Learning Objective

- After finishing this course, you will:
  - Be able to explain what a programming language is, such as Python.
  - Understand the concept of variables.
  - Be able to define appropriate variables.
  - Understand the types of operations.
  - Be able to use operations for computation.
  - Understand the flow control such as branchings and repetitions.
  - Be able to write complex programs using flow controls.

Course Outline

- Introduction to Python Fundamentals
  - Module 1: Introduction to Python
  - Module 2: Variables and Operations
  - Module 3: Flow Control - Branching
  - Module 4: Flow Control - Repetition

Module 1: Introduction to Python

- Before we learn Python, we first need to know what are computers, and
- How do computers work.

Computers

- What are computers?
  - Electronic devices that compute data according to instructions.
  - Components:
    - CPU(Central Processing Unit),
    - Memory,
    - Storage,
    - Input/output devices.

Instructions

- What are the instructions?
  - Computers can only understand binary (0 and 1)
  - We cannot understand / use machine language directly.
  - We need high level languages that:
    - Similar to our natural language.
    - Can be converted to machine language.
- So we have Programming Languages.
  - A programming language is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.

Python

- Python is a high-level general purpose programming language. Python consistently ranks as one of the most popular programming languages.
- Strength of Python (Why we use Python):
  - Smaller code length,
  - More friendly to beginners,
  - Extensive libraries (packages),
  - Best for Data Science!
- Python 3 is the current standard.

Python Environment

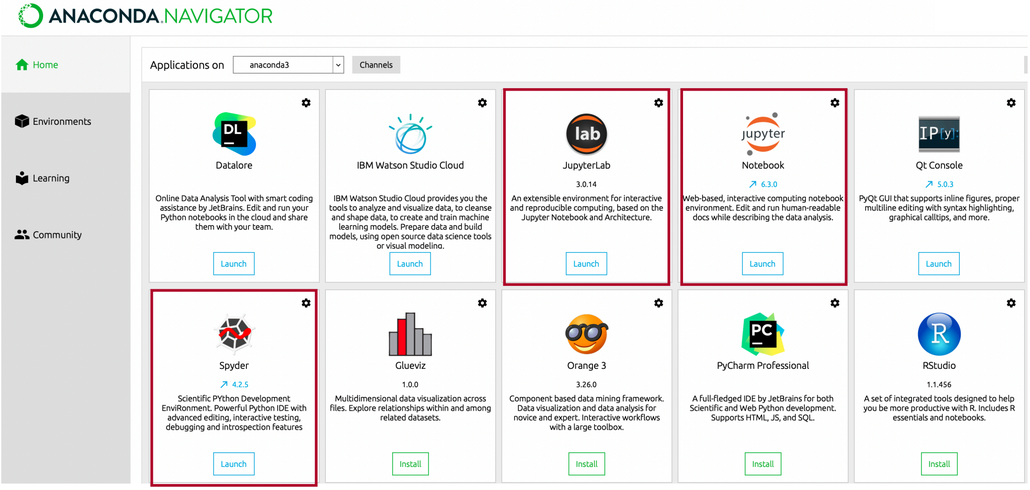
- To write and run Python programs, we need to setup Python environment.
  - We can use online environment to avoid installment and configuration (Recommend).
  - Or install a local IDE (Integrated Development Environment).

Google Colab

- Google Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser.
  - Zero configuration required.
  - Easy sharing.
  - Free with a Google Account.
  - Recommended for our specialization.

Anaconda

- You can install a local IDE, such as Anaconda
  - Run JupyterLab or Jupyter Notebook for Interactive Python Environment.
  - Run Spyder for script Python environment.



Google Colab

File Edit View Insert Runtime Tools Help

Share Di

Table of contents

Getting started
Data science
Machine learning
More Resources
Machine Learning Examples

Section

+ Code + Text Copy to Drive Connect Editing

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

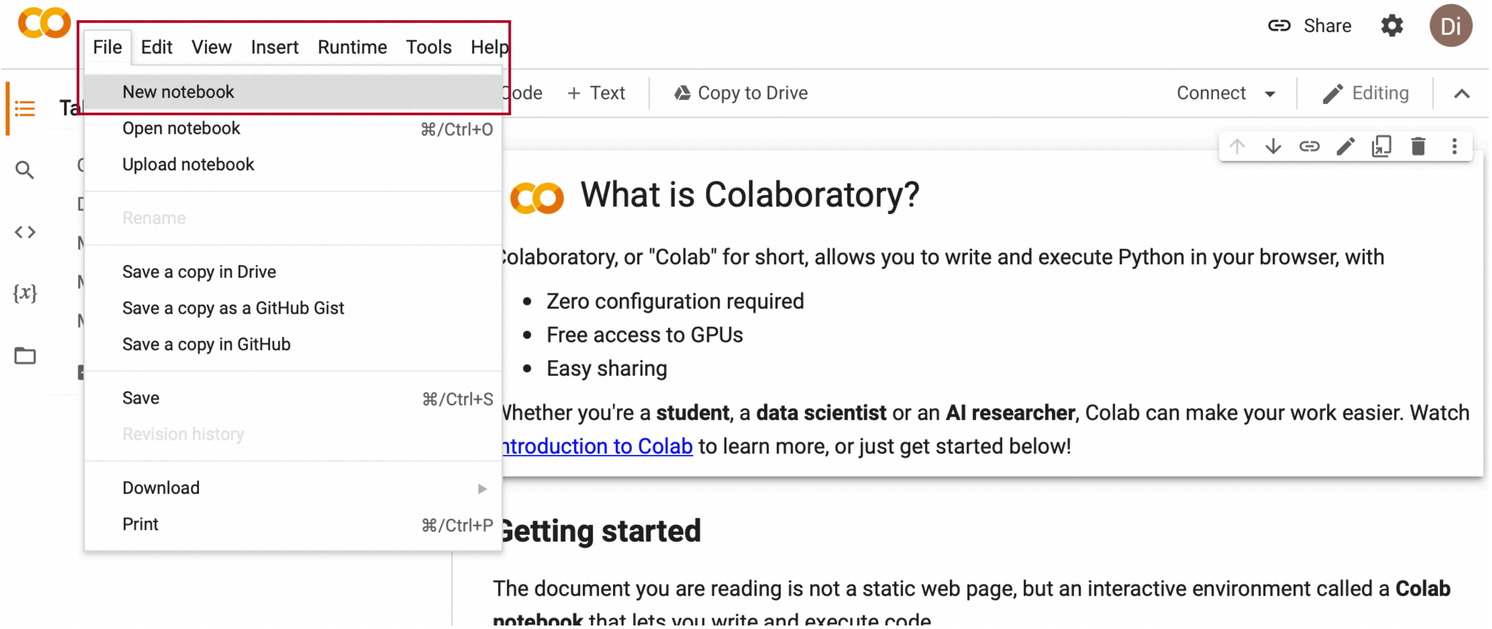
Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

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

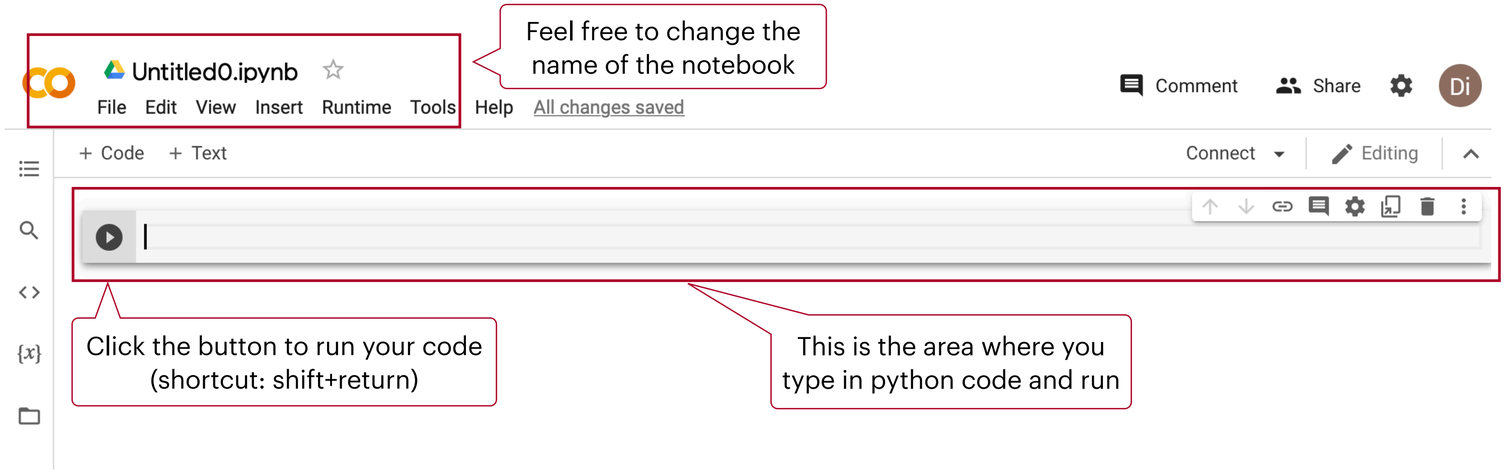
Google Colab

- Let’s create a new notebook
  - Click File, and Click New notebook



Google Colab

- Now we are ready to play!



Hello, world!

- Let’s write your first program
- Type in: print('Hello, world!')

```
print('Hello, world!')
```

- Click the button or shift+return to run

```
print('Hello, world!')
```

```
Hello, world!
```

- Congratulations! This is your first Python program!

print(‘Hello, world!’)

- Let’s look at this first program.
  - print is a keyword predefined by Python library for output.
  - () is indicating you are going to call a function.
  - 'Hello, world!' is the message you tell the function to print.

print(‘Hello, _____!’)

- Let’s play a little bit more with the message.
  - Now, you can insert a new line of code by click “+ Code”
  - Type: print('Hello, ____!') where the ____ is your first name.
    - For example, I will type: print('Hello, Di!')
  - Don’t forget there is a ' at the beginning before H, and there is a ' at the end after !
  - Run and see the result.

' and '

- Now you may have noticed that the ' and ' are crucial for the program - they wrap the message.
  - We can actually use " " or ' ' interchangeably.
  - If we start with ", we should end with "
  - If we start with ', we should end with '
  - Let's try to print another message: I'm OK!
  - Try use " " or ' ' to see which one worked.
  - print("I'm OK!")
  - print('I\'m OK!')

\ Escape

- Since the ' ' and " " we use to wrap messages are part of the Python syntax, when there are another ' or " in the message, Python will be confused.
  - We can use a different quotation to avoid this confusion.
  - We can also use a Escape mark \ to let Python know this character is something special

\ Escape

- \n Insert a new line
- \t Insert a tab
- \\ insert a \
- \" insert a "
- \' insert a '

More About Quotation Marks

- If you are going to print a message which has multiple lines, you can use triple quotation.
- In practice, triple quotations are used for commentaries and documentation.

Let’s Play in Colab

- Download the M1Lab1.ipynb file.
- Upload it to your Colab.
- Finish the tasks.
- Use the discussion board to ask for help.

Is It Too Easy?

- Let's add some spices.
- Now type in: print(input('What is your name?'))

```
print(input('What is your name?'))
```

- Run and see the result.
  - You will need to type your name; and the name will be printed out

```
print(input('What is your name?'))
...  What is your name? Di
```

```
print(input('What is your name?'))

What is your name?Di
Di
```

input(‘What is your name?’)

- Let’s look at this program.
  - input is a keyword predefined by Python library for input.
  - () is indicating you are going to call a function.
  - 'What is your name?' is the message as instruction for input.
  - The name you typed in, will be recorded for future computation.
  - In this program, the name will be used by print() and be displayed.

Let’s Play in Colab

- Download the M1Lab2.ipynb file.
- Upload it to your Colab.
- Finish the tasks.
- Use the discussion board to ask for help.

Hello, ____

- Let's revisit our first program.
  - Rather than type your name in the message directly, let's make it interactive.
  - We use input() to get the name, and use print() to print the name with the message.

```
print('Hello,', input('What is your name?'))
```

  - Run and see how the program works:

```
print('Hello,', input('What is your name?'))
What is your name? Di
```

```
print('Hello,', input('What is your name?'))
What is your name?Di
Hello, Di
```

print()

- Let’s revisit print().
  - You can have multiple messages in print()
  - Each message will be separated by ,
    - Such as print('message1', 'message2')

Let’s Play in Colab

- Download the M1Assignment1.ipynb file.
- Upload it to your Colab.
- Finish the tasks.
- Submit your assignment.
- Use the discussion board to ask for help.

Congratulations!

- You completed Module 1.
- Programming in Python is not that hard, right?
- We are going to learn more in next module and get more power unlocked!
- See you soon!