PINK

{programming}

# Introduction to Python

@SEB

*November 17th, 2019*

# Agenda

- Introduction to programming
- Introduction to Jupyter Notebook
- Programming:
  - Variables
  - Data types
  - Lists
  - For-loops
  - Dictionaries
  - Functions
  - Currency converter

Your expectations?

# Introduction to programming

## What is a program?

- "*A sequence of instructions, written to perform a specific task on a computer*" - Wikipedia

- The instructions has to be written in a language that the computer can understand, **a programming language**
    - Ex: Java, JavaScript, C++, Python

# Introduction to programming

## Why learn Python?

- Python is the fastest-growing major programming language today
- Syntax is simple and easy to learn
- Versatile language:
  - Web development
  - Data science
  - Machine learning
  - Game development
  - ...and much more!
- Widely used in the industry, both in large and small companies
- Extremely popular with a huge community of developers who can support you

Let’s move around :)

# Introduction to Jupyter Notebook

## What is it?

- A very popular and powerful tool that combines:
  - Code
  - Rich text
  - Images
  - Mathematical equations
  - Plots
  - Interactive figures and widgets

...and much more, into a single document.

# Introduction to Jupyter Notebook

Let's try it out!

1. Download .ipynb file from https://github.com/mykys
2. Open the file in https://jupyter.org/try

# Introduction to Jupyter Notebook

Exercise

1. **Add new cell**
    a. [+]
    b. esc + a
    c. esc + b
2. **Remove cell**
    a. [scissor]
    b. esc + dd
3. **Run cell**
    a. Shift + enter

# Python

## Data types

- int : 10
- float : 10.5
- String : “Pink”
- boolean : True

# Python

## What is a variable?

- a reserved memory location to store values
- a variable name can be anything - the more descriptive, the better :)
  - x = 30 | age = 30
- good to remember when naming variables:
  - the name should start with a letter
  - cannot start with a number
  - alpha-numeric characters (A-z & 0-9) and underscores
  - case sensitive, Age and AGE are different
- snake case is preferred
  - pink_programming | PinkProgramming | pinkProgramming

# Python

## What is a variable?

- how much memory being reserved depends on what value you want to store
- no need to declare (create) a variable explicitly
- the equal sign (=) is used to assign a value to a variable
    - age = 30 | variable to the left and value to the right
- the value of a variable can change
    - age = 40

**Exercise:** Create four variables and assign values of different types to them (1a-c)

# Python

How do we use variables?

**Exercise:** Explore with arithmetic operators +, -, *, /! (2)

# Python

## Lists

- a data structure that can store a collection of items
- defined using square brackets [item1, item2, item3, ....]
- each item is separated with a comma
- index (position) starts from 0,1,2,...,n

**Exercise:** Get the first, second and last value from the list fruits (ex)

**Exercise:** Create a new list with names of your friends. Repeat the exercise above. (3a-b)

# Python

## Lists

- count number of items in the list : len([...])
- add an item to the end of the list : append(*item*)
- insert an item at a given position : insert(*position*, *item*)
- remove the first occurence of this item : remove(*item*)

These are some of the basic functions. More can be found in the documentation :)

**Exercise:** Try some of the basic functions implemented for lists (4a-f)

# Python

## For-loops

- useful when we want to go through every item in i.e. a list

**Exercise:** Print each name in your list of names.

**Exercise:** Add a small change to each name in your list

# Python

## For-loops

- useful when we want to go through every item in i.e. a list

**Exercise:** Print each name in your list of names.

**Exercise:** Add a small change to each name in your list

# Python

## Dictionaries

- a data structure that can store a collection of key-value pairs
- defined using curly brackets {key1 : value1, key2 : value 2, ...}
- a colon (:) separates each key from its associated value
- each key-value pair is separated with a comma

**Exercise:** Get the capital of other countries using the dictionary *countries* (6a)

# Python

## Dictionaries

- add a new key-value pair : countries[“germany”] = “berlin”
- update existing key-value pair : countries[“sweden”] = “malmo”
- remove existing key-value pair : **del** countries[“sweden”] (delete key)

**Exercise:** (6b-g)

- Add a new key-value pair to *countries*
- Update existing key-value pair
- Remove existing key-value pair

# Python

## Dictionaries

**Exercise:** Print all countries and capitals in your dictionary *countries*

# Python

## Functions

- What is a function?
  - a block of code that will run when being called
- Why do we want to use functions?
  - reuse code
  - a function usually performs one action, i.e. add two numbers
- How do we create a function?
  - See Jupyter Notebook

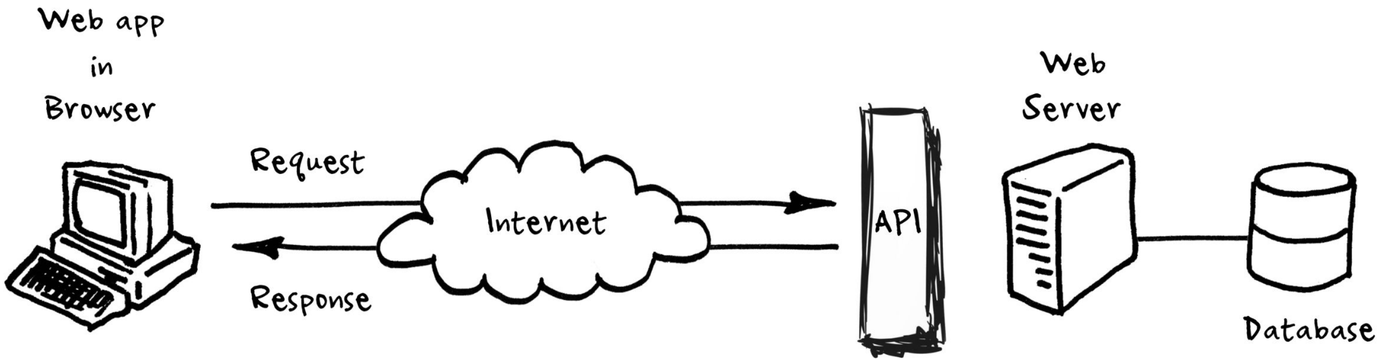
**Exercise:** Create three functions that can subtract, multiply and divide two numbers.

# Currency converter

# API

## Application Programming Interface

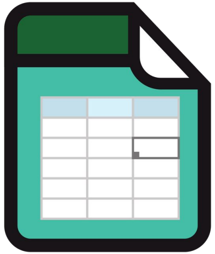
- What exactly is it?
  - API allows applications to communicate with one another.

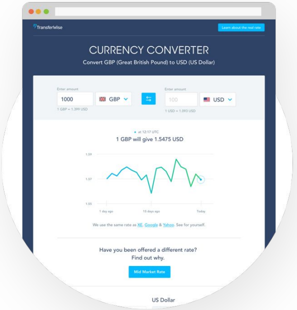


# API

## Application Programming Interface

- Why do we want to use API’s?
    - Imagine, you want to build a currency converter application and need access to the latest conversion rates from SEB…





CURRENCY CONVERTER

Convert GBP (Great British Pound) to USD (US Dollar)

Enter amount: 1000 GBP

1 GBP = 1.399 USD

Enter amount: 100 USD

1 USD = 1.093 USD

at 12:17 UTC

1 GBP will give 1.5475 USD

We use the same rate as XE, Google & Yahoo. See for yourself.

Have you been offered a different rate?
Find out why.

US Dollar

# API

## How do we use it?

1. Go to [https://developer.sebgroup.com/](https://developer.sebgroup.com/)
2. Click on **API Products**
3. Click on **Documentation** for *Foreign Exchange Rates*
4. Important information:
    - The url address to access the exchange rates
    - An APIKey is needed for an input parameter called header
    - Parameters to get specific exchange rates:
        - listed_currency | the currency to convert *from*
        - unit_currency | the currency to convert *to* (only SEK is supported)

# Currency converter

Create a function to convert USD into SEK

- **import** requests
  - standard API library in Python
  - a library is a set of functions someone else has written
- headers
  - additional information to your API call
    - APIKey
    - Data format

# Public APIs

https://github.com/public-apis/public-apis

How do you feel?

Thank you for your participation! :)