Python Tsunami

Who are we?

Who are we

Center for Health Data Science (HeaDS)

- The Data Lab
  - Provides data science support for all research groups at SUND
  - Organizes workshops/seminars

- Research Units
  - work on different areas and topics within the field of health data science

https://heads.ku.dk

Who are we

Center for Health Data Science (HeaDS)

- Upcoming events:
  - Excel to R (June 15th & 16th)
  - Computerome user workshop (May 6th)
  - Git and Github workshop (date tbd)

https://heads.ku.dk

About this course

About this course

Originally developed at the Center for Protein Research (CPR) by:

- Alberto Santos Delgardo (University of Oxford)
- Henry Webel (NNF CPR)
- Annelaura Bach Nielsen (NNF CPR)
- Rita Colaço (PRI)

We say thank you for the course material (which we have adapted).

About this course

Your teachers:

Jose Alejandro Herrera Romero - Alex (HeaDS)

Marilena Hohmann (HeaDS)

Rita Colaço (PRI)

Inigo Prada Luengo (HeaDS)

Viktoria Schuster (HeaDS)

Henrike Zschach (HeaDS)

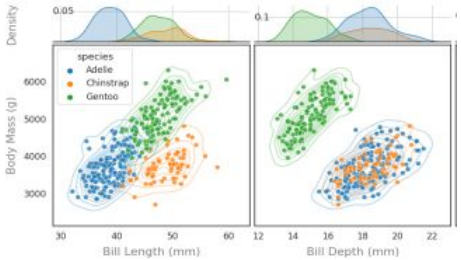
About this course

PROGRAM - Python Tsunami Part I		
	DAY 1 - Mon 20th of June	DAY 2 - Tue 21st of June
08:30 - 08:45	Morning coffee (optional)	
08:45 - 09:05	Introduction and Motivation	Libraries, Objects, References
09:05 - 09:45	Variables and data types	Questions from yesterday
09:45 - 10:00	Coffee break	
10:00 - 10:45	Iterables I: Lists	Pandas
10:45 - 11:00	Coffee break	
11:00 - 12:00	Iterables II: sets, dicts, tuples	Pandas
12:00 - 13:00	Lunch	
13:00 - 14:00	Booleans, operators and conditions	Pandas
14:00 - 15:15	Loops	Visualization
15:15 - 15:30	Coffee break	
15:30 - 17:00	Functions	Dataset Exercise

What is programming?

What is programming?

Programming is a set of **machine-readable** instructions that transform your input into your desired output.



input → program → output

We will attempt to shed some light on this part

Why is programming nice?

- Learning by doing:

  Difficult to ‘break’ a computer with wrong programming

- Reproducibility:

  The same thing should happen every time you run (*though some tasks involve some randomness)

- Transferable:

  Easily share your work with colleagues

- Many useful online resources
- Automate complex analysis workflows
- Important tool for working since we live in a data driven world

Why Python?

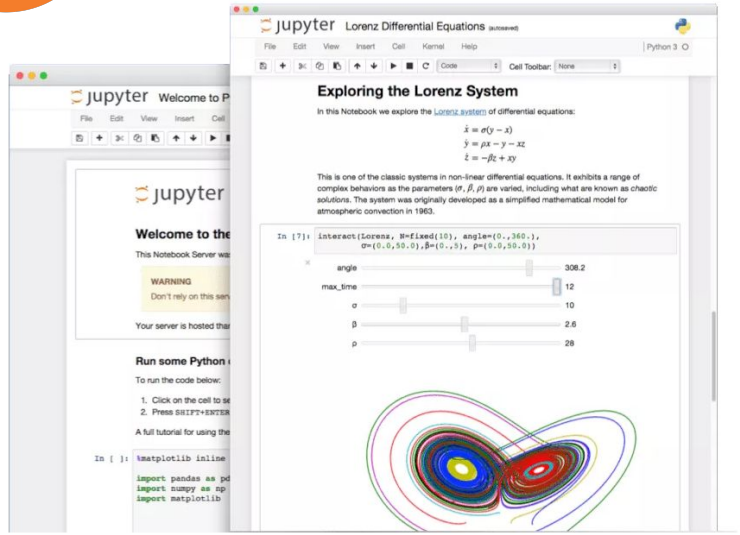
Python is a great programming language for both beginners and advanced programmers:

- Easy to grasp, close to natural language
- Many learning resources available
- Large community (i.e. stackoverflow for questions)
- Libraries
- Can do very advanced things like neural networks

Python environments

Jupyter notebook

The Jupyter Notebook is an **open-source application** to create and share documents that contain code, equations, visualizations and text (markdown).



- Browser-based development environment for creating, running and sharing python code
- Combine code with text and output
- Runs from your **local installation**. I.e. you need python and the libraries you want to use installed on your computer

Google Colab

Google Colab is a jupyter notebook hosted on google's servers, not your own machine. It still runs in your browser.

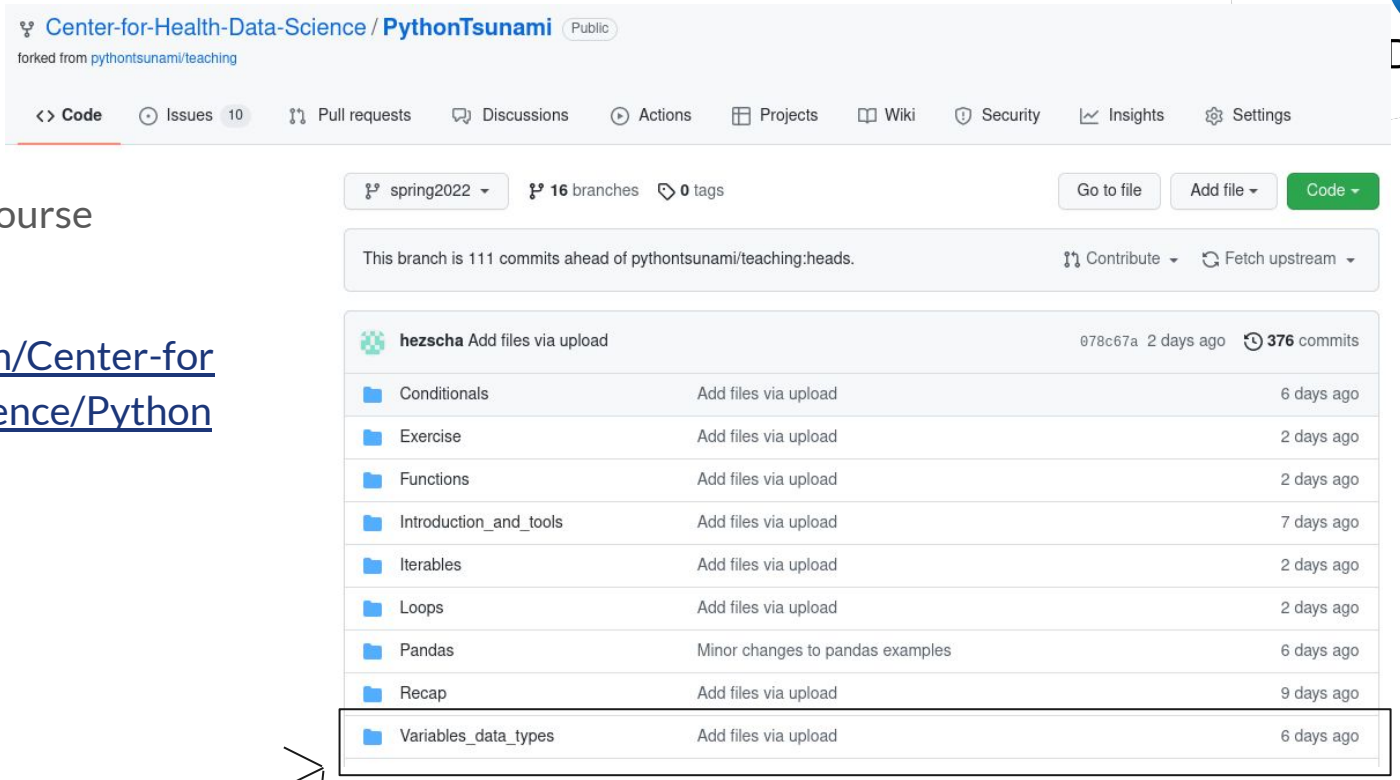
- tool to write, execute and share python code through the browser
- requires no setup to use and provides free access to computing resources on google's servers including GPUs
- is connected to a Google account and data and notebooks can be accessed through Google Drive.

We'll use colab during the course.

Course material

You can find the course material here:

https://github.com/Center-for-Health-Data-Science/PythonTsunami



Course material

spring2022 | PythonTsunami / Variables_data_types / | Go to file | Add file | ···

This branch is 111 commits ahead of pythontsunami/teaching:heads. Contribute | Fetch upstream

hezscha Add files via upload ab1c127 6 days ago **History**

..

README.md	Minor fixes	7 days ago
variables.ipynb	Update Colab link within the notebook	6 days ago
variables_solutions.ipynb	Add files via upload	6 days ago

README.md

HeaDS

Python Tsunami

notebook	content
variables.ipynb Open in Colab	Variables and data types

Course material

Remember to **save a copy** to your own google drive so you can save your notes and exercises!

Questions

You are always welcome to ask question during the course.

We also have a **padlet** board you can post your questions to. If you see a question posted that you also have, press the ‘like’ button.

At the start of day 2 we will tell a bit more about the inner workings of python and take up the most popular questions from the board.

Short Introduction

Take the next 5-7 mins to introduce yourself at your table:

- Name
- Position
- Unit
- Research topic (very briefly!)

Using libraries/packages

Libraries

Python has many libraries, also called packages, that other programmers have developed. Find and **use** them!

Well-maintained libraries generally are:

- Tested
- Optimized
- Documented

There is no need to reinvent the wheel. During this course we will use:

- Pandas (all the data analysis!)
- Math (basic math)
- Plotly express (visualization)

Libraries

If you are running python from a local installation, you need to have libraries **installed** before you can use them.

On google colab you can generally just import, they are already installed.

- Import the math library:

```
import math
```

- Now I can use functions from that library, i.e. calculating the logarithm or square root:

```
math.log(3)

math.sqrt(4)
```

Objects, variables, references

The language of python

Python is an **object-oriented** programming language. Therefore it helps if you primarily think of two different things:

Objects and functions*

- Objects are **pieces of information** (i.e. a number, a string of letters, a data table).
- We perform functions on objects. They are what we **do** to the information pieces.
- Which functions we can perform depends on the type of the object.

*technically functions are objects too, but let's not get too technical

An example

An object:

```
my_int = 3
```

A function:

```
math.log(my_int)
```

This does not work because the object is the wrong type:

```
math.log('hello')
```

Variable types

Often, we will be calling our information pieces **variables**. Don't be confused, all variables are also objects in Python!

Variables have **types**, which is the kind of object they are:

- This is an integer: `3`
- This is a float: `3.3`
- This is a string: `'Hello'`

There are also more complex object types that can hold several pieces of information, such as lists or data tables.

Variable types

Python applies **dynamic typing**. That means you do not need to declare what type your variable is when you create it. Python infers type from clues.

- This is a string because it is wrapped in quotes:

```
String123 = "Hello World!"
```

- This is a decimal number because it has a comma:

```
my_float = 3.3
```

- Calling something string does not make it a string:

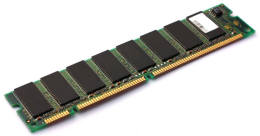
```
string123 = 3.3
```

- But casting does*:

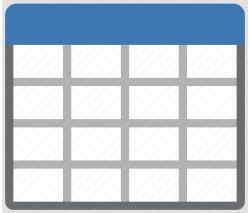
```
string(string123)
```

* A 'cast' is a function that changes an object's type

Variables and Objects



RAM = physical memory



Object = piece of info, i.e. a pandas dataframe

```
my_df = read.csv(...)
```

Variable inside python program

Every piece of data you use in python is stored somewhere in memory as an **object.**

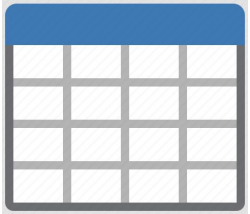
Variables are **references** to the stored object.

Objects can be **mutable** (most complex objects like dictionaries, lists, pandas dataframes) or **immutable** (most simple objects like strings, numbers, tuples).

Variables and Objects

Two variables can point to the **same** object. If they do, manipulating the content of one of the variables will also manipulate the other because there is **only one** actual object in memory, though they both reference it!

Variable 1



```
my_df = read.csv(...)
```

```
my_df2 = my_df
```

Variable 2

Object in memory

```
my_df.dropRows(...)
```

my_df2 is also changed!

This behavior is called **pass/call by object reference.**

It causes **mutable** objects to behave as **pass by reference** and **immutable** objects to behave as **pass by value**.