A picture containing outdoor, object, clock, sitting

Description automatically generated

A guide to the course

Contents

[Introduction 4](#_Toc29153603)

[Why should you learn programming language? 4](#_Toc29153604)

[Structure of the course 4](#_Toc29153605)

[Grading 5](#_Toc29153606)

[Project 5](#_Toc29153607)

[Assignments 5](#_Toc29153608)

[Installation 6](#_Toc29153609)

[Installation of Anaconda 6](#_Toc29153610)

[Preparing of Anaconda 8](#_Toc29153611)

[Version Control 10](#_Toc29153612)

[What is Github? 10](#_Toc29153613)

[Cloning Github repository 10](#_Toc29153614)

[Modules 12](#_Toc29153615)

[Variables 13](#_Toc29153616)

[String 13](#_Toc29153617)

[Numerical values 14](#_Toc29153618)

[Boolean 15](#_Toc29153619)

[Conversion 15](#_Toc29153620)

[Naming conventions 15](#_Toc29153621)

[Operators 16](#_Toc29153622)

[Arithmetic operators 16](#_Toc29153623)

[Assignment operator 16](#_Toc29153624)

[Comparison operators 16](#_Toc29153625)

[Logical operators 17](#_Toc29153626)

[Flow Control 18](#_Toc29153627)

[If – Else 18](#_Toc29153628)

[ELif 18](#_Toc29153629)

[Objects 19](#_Toc29153630)

[Loops 20](#_Toc29153631)

[Functions 21](#_Toc29153632)

[Classes 22](#_Toc29153633)

# Introduction

Welcome to the *Introduction to the Computational Science*course by School of Energy Systems! My name is IK and I will guide you through the course.

Programming is literacy of the XXI century. Without programming we would not live in the world as it is now. Software is used everywhere and for various purposes, starting from clocks on your phone to controlling spaceships. For us, as engineers, it is useful tool for controlling and designing of the mechanical systems.

During the course you will need only a computer I strongly recommend to use your own computer, because it will make studying easier. Before starting the course

## Why should you learn programming language?

* Self-learner’s advantage: tons of material in internet
* Gaining problem solving and thinking skills
* Becoming more efficient and productive
* Combining technical skills with creativity
* Understanding how software works
* Freedom and flexibility in life
* Improving your communication and collaboration skills
* Becoming self-employed, starting your own business or work as freelancer
* Improving your portfolio

## Structure of the course

We will cover following topics:

* Syntax
* Files System
* Version Control
* Best Coding Practices
* Variables
* Python Objects (String, Dictionary, Tuple, Sequence, Set, Lists)
* Control Flow (Boolean Expressions/Operators, If, Else, Else If)
* Loops (For, While),
* Functions
* Classes
* Matlab Introduction (optional)

## Grading

## Project

Should be submited as Python file 🡪 No report needed

Group of 3 – 4 students

Deadline: 15.02.2020

Will be evaluated by following criterias:

* Creative idea, which is interesting for everybody. Useful for your studies, hobby or everyday life.
* Code is easy to read and maintain
* If used Github or Stackoverflow code must be changed for your needs. **NO COPYPASTING!**

## Assignments

Made during the lecture before break

Will be done in Microsoft Visual Studio Code

Submit Python file (.py extension) via Github, assignments folder

Deadline: Next week

Will be evaluated by following criterias:

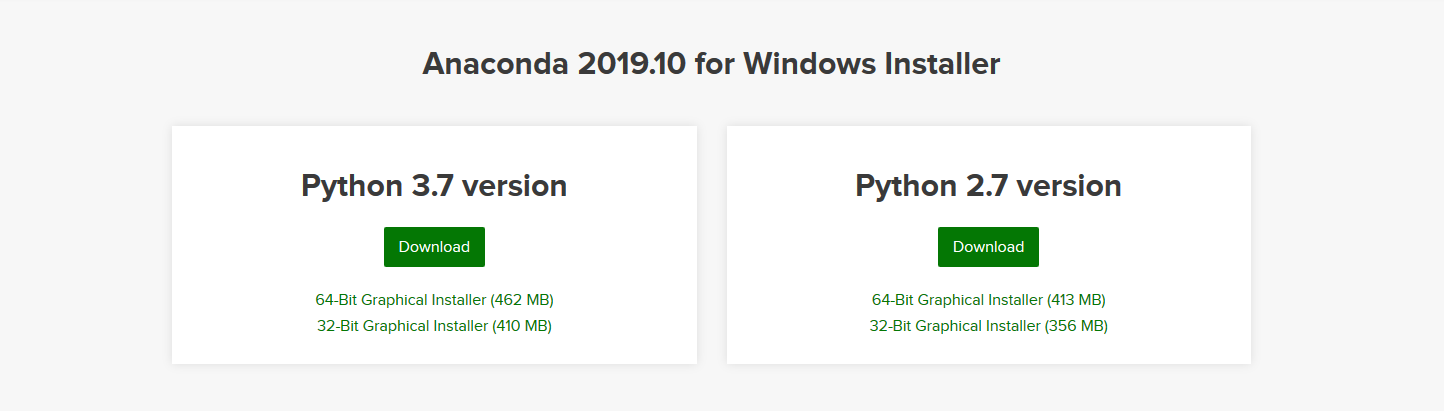
* Runs without exceptions
* Code is easy to read

# Installation

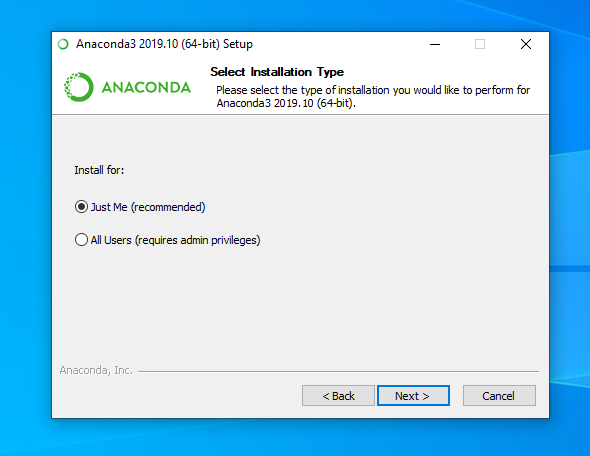
Anaconda is a distribution of Python made for scientific computing applications, such as data science, machine learning and etc. During this course we will use it for learning purposes.

## Installation of Anaconda

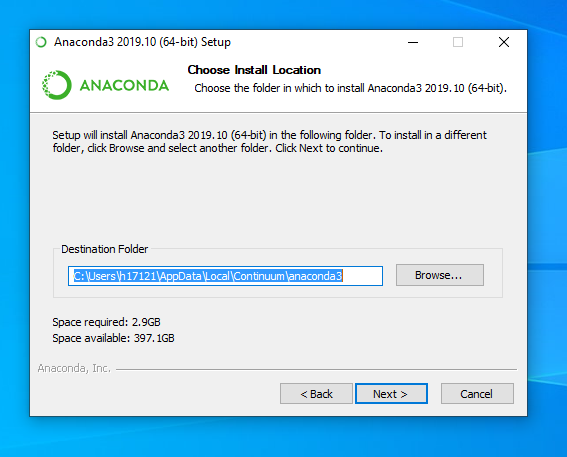
Step 1: Go to <https://www.anaconda.com/distribution/#download-section>. Load Python 3.7 version.



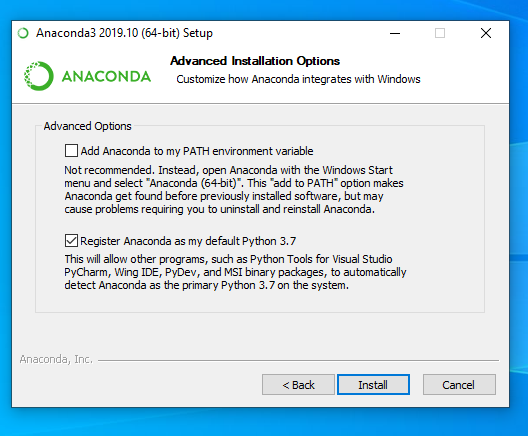
Step 2: Run the .exe file. Use install for **Just Me** option.



Step 3: Install to your user location (default).



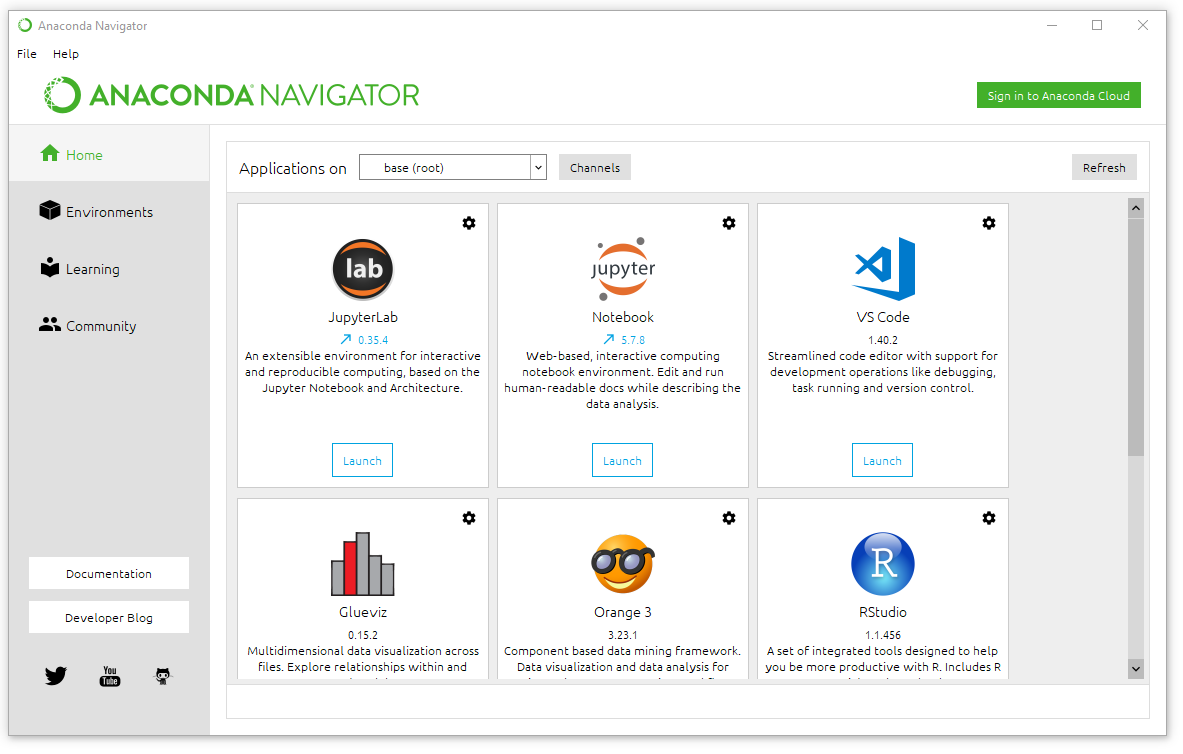
Step 4: Keep everything as default and click install.



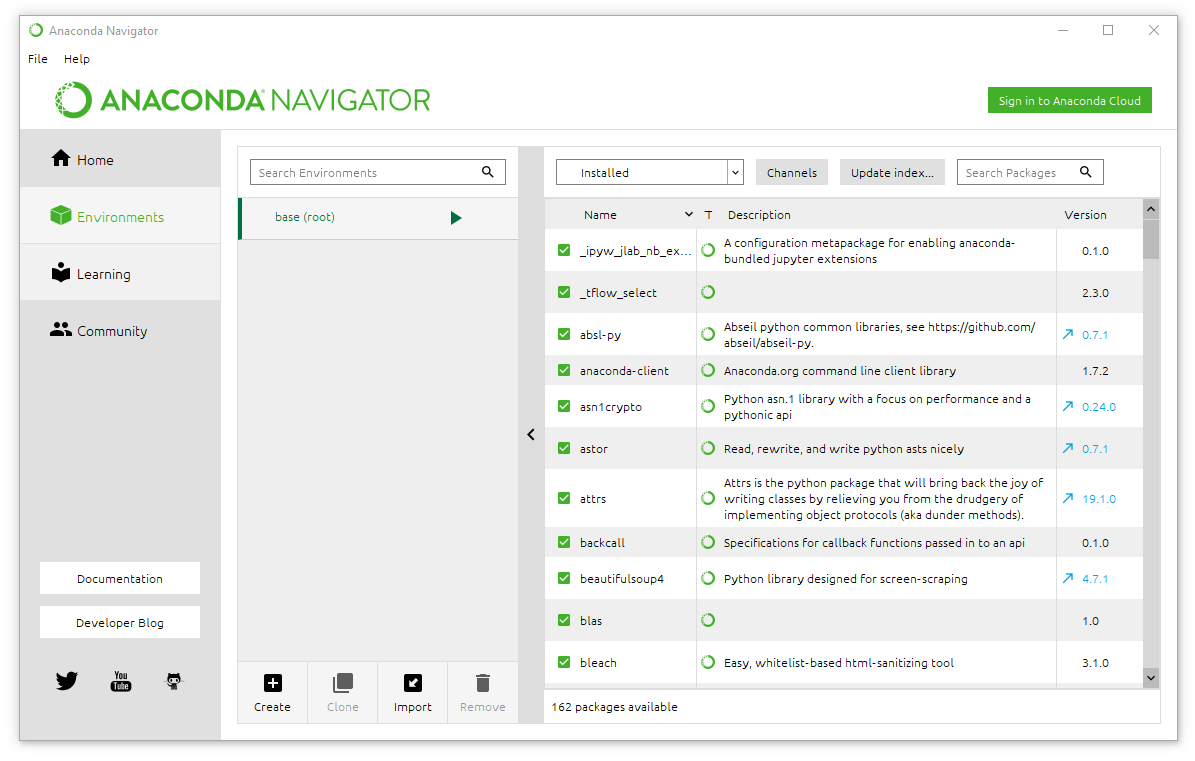
## Preparing of Anaconda

**WARNING! If you use your own computer, you can skip this step.**

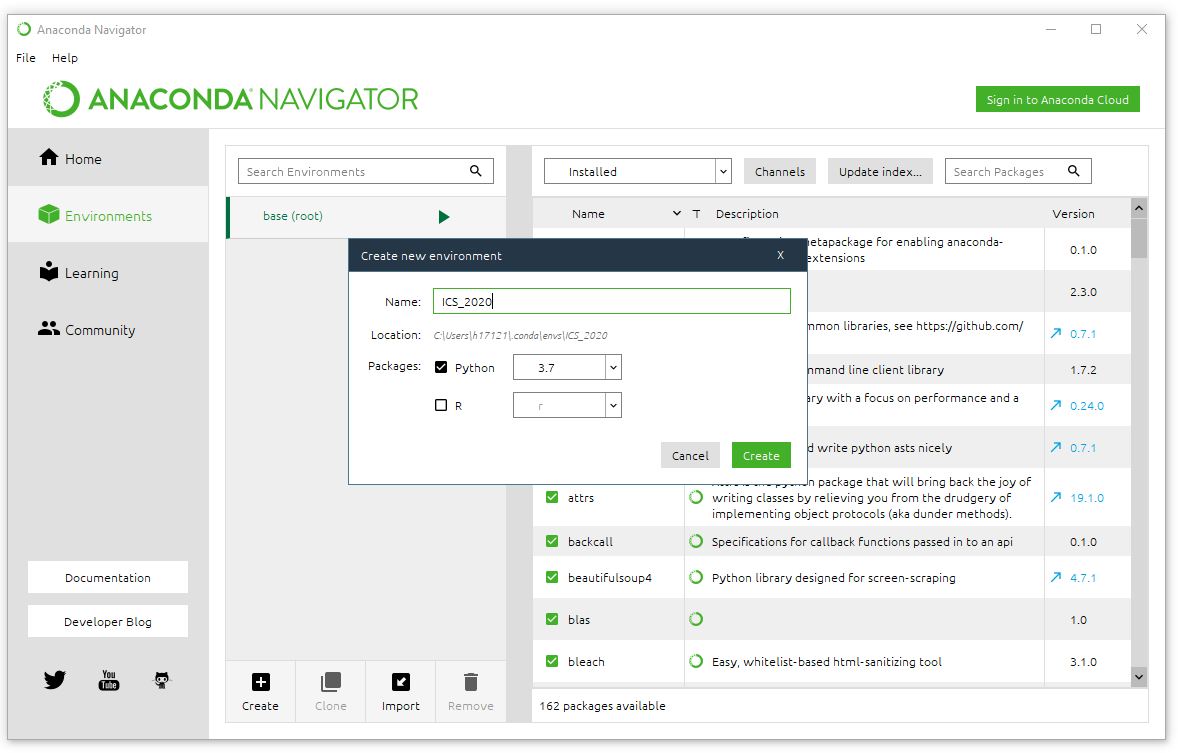
Step 1: Go to Anaconda Navigator. Open the Environments tab.



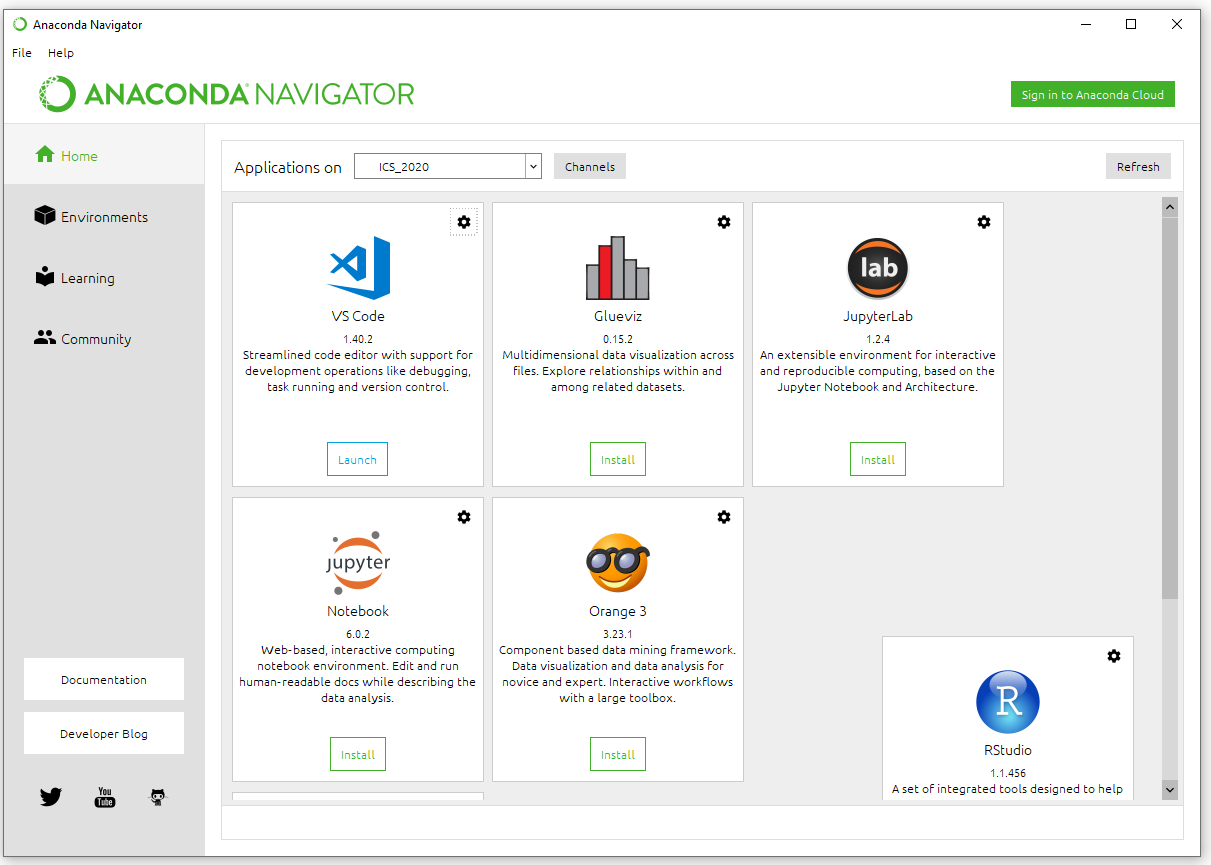
Step 2: Create new environment by hitting Create button.



Step 3: Create a new environment. Make sure that you use Python 3.7 version!



Step 4: The environment is ready for work. Now click Install button under the Jupyter Notebook. If computers asks admin rights, just click cancel.



# Version Control

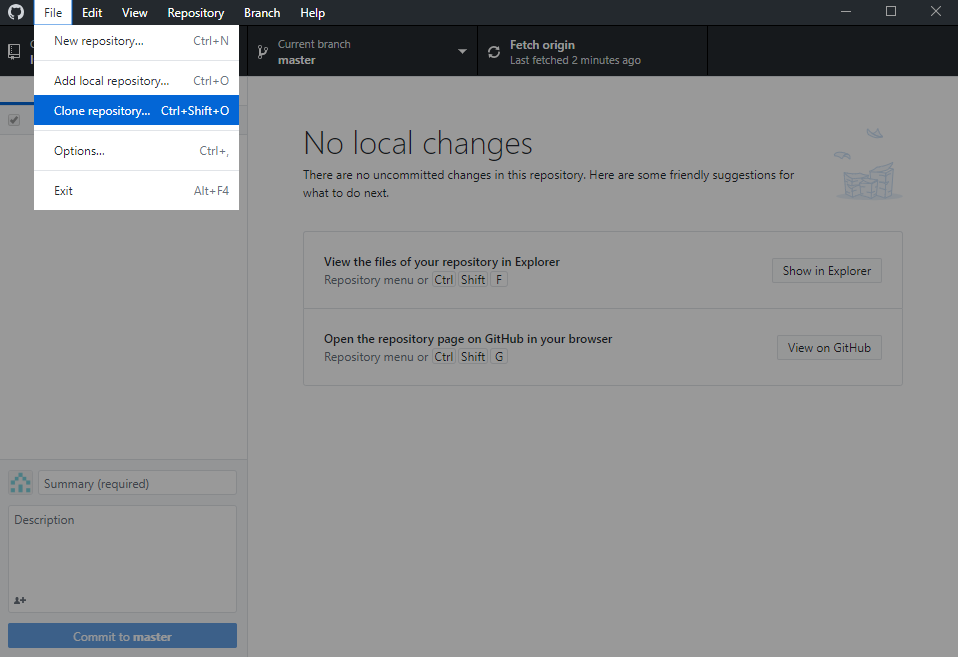
## What is Github?

GitHub is a platform for version control and collaborative work on code. It allows team of developers to work on the same project and track what was done before.

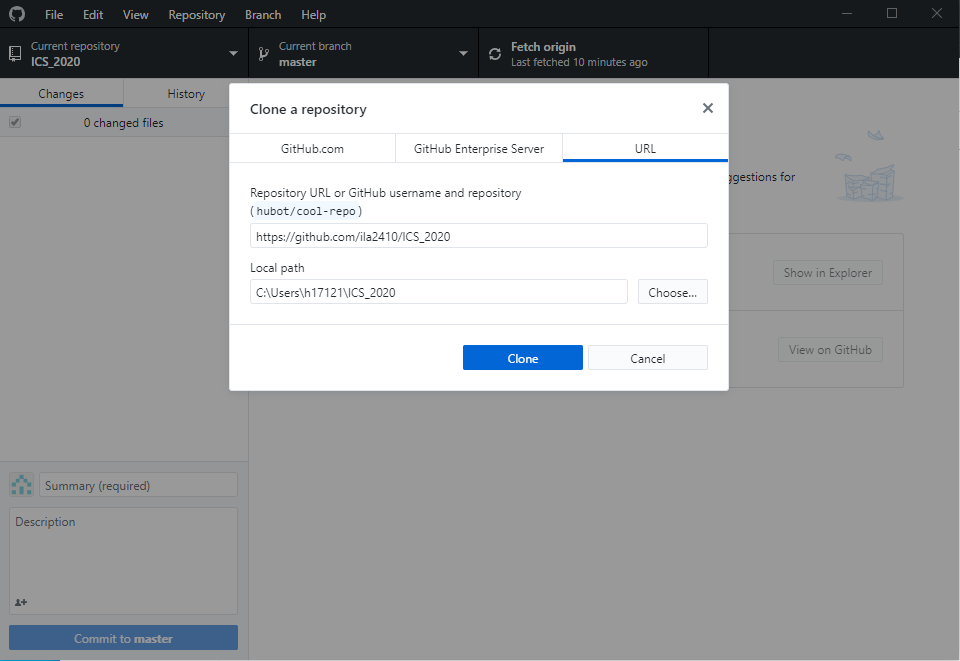
## Cloning Github repository

During this course all materials and assignment are located at the course Github repository. You should clone the repository using Github Desktop. You can clone repository by following these steps:

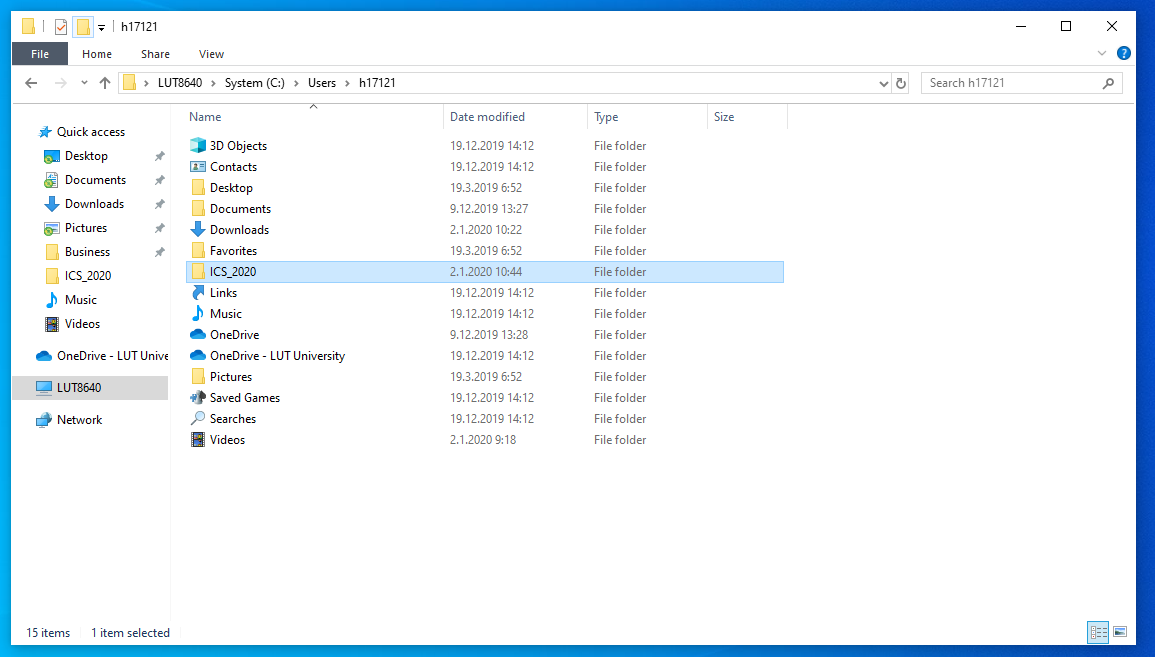
Step 1: Open Github Desktop. Go to File🡪Clone repository



Step 2: Go to URL tab, put URL of the repository (<https://github.com/ila2410/ICS_2020>) and choose path where to clone it. Then click clone button. **WARNING! If you are using LUT computer save to your C:\Users\your\_user\_number (for example C:\Users\n5198) folder, otherwise, it would not save anything.**



Step 3: Open File Explorer and go to the path. Now you will have all needed materials.

0

# Modules

Modules are files, which containing a script done by other programmers. It is used by programmer for not “reinventing the wheel”. You can add module by import command. For example:



For making your program more lightweight you can add only needed parts by using from. For example:

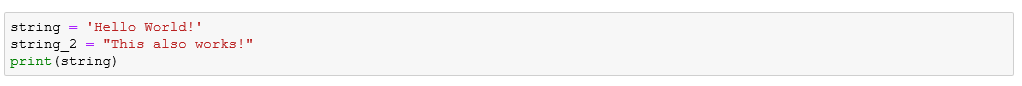


# Variables

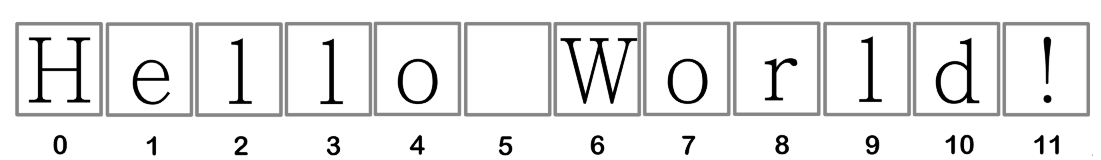
We need some way of storing any information and manipulate it. This is where variables come into the picture. Variables are exactly what the name implies - their value can vary, i.e., you can store anything using a variable. Variables are just parts of your computer's memory where you store some information. Unlike literal constants, you need some method of accessing these variables and hence you give them names. We will cover three major topics here: Numerical data, Text data and naming rules.

## String

**String** represents **text** data. A string is a sequence of characters. Strings are basically just a bunch of words. You will be using strings in almost every Python program that you write, so pay attention to the following part. String always is enclosed into quotation marks '' or "".



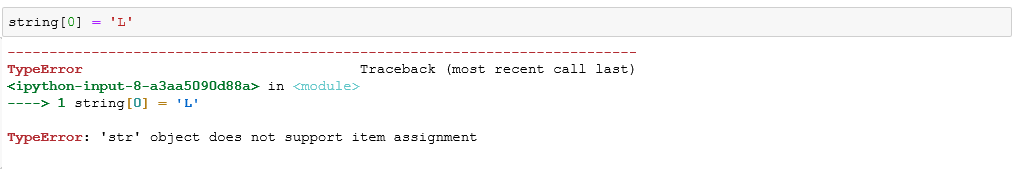
You can access a single character from string. Each character marked from 0 to number of characters.



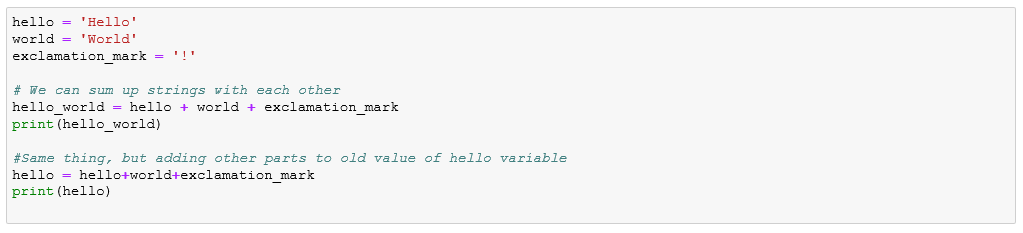
For example:



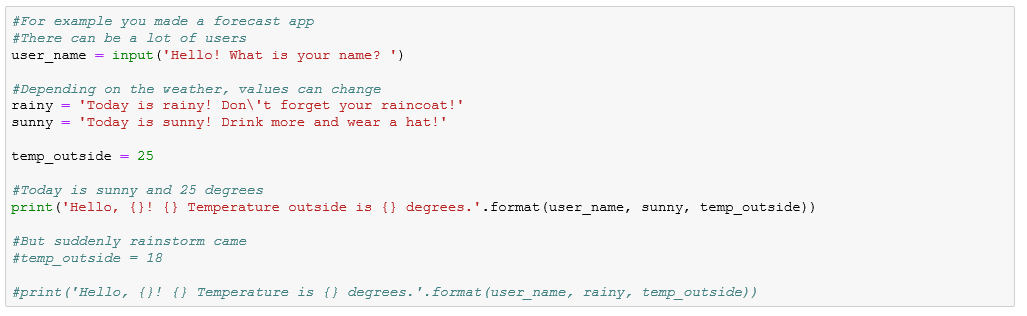
Once String is initiated, you cannot directly change it. It will show following error:



You can concatenate strings. In other words, create string by adding multiple strings together.



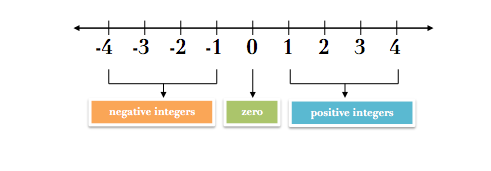
You can format the string by using format() method. By using this method you can add some value to the string in any place. In Python 3 you should add to the string {} brackets. For example:



## Numerical values

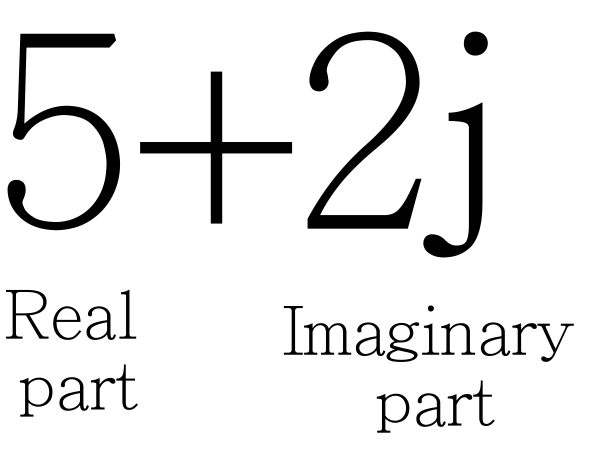
Python supports four different numerical types: integers(int), float, long and complex.

**Integers** are whole numbers, which can be negative or equal to zero, as shown at the figure below.



**Floats** are floating point numbers, meaning they have a decimal place. Same as integers, they are signed and can hold positive and negative values.

**Complex** are numbers with real and imaginary parts.



## Boolean

True or False variables. Used for tracking of events, in other words, check conditions if they are True. They will be very useful in flow control and while loops.



## Conversion

You can convert variables into other types:

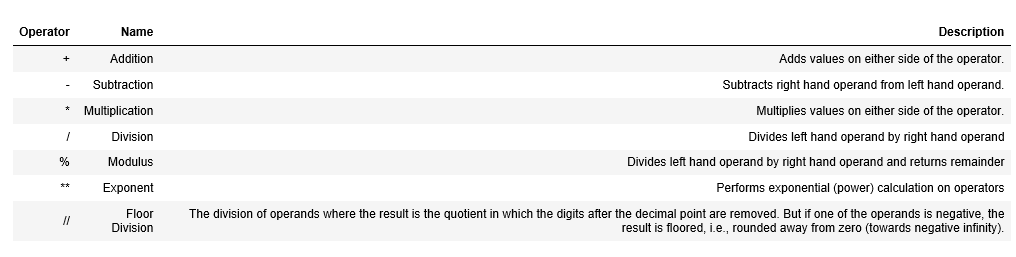
* str() – converts variable into string (bool, int, float, complex)
* int() – converts variable into integer (int, float, complex). You can convert string as well, but make sure, that string have only numerical values
* float()- converts variable into integer (int, float, complex). You can convert string as well, but make sure, that string have only numerical values
* complex() - converts variable into integer (int, float, complex). You can convert string as well, but make sure, that string have only numerical values

## Naming conventions

* Avoid using too general names. For example: variable\_1, number\_1, my\_list. Instead use descriptive names. For example, if you need to store the name of student use: student\_name
* You can use underscores or camelcase: under\_score CamelCase
* Don't start names from number
* Don't start from signs.

# Operators

## Arithmetic operators



## Assignment operator

**Assignment** operators are used to **assign values** to a variables.

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Example | Equivalent to | Explanation |
| = | x=5 | x=5 | - |
| += | x+=5 | x=x + 5 | Adds some value to the existing value of a variable. |
| -= | x-=5 | x=x - 5 | Subtracts some value to the existing value of a variable. |
| \*= | x\*=5 | x=x \* 5 | Multiplies existing value (x) of a variable to some value (5). |
| /= | x/=5 | x=x / 5 | Divides existing value (x) of a variable to some value (5). |
| %= | x%=5 | x=x % 5 | Takes modulus of existing value (x) to some variable (5). |
| //= | x//=5 | x=x // 5 | Floor division of existing value (x) to some variable (5). |
| \*\*= | x\*\*=5 | x=x \*\* 5 | Puts existing value (x) in power of some variable (5). |

## Comparison operators

These operators compare the values on either side of them and decide the relation among them. They are also called Relational operators.

|  |  |  |
| --- | --- | --- |
| Operator | Example | Explanation |
| > | x>5 | Greater than. |
| < | x<5 | Less than |
| == | x==5 | Equal |
| != | x!=5 | Not equal |
| >= | x>=5 | Greater or equal. Meaning, that it includes a value. |
| <= | x<=5 | Less or equal. Meaning, that it includes a value. |

## Logical operators

Used for processing of Boolean or statements.

|  |  |  |
| --- | --- | --- |
| Operator | Example | Explanation |
| AND | x and y | Returns True if both of the values are True |
| OR | x or y | Returns True if one of the values are True |
| NOT | x not y | Returns True if if operand is False |

# Flow Control

Flow control allow to execute some part of the code depending on the value of some conditions.

## If – Else

If/Else statement have the following structure:



It is important to know, that everything under if must have an indentation. It means, that you need to add tab or 4 spaces. We use these statements in real life without even noticing. For example:



## ELif

Elif statement is somewhat self-descriptive. It is equal to adding another if to the script. It have the following structure:



For example:



# Objects

COMING SOON

# Loops

COMING SOON

# Functions

COMING SOON

# Classes

COMING SOON