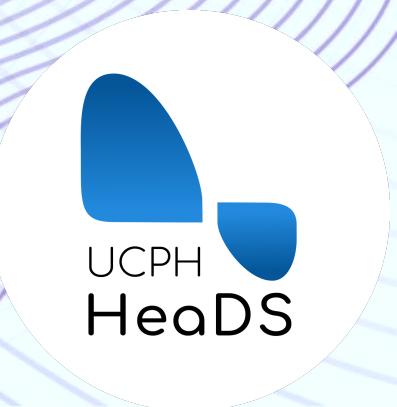


Python Tsunami

– November 3rd-5th –



Center for Health Data Science

<https://heads.ku.dk>

- **The Data Science 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



The Team

1. Thilde Terkelsen (HeaDS)
2. Annelaura Bach Nielsen (NNF CPR)
3. Jose Alejandro Herrera Romero - Alex (HeaDS)
4. Marilena Hohmann (HeaDS)
5. Rita Colaço (PRI)
6. Inigo Prada Luengo (HeaDS)
7. Viktoria Schuster (HeaDS)
8. Henry Webel (NNF CPR)



Former Team Members

1. Alberto Santos Delgado (HeaDS)
2. Dhouha Grissa (NNF CPR)
3. Grzegorz Jerzy Maciag (BRIC)
4. Katerina Nastou (NNF CPR)
5. Kübra Altinel (BRIC)
6. Marilena Hohmann (HeaDS)
7. Marta Matos (GENOME Center)
8. Nicholas Luke Cowie (DTU)



Practical Things about the Course

Program

Coffee and Q&A

Teams

Breakout rooms

Datathon

Practical Things about the Course

-- Program --

		Wednesday	Thursday	Friday
9:00-09:45	Introduction and motivation		Conditions	Visualization I
9:45-10:00			Coffee break	
10:00-11:00	Tools		Loops	
11:00-12:00	Variables and data types Numbers and operators		Functions	Visualization II
12:00-13:00			Lunch break	
13:00-14:00	Importing data			
14:00-14:45	Data structures		Numpy	Datathon
14:45-15:15			Coffee break	
15:15-15:45				Datathon
15:45-16:30	Data structures		Pandas	Presentations

Practical Things about the Course

-- Teams --

52 participants divided into **8 fixed teams**

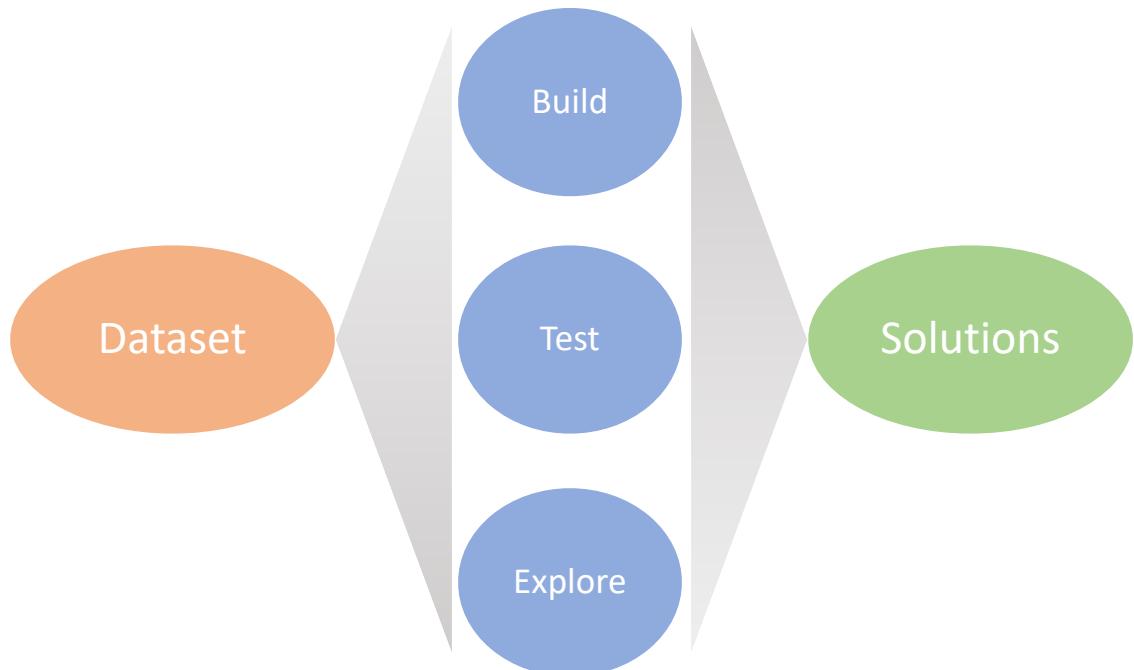
2 working modes:

- **Individual:** exercises
- **Teamwork:** discussions, practice and the Datathon

Practical Things about the Course

-- Datathon --

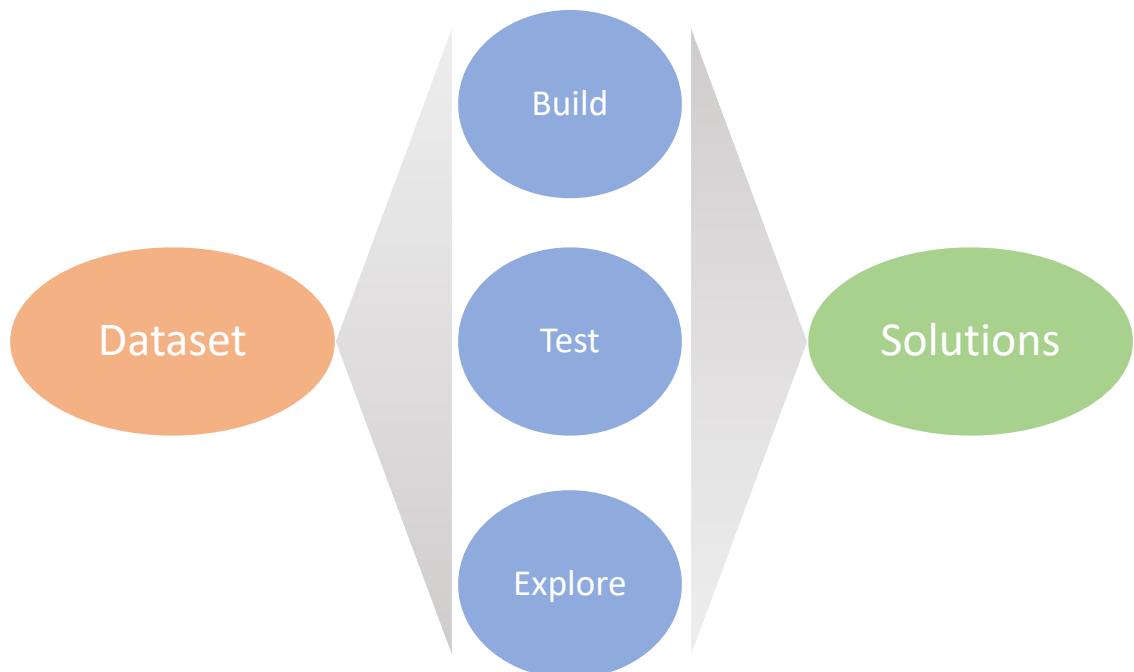
A Datathon is a **data-focused competition** — given a **dataset** and a limited amount of time, participants are challenged to use their **creativity** and **data science skills** to:



Practical Things about the Course

-- Datathon --

A **Datathon** is a **data-focused competition** — given a **dataset** and a limited amount of time, participants are challenged to use their **creativity** and **data science skills** to:



What will you learn in this course?

Tools to work with Python

The basics of Python

Some of the most relevant
scientific libraries

Visualization

Good practice

Motivation

Why will programming help you?

Programming is yet another **laboratory technique**

It helps you **automate processes** that you need to repeat again and again

It will **save you time**

It gives you **freedom** to process, analyze and plot your data as you want

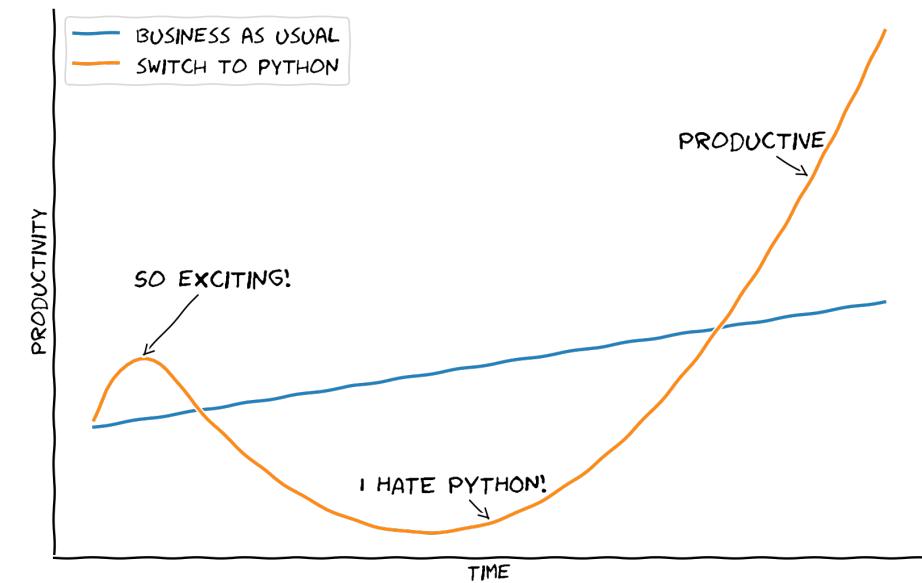
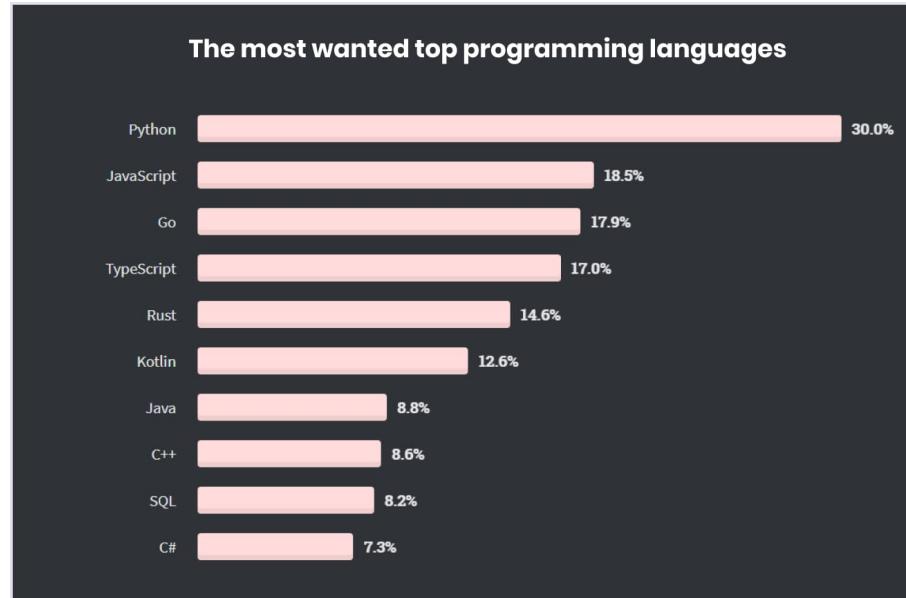
It will help you **demystify** bioinformatics

It will facilitate **communication** with bioinformaticians

It will **improve** your **CV**

Why Python?

- Python is **easy to use, powerful, and versatile**
- A great choice for **beginners and experts** alike
- Python's readability makes it a great **first programming language**
- It has a **huge community behind** developing useful libraries in many different fields (i.e biology, imaging, etc.)



Basic concepts

What's programming?

Variables, Functions

How do you approach a problem?

Car example

Class, Object, attribute, functions

What's programming

- Programming is a way of **communicating** with a device: computer, cellphone, ..., machine of any kind
- This communication is possible if you speak a **language** that the machine understands
- Programming is **not difficult**, mastering it might be a bit more **challenging**



Variables

variable_name = value

Variable is a way of **storing values** that you want to use later

- To define variables, we use **name** of variable and ‘=’ to assign values:

my_first_variable = 3

- Variables can have **different types**:

my_first_variable = 3 # integer

my_second_variable = "This is my second variable" # string

my_third_variable = 3.0 # float

Functions

def function_name (parameters)

Function is the way to **define actions**, i.e sum, print on the screen.

To define functions, we use the **reserved word def**:

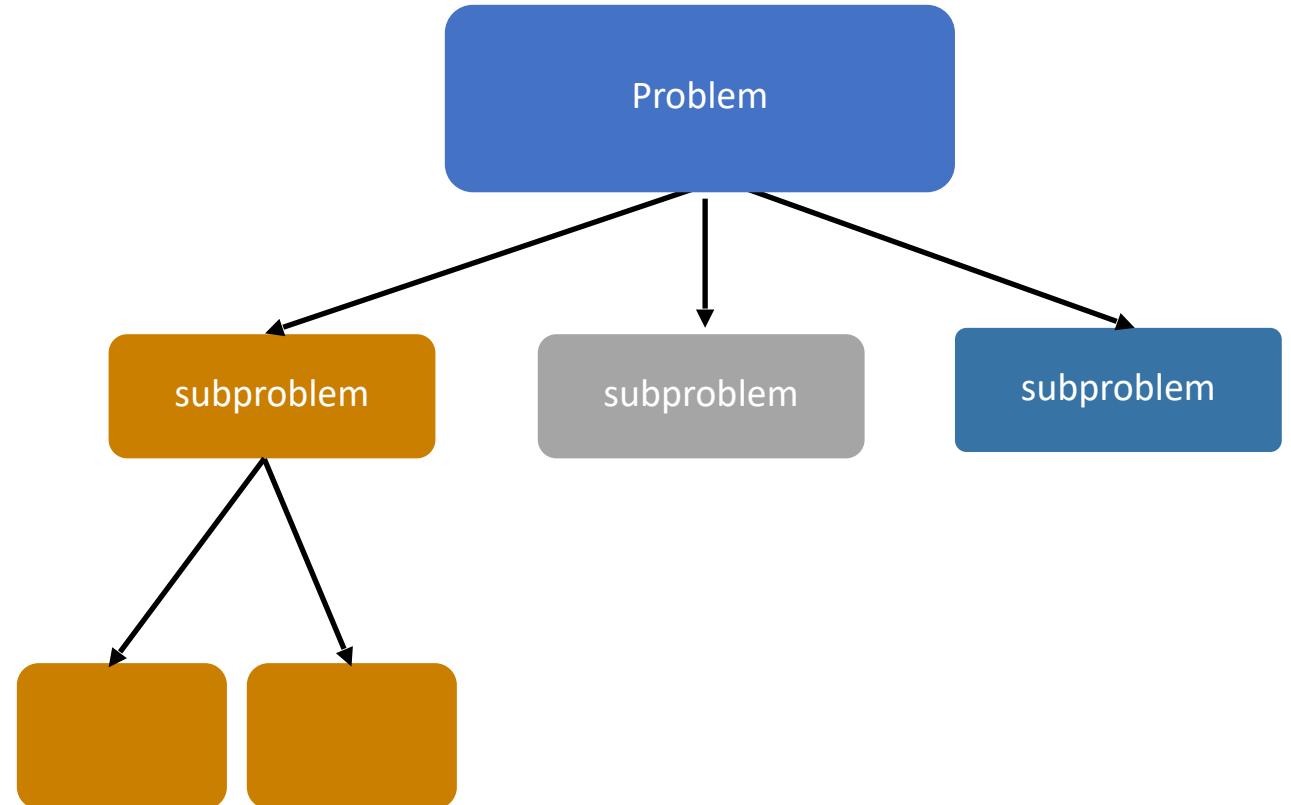
```
def sum_two_numbers(a, b):  
    return a + b  
  
def say_hi():  
    print("Hi")
```

Functions can be **called** by their **name** and specifying the **parameters**:

```
sum_two_numbers(a=7, b=5)  
> 12  
  
say_hi()  
> Hi
```

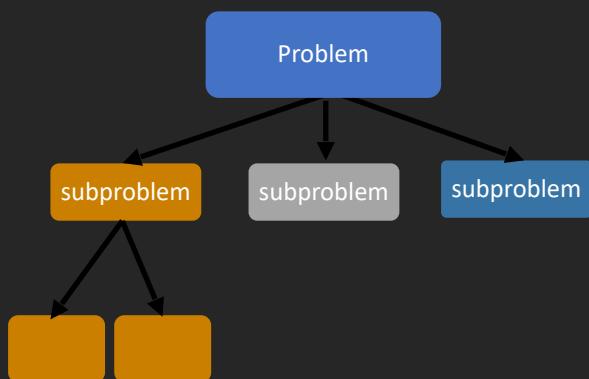
Strategy for Programming

Divide and conquer



The Car Problem

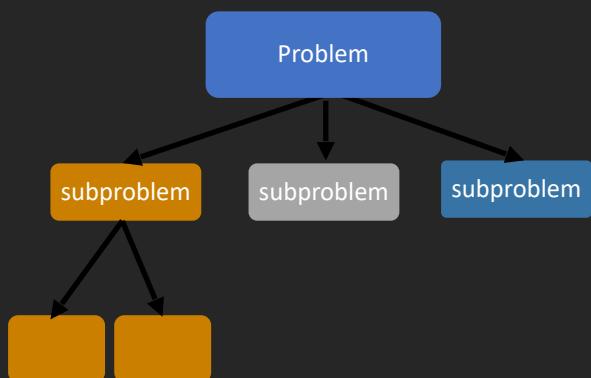
Describe a car



- Describe this **object**:
 - **Parts**: wheels, a stirring wheel, a frame, etc.
 - **Actions**: moves, breaks, etc.

The Car Problem

Describe a car



- Describe this **object** → **Class**:
 - **Parts**: wheels, a stirring wheel, a frame, etc. → **variables or attributes**
 - **Actions**: start, change gear, etc. → **functions**

Variables:

*color = "blue"
number_of_wheels = 4
motor = True
power = "gas"
gear = None*

Functions:

*def start_engine():
...
def change_gear(gear):
...*

Coffee with your team

- Introduce yourself and **what you do**
- Explain your **motivation** to take the course
- Discuss what **data** could be relevant for you



