Data Mining for Computer and Systems Sciences (DAMI)

Lab Session 0:

Introduction to Python

Stockholm University

OUTLINE

1. Introduction to Python
   - Interpreter, Math Operations, Variables, Function Call
2. Data Types and Data Structures
   - Variables, List, Tuples, String, Dictionary
3. Package Management (PIP)
   - numpy, pandas, jupyter, scikit-learn, scipy
   - Working with existing open-source projects
4. Conditionals and Loops with Jupyter
5. Definition of Functions

ILOs:

- Get acquainted with Python interpreter
- Learn basic programming data structures
- Understand Python Package Management
- Solve basic programming tasks using conditionals, loops and functions in Jupyter

NOTE: This lab session is not graded and mainly intended for beginners in programming.

Stockholm University

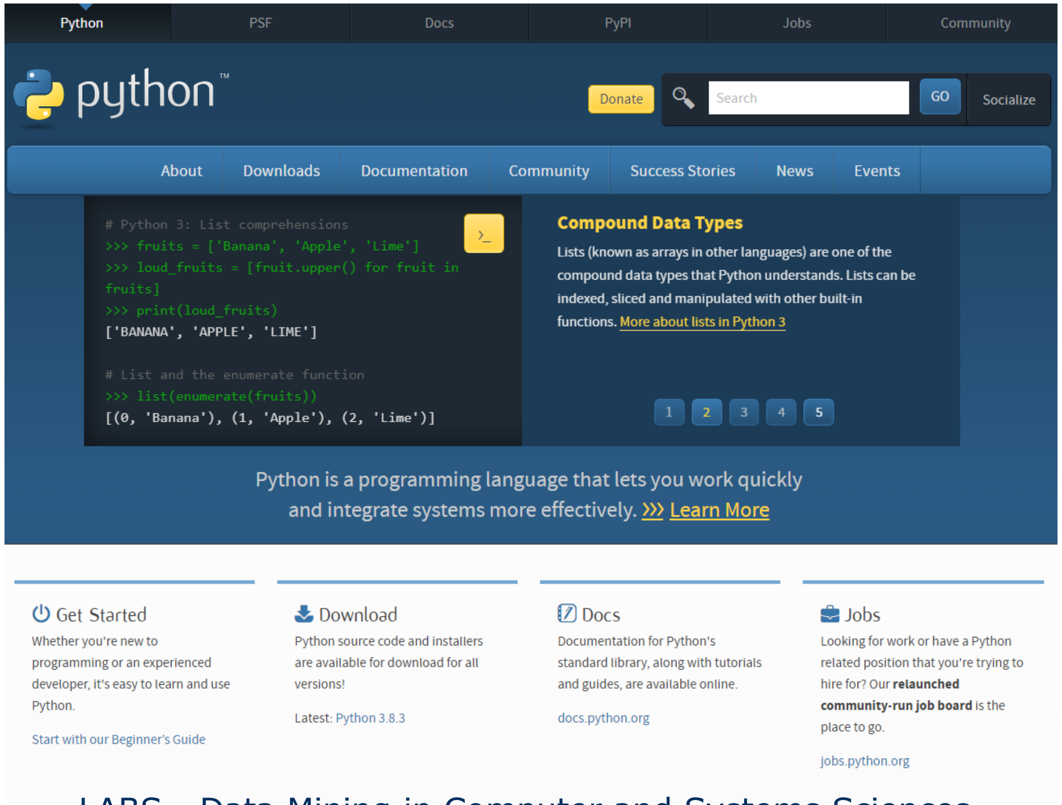
1. INTRODUCTION TO PYTHON

Why Python?

- Web development (servers)
- Software development
- **Rapid prototyping**
- **Scientific Computing**
  - Leverage computing capabilities to create algorithms that solve complex problems: Math models, biological simulations, etc.

Installation

https://www.python.org/



```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. More about lists in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Get Started
Whether you're new to programming or an experienced developer, it's easy to learn and use Python.
Start with our Beginner's Guide

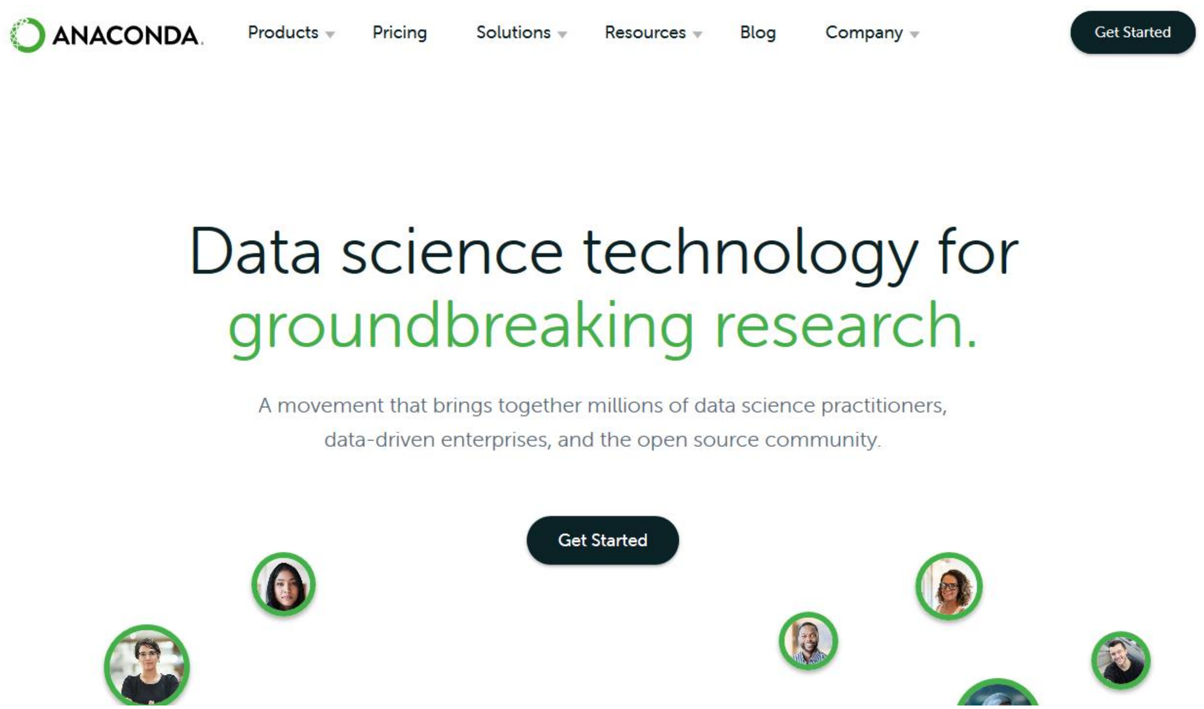
Download
Python source code and installers are available for download for all versions!
Latest: Python 3.8.3

Docs
Documentation for Python's standard library, along with tutorials and guides, are available online.
docs.python.org

Jobs
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.
jobs.python.org

Installation

https://www.anaconda.com/



Data science technology for groundbreaking research.

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

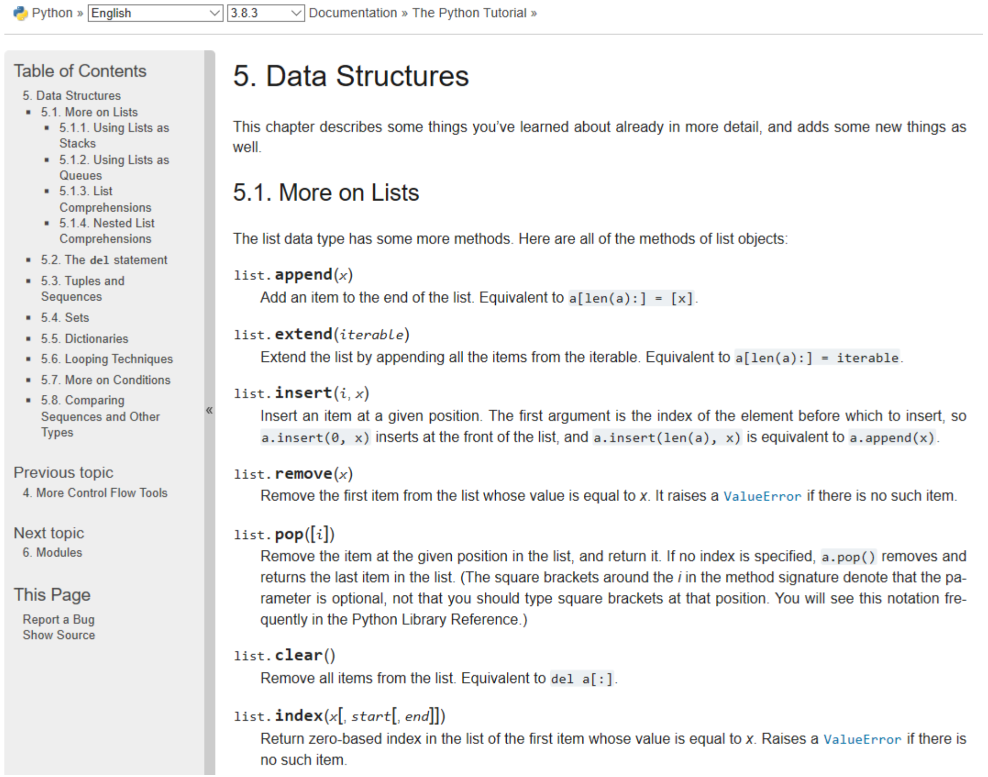
Anaconda

- Includes packages necessary for data science.
- Suitable for data-driven enterprise solutions.
- **Not explained during DAMI labs.**

Stockholm University

Documentation

https://docs.python.org/3/contents.html



Python » English 3.8.3 Documentation » The Python Tutorial »

Table of Contents

Previous topic

4. More Control Flow Tools

Next topic

6. Modules

This Page

Report a Bug
Show Source

5. Data Structures

This chapter describes some things you've learned about already in more detail, and adds some new things as well.

5.1. More on Lists

The list data type has some more methods. Here are all of the methods of list objects:

`list.append(x)`
Add an item to the end of the list. Equivalent to `a[len(a):] = [x]`.

`list.extend(iterable)`
Extend the list by appending all the items from the iterable. Equivalent to `a[len(a):] = iterable`.

`list.insert(i, x)`
Insert an item at a given position. The first argument is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list, and `a.insert(len(a), x)` is equivalent to `a.append(x)`.

`list.remove(x)`
Remove the first item from the list whose value is equal to *x*. It raises a `ValueError` if there is no such item.

`list.pop([i])`
Remove the item at the given position in the list, and return it. If no index is specified, `a.pop()` removes and returns the last item in the list. (The square brackets around the *i* in the method signature denote that the parameter is optional, not that you should type square brackets at that position. You will see this notation frequently in the Python Library Reference.)

`list.clear()`
Remove all items from the list. Equivalent to `del a[:]`.

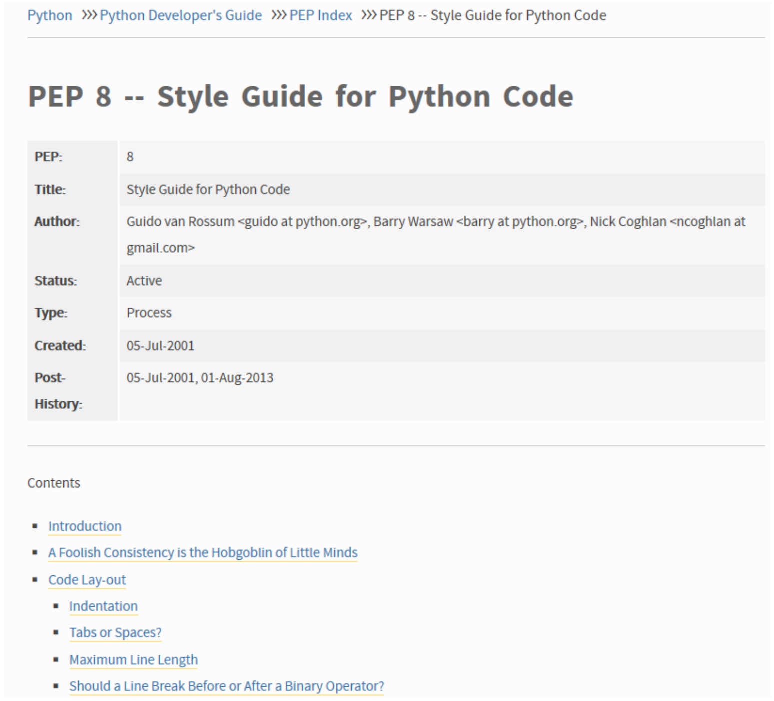
`list.index(x[, start[, end]])`
Return zero-based index in the list of the first item whose value is equal to *x*. Raises a `ValueError` if there is no such item.

- Usually we program having at arm's distance the documentation of the packages we plan to use.

Stockholm University

Style Guide PEP-8

https://www.python.org/dev/peps/pep-0008/



Python ››› Python Developer's Guide ››› PEP Index ››› PEP 8 -- Style Guide for Python Code

PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active
Type:	Process
Created:	05-Jul-2001
Post-History:	05-Jul-2001, 01-Aug-2013

Contents

- Example of good practices:

4 indentation spaces

79 maximum line length

Definition of variables:

```
my_long_variable_with_lowercase = 3
```

Definition of functions

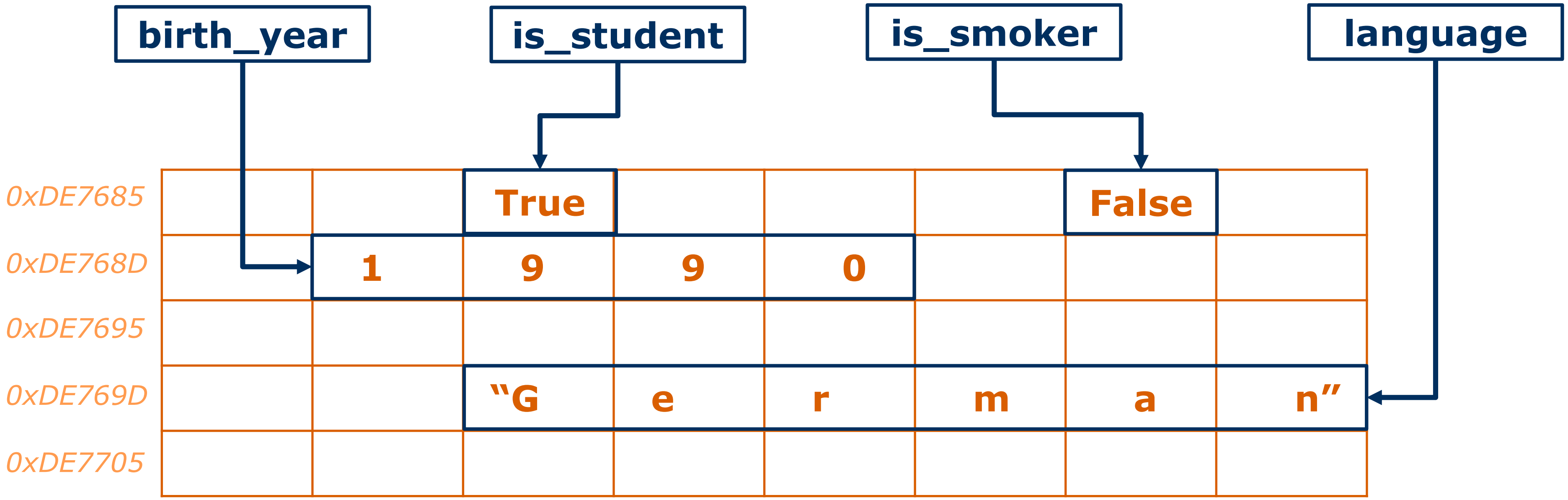
```
def my_custom_function():
```

Stockholm University

2. VARIABLES, DATA TYPES, DATA STRUCTURES

Variables

- Containers for storing data values



Data Representation of Computer’s Memory

Most Common Data Types and Data Structures

• String	Texts
• Bool	True/False
• Integer	Numeric
• Float	Numeric with decimals
• List	Sequence of elements
• Dictionary	Mapping keys-values

Others: Tuples, Sets

Header Block

- Good practice for all codes in Python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# =============================================================================
# Created By  : AUTHOR
# Created Date: DATE
# =============================================================================
"""
DOCSTRING DESCRIBING THE CODE
"""
# =============================================================================
# Imports
# =============================================================================
from ... import ...
# =============================================================================
# CODE
# =============================================================================
<More code…>
```

← Shebang line for Unix-like OS.

← Source code encoding recommended from PEP263.

← Authoring information

← Lines indicating 79 characters to fulfill PEP-8 suggestions.

← Docstring for future automatic documentation of packages

← Separation of where the imports and rest of the code goes.

Example from https://stackoverflow.com/a/51914806

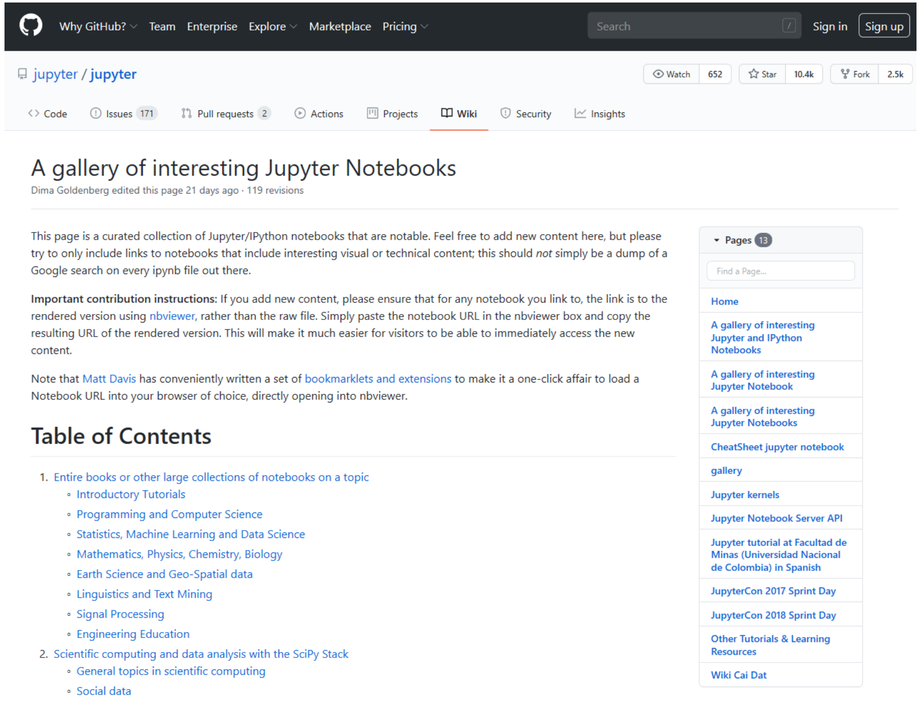
3. PACKAGE MANAGEMENT

Package Manager PIP

- Included by default in Python since v3.4
- Makes easier to download and reuse code (wrapped as packages) from other developers in your own project.
- The list of available packages that can be downloaded through PIP is found on **https://pypi.org/**
- To install packages, use the console command:

```
›  pip install numpy pandas jupyter
```

Jupyter Notebooks



A gallery of interesting Jupyter Notebooks

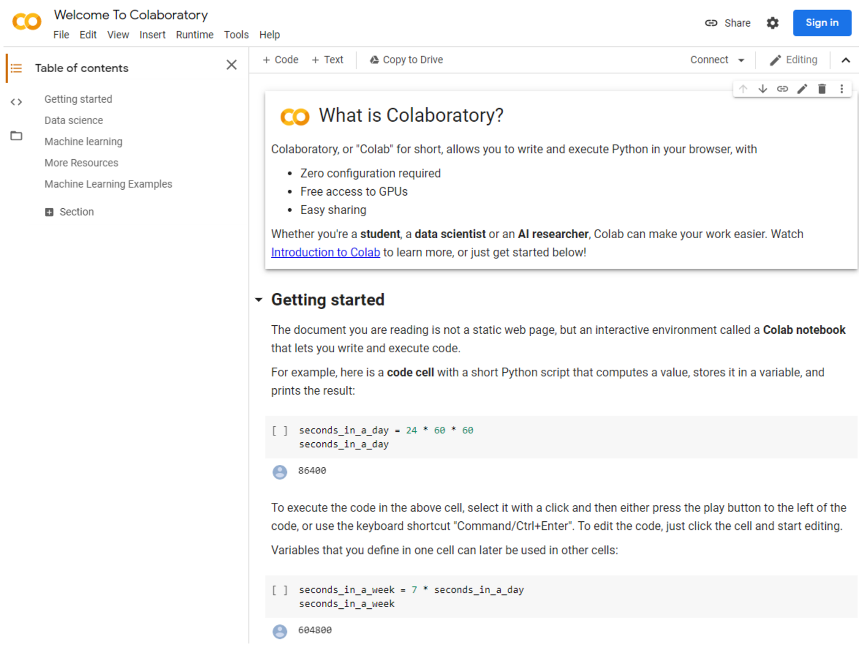
Dima Goldenberg edited this page 21 days ago · 119 revisions

This page is a curated collection of Jupyter/IPython notebooks that are notable. Feel free to add new content here, but please try to only include links to notebooks that include interesting visual or technical content; this should *not* simply be a dump of a Google search on every ipynb file out there.

Important contribution instructions: If you add new content, please ensure that for any notebook you link to, the link is to the rendered version using nbviewer, rather than the raw file. Simply paste the notebook URL in the nbviewer box and copy the resulting URL of the rendered version. This will make it much easier for visitors to be able to immediately access the new content.

Note that Matt Davis has conveniently written a set of bookmarklets and extensions to make it a one-click affair to load a Notebook URL into your browser of choice, directly opening into nbviewer.

Table of Contents



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day
```

```
86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
    seconds_in_a_week
```

```
604800
```

https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks

https://colab.research.google.com/

Stockholm University

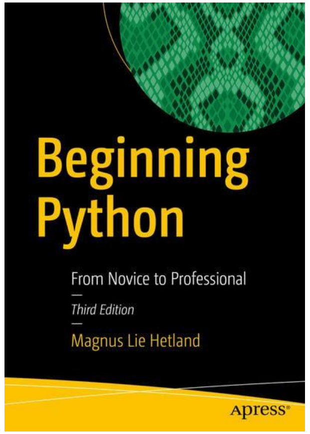
FURTHER READING

If you have no previous experience with Python...

Beginning Python, From Novice to Professional

(Download from SU library)

https://link-springer-com.ezp.sub.su.se/book/10.1007%2F978-1-4842-0028-5

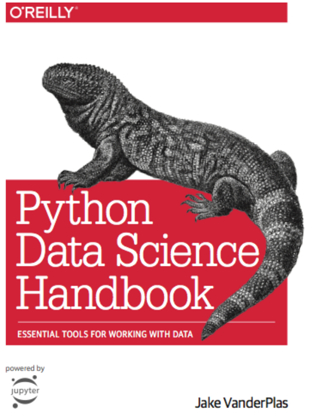


If you want a reference guide for IPython/Jupyter…

Python Data Science Handbook

(Available online)

https://jakevdp.github.io/PythonDataScienceHandbook/


O'REILLY
Python Data Science Handbook
ESSENTIAL TOOLS FOR WORKING WITH DATA
powered by jupyter
Jake VanderPlas

Summary Lab 0

- Installation
- Python Interpreter
  - Console, File, VS Code
- Variables and Data Types
- Data Structures
- Conditionals, Loops
- Definition of functions
- Package Management with PIP
- Basics of Jupyter, Numpy