython



```
IPython: C:/python-programmin x + v - □ x
(base) C:\python-programming-for-linguists>ipython
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.16.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: text = "Hello World!"

In [2]: print(text)
Hello World!

In [3]: |
```


Our Tools

- *Git* and *GitHub* in particular
- *Jupyter Notebook* and *Google Colab*

Have a look at *Setting Up Your Development Environment*

If you have a Google account, you can run the exercises and notebooks directly via *Google Colab*.

If you do not want to rely on Google, you can install *Jupyter* locally and run the exercises this way.



Basics – The Building Blocks

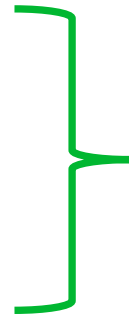
1. **Variables** = a container to put data in (`r = 13`)
2. **Lists** = a list of data-things (e.g., variables) (`l = [1, 2, 3]`)
3. **Loops** = repeating something until some condition is met
4. **If-Constructions** = do something if some condition is met
5. **Functions** = a unit of code that completes a specific task
6. **Dictionaries** = a data-thing that contains key: value pairs (`cat = {'age': 2}`)

1. Variables

a = 13

b = 'Hello World'

c = 42.42



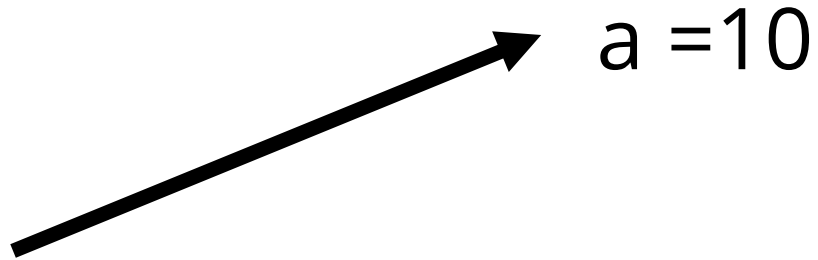
Three **variables** (containers) of three different types:
integer, string, and float

2. Lists

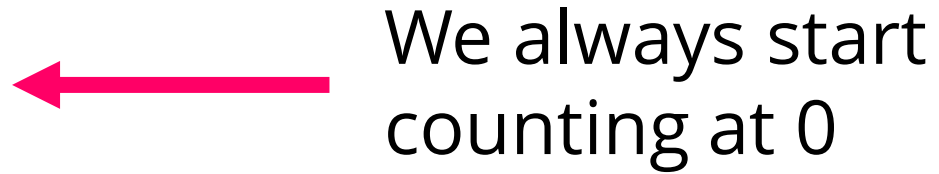
`l = [1, 2, 3, 4, a]`  A list (named `l`) containing 4 integers and the variable `a`.

2. Lists

`l = [1, 2, 3, 4, a]`



`a = 10`



We always start counting at 0

`l[0] → 1`

`l[3] → 4`

2. Lists

la = [1, 2, 3]
lb = [4, 5, 6]

la[1] → 2

lol = [la, lb] → [[1,2,3], [4,5,6]] } A list of lists

↑

lol[0][1] → 2

2. Lists

```
l = [1, 2, 3, 4, 5]
```

```
l.append(0)
```

```
l.append(3)
```

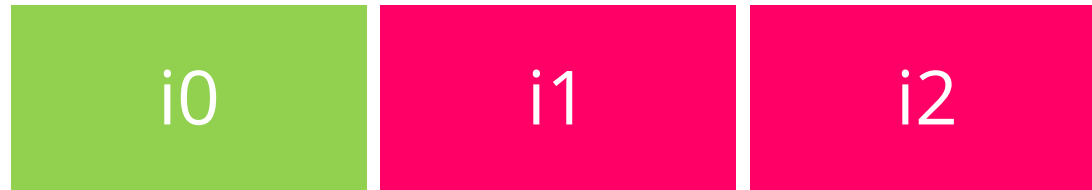
```
l.sort()
```

3. Loops

```
box = ['i0', 'i1', 'i2']  
for item in box:  
    print(item)
```



Step: **0**

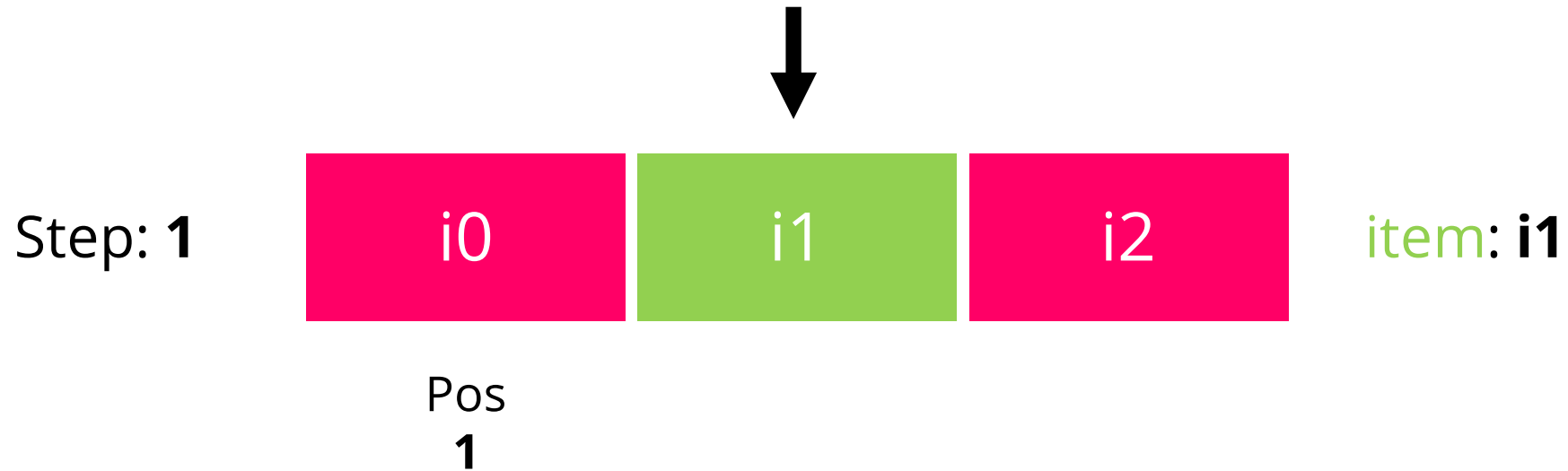


Pos
0

item: **i0**

3. Loops

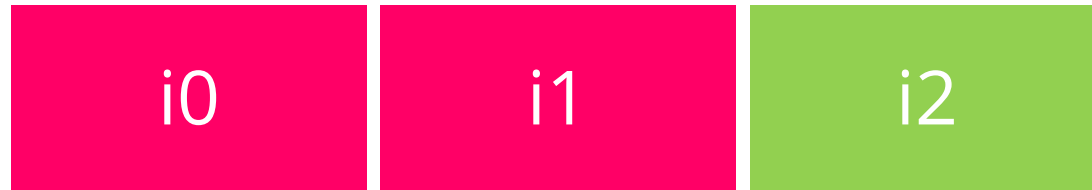
```
box = ['i0', 'i1', 'i2']  
for item in box:  
    print(item)
```



3. Loops

```
box = ['i0', 'i1', 'i2']  
for item in box:  
    print(item)
```

Step: **2**



Pos
2

item: **i2**

4. If-Constructions

```
a = 10
```

```
if a > 15:
```


```
    print ('A is greater than 15')
```

```
else:
```

```
    print ('A is not greater than 15')
```

5. Functions

Two parameters which we pass to the function.



```
def add(a, b):  
    result = a + b
```

```
    return result
```




What the function *returns* after *calling*

add(5, 10) → 15

add(2, 2) → 4

6. Dictionaries

```
word1 = {  
    'lemma': 'cat',  
    'pos': 'noun',  
}
```



Two key: value pairs

word1['pos'] → 'noun'

Works Cited

Rosenberg, Scott. 2006. *Dreaming in Code*. New York: Three Rivers Press.