

Introduction To Python Programming



Introduction to Python programming Course Outline

- Intro to Computer Science
- Environment Setup (Anaconda)
- Command Line
- Conda & pip package managers
- Jupyter Notebook
- Input & Output
- Variables
- Data types
 - Numbers & Math
 - Boolean & Comparison and Logic
 - Strings
 - Lists
 - Tuples
 - Sets
 - Dictionaries

- File Handling
- If Conditions
- For Loops
- Built-in functions & Operators (zip, enumerate, range, ...)
- List Comprehensions
- Functions
- Lambda Expressions
- Map, Filter, Reduce
- Variables Scope
- Modules & Packages

Introduction to Python programming Course Outline



Intro to Computer Science



Environment Setup (Anaconda)



Command Line



Conda & pip package managers



Jupyter Notebook



Input & Output



Variables



Data types

- Numbers & Math
- Boolean & Comparison and Logic
- Strings
- Lists
- Tuples
- Sets
- Dictionaries



File Handling



If Conditions



For Loops



Built-in functions & Operators (zip, enumerate, range, ...)



List Comprehensions



Functions



Lambda Expressions



Map, Filter, Reduce



Variables Scope



Modules & Packages








Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



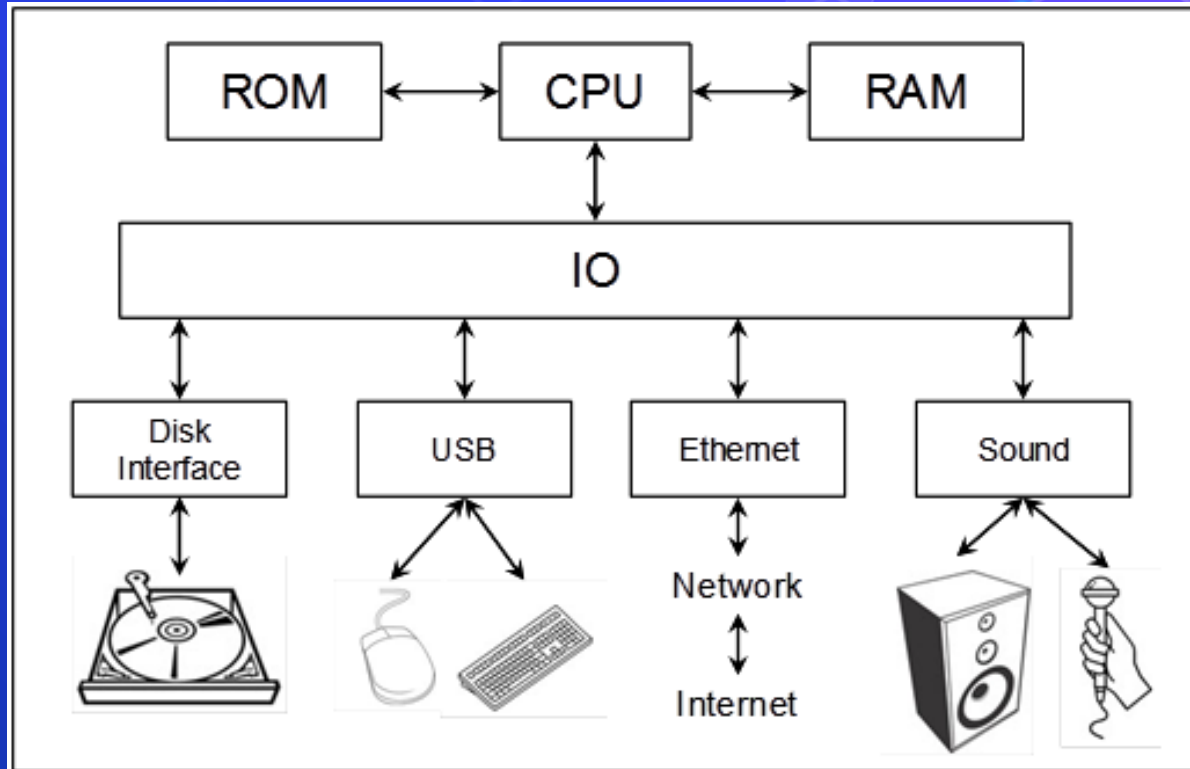
Intro to Computer Science

How Computers Work

-  Decimal & Binary numbering systems
-  How computer stores Words (ASCII)
-  How computer stores Images
-  How computer stores Videos
-  How computer stores Audio
-  Technology Tree
-  Why python



How Computers Work

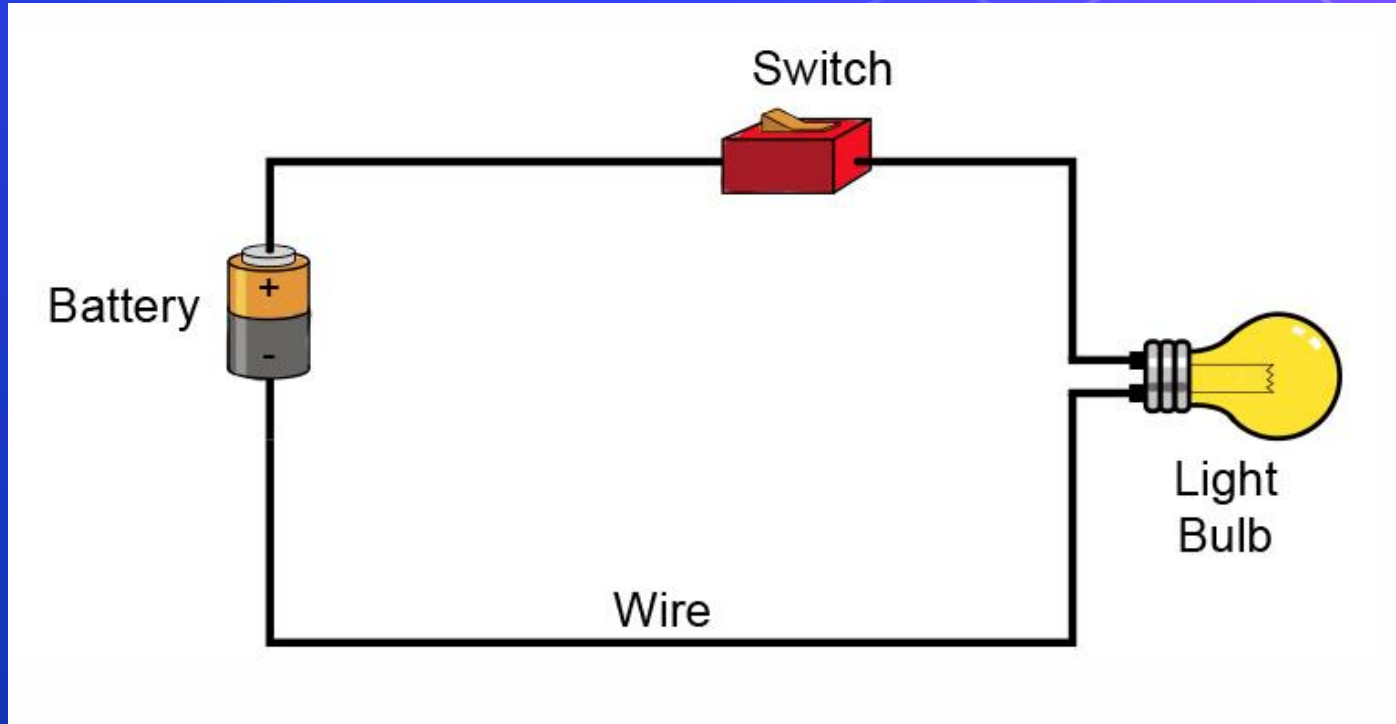


Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



Decimal & Binary numbering systems



Decimal & Binary numbering systems

Decimal

$$\begin{array}{r} 100's \quad 10's \quad 1's \\ 1 \quad 5 \quad 4 \\ 1 \times 100 = 100 \\ 5 \times 10 = 50 \\ 4 \times 1 = 4 \\ \hline 154 \end{array}$$

Binary

$$\begin{array}{r} 128's \quad 64's \quad 32's \quad 16's \quad 8's \quad 4's \quad 2's \quad 1's \\ 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \\ 1 \times 128 = 128 \\ 1 \times 16 = 16 \\ 1 \times 8 = 8 \\ 1 \times 2 = 2 \\ \hline 154 \end{array}$$

Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



How computer stores Words (ASCII)

ASCII BINARY ALPHABET

A	1000001	N	1001110
B	1000010	O	1001111
C	1000011	P	1010000
D	1000100	Q	1010001
E	1000101	R	1010010
F	1000110	S	1010011
G	1000111	T	1010100
H	1001000	U	1010101
I	1001001	V	1010110
J	1001010	W	1010111
K	1001011	X	1010111
L	1001100	Y	1011001
M	1001101	Z	1011010

Full Table

<http://www.asciitable.com/>



Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



How computer stores Images



What We See

```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08 08 02 22 97
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00 49 49 99 40
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65 81 49 31 73
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 57 02 36 91 52 70 95 23
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80 22 31 16 71
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50 24 47 32 60
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70 32 98 81 28
67 26 20 68 02 62 12 20 95 63 94 39 43 08 40 91 66 49 94 21 67 26 20 68
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 43 72 24 55 58 05
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95 21 36 23 09
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92 78 17 53 28
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57 16 39 05 42
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58 86 56 00 48
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40 19 80 81 68
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66 04 52 08 83
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69 88 36 68 87
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36 04 42 16 73
20 69 36 41 72 30 23 88 34 62 99 69 82 47 59 85 74 04 36 16 20 69 36 41
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54 20 73 35 29
01 70 54 71 88 51 54 69 16 92 33 48 61 43 52 01 89 19 47 48 01 70 54 71
```

What Computers See

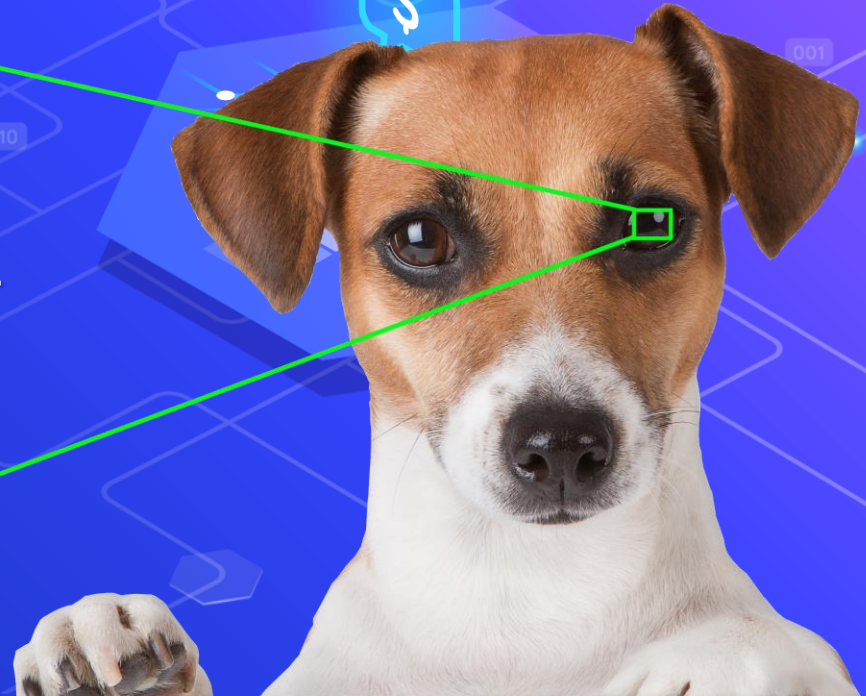
How computer stores Images

Pixels

Every pixel is one color

Height

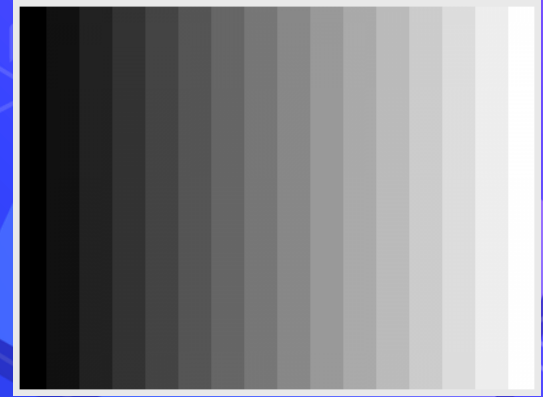
Width



How computer stores Images

Grayscale Digital Images

We can generate a grayscale image by using Pixels each pixel has a single value between 0 (White) and 255 (Black) and values in between are gray variations.



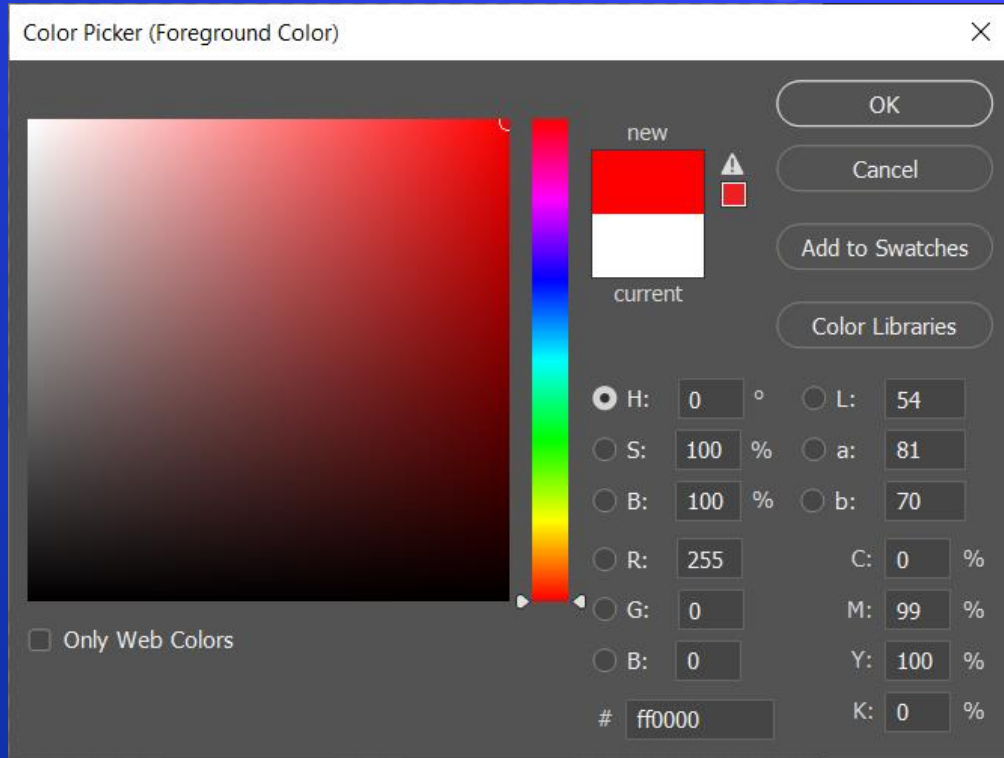
How computer stores Images

RGB Digital Images

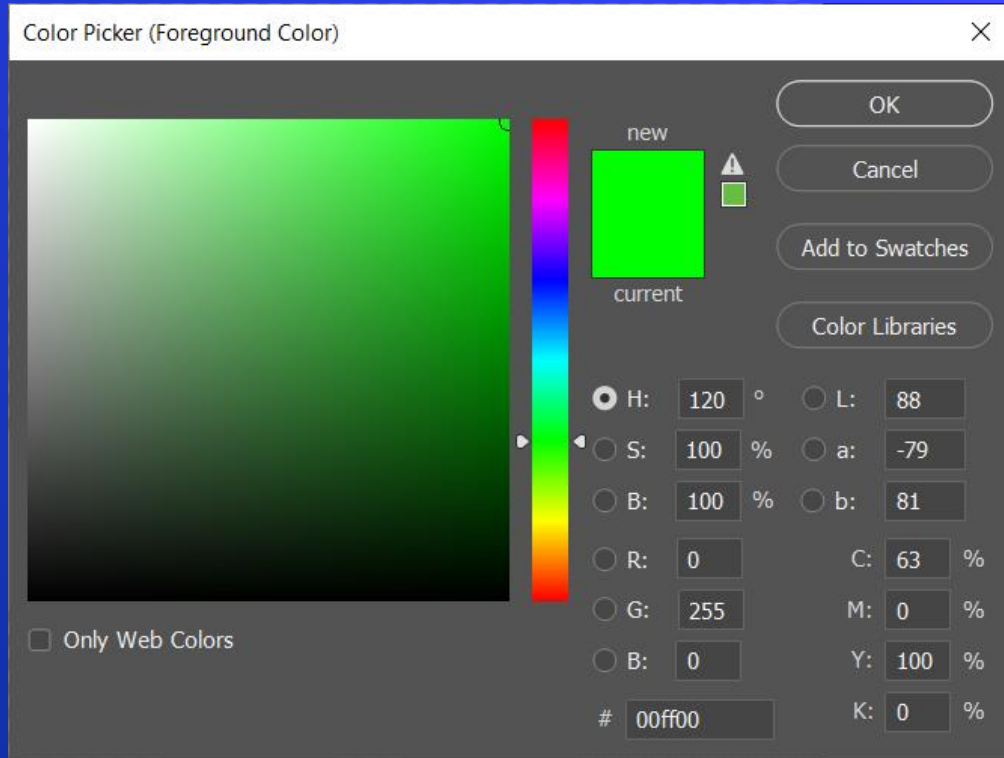
We can generate an RGB image by using Pixels each pixel has 3 values for each color (red, green, blue) each color has value between 0 (Dark) and 255 (Light) and values in between are color variations .



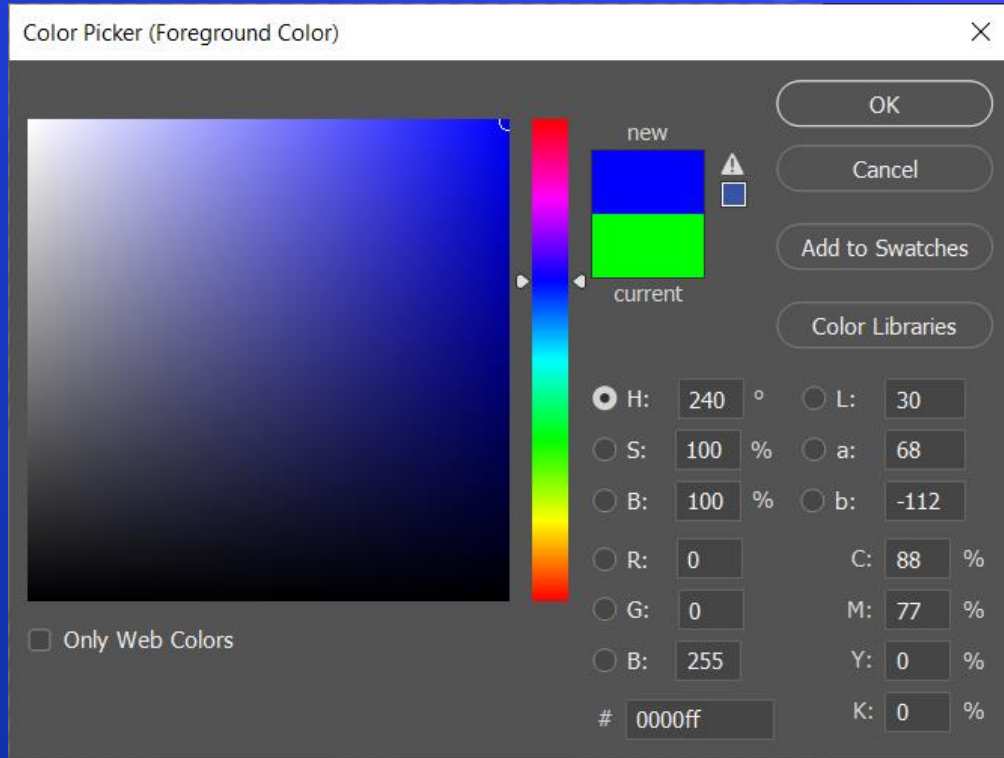
How computer stores Images (Red)



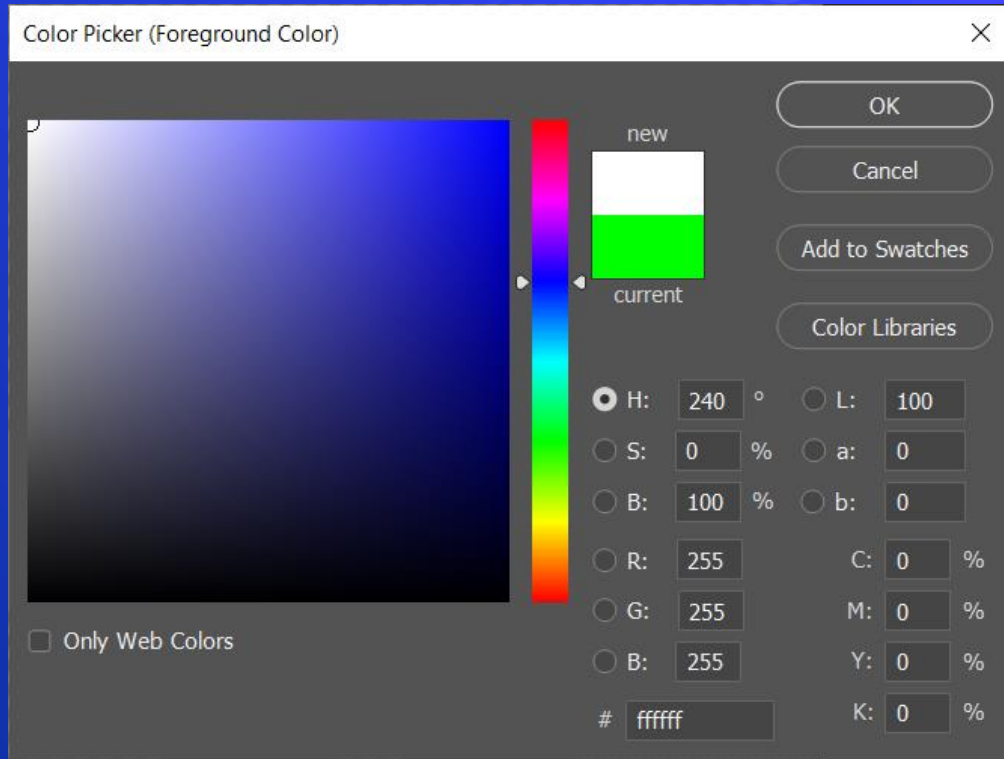
How computer stores Images (Green)



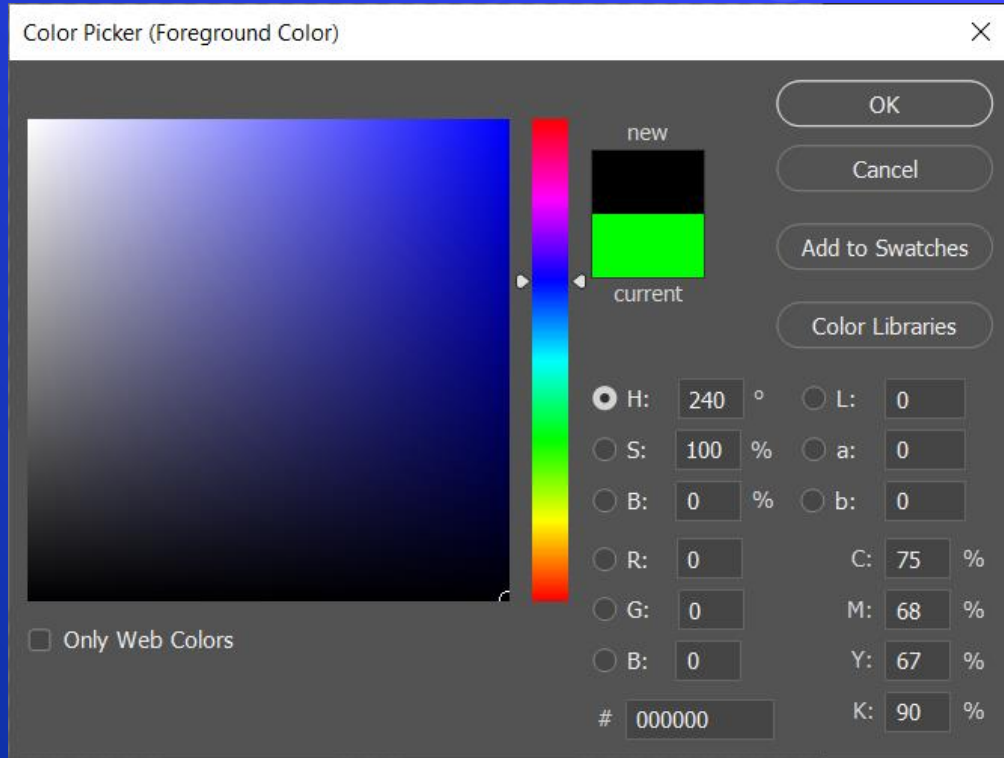
How computer stores Images (Blue)



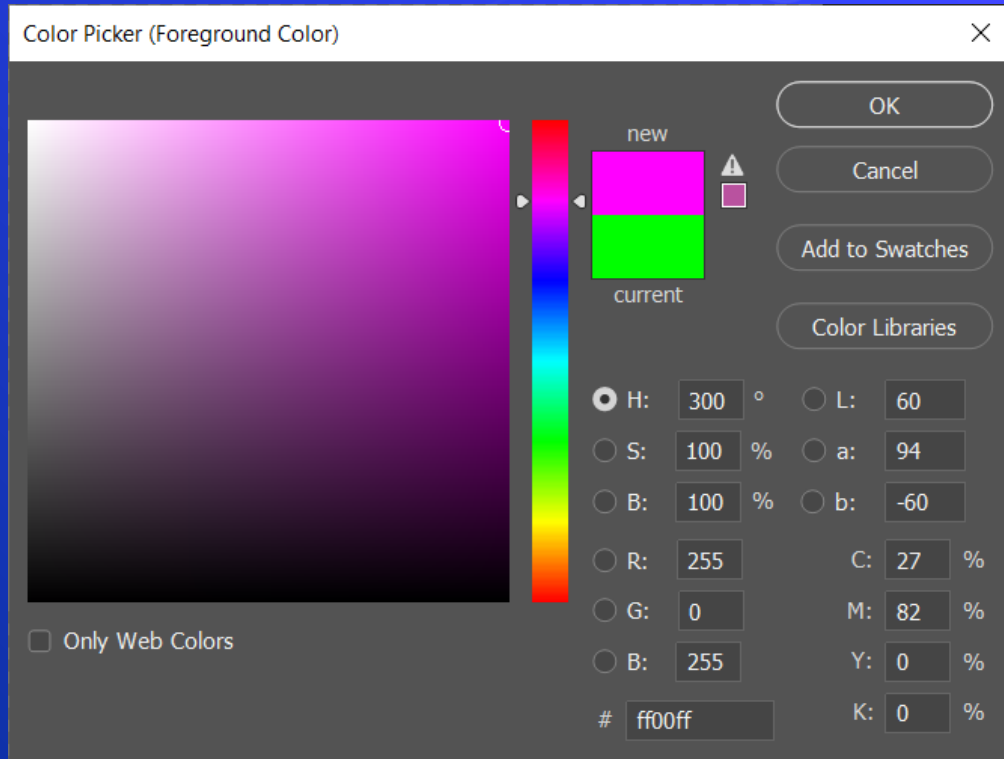
How computer stores Images (White)



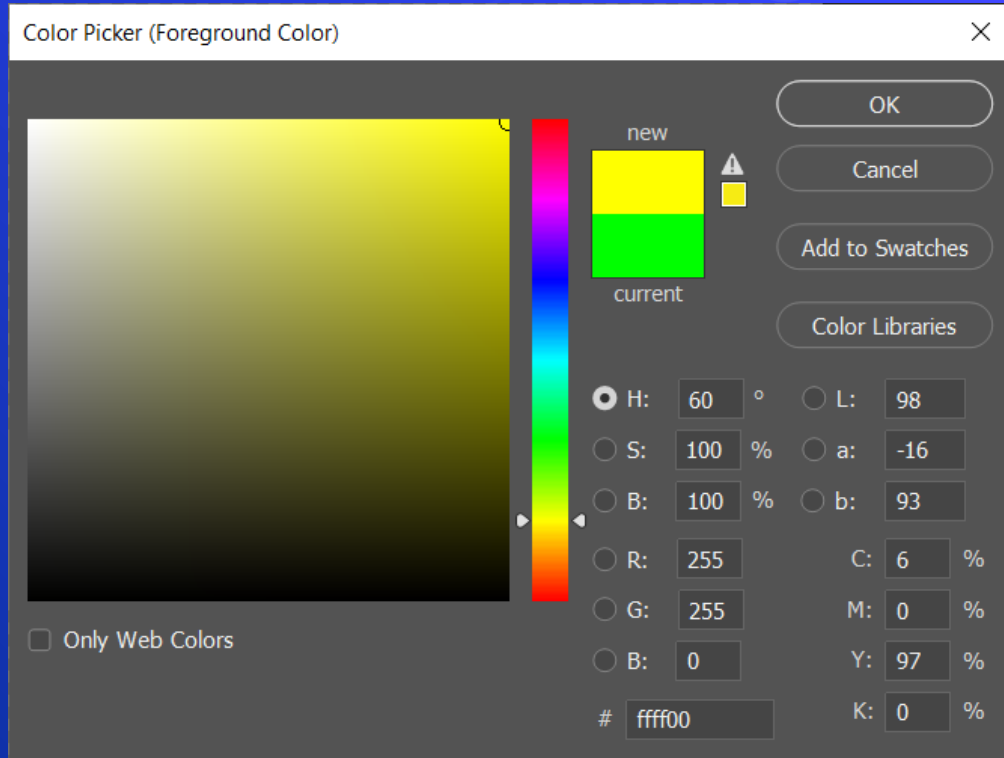
How computer stores Images (Black)



How computer stores Images (Pink)



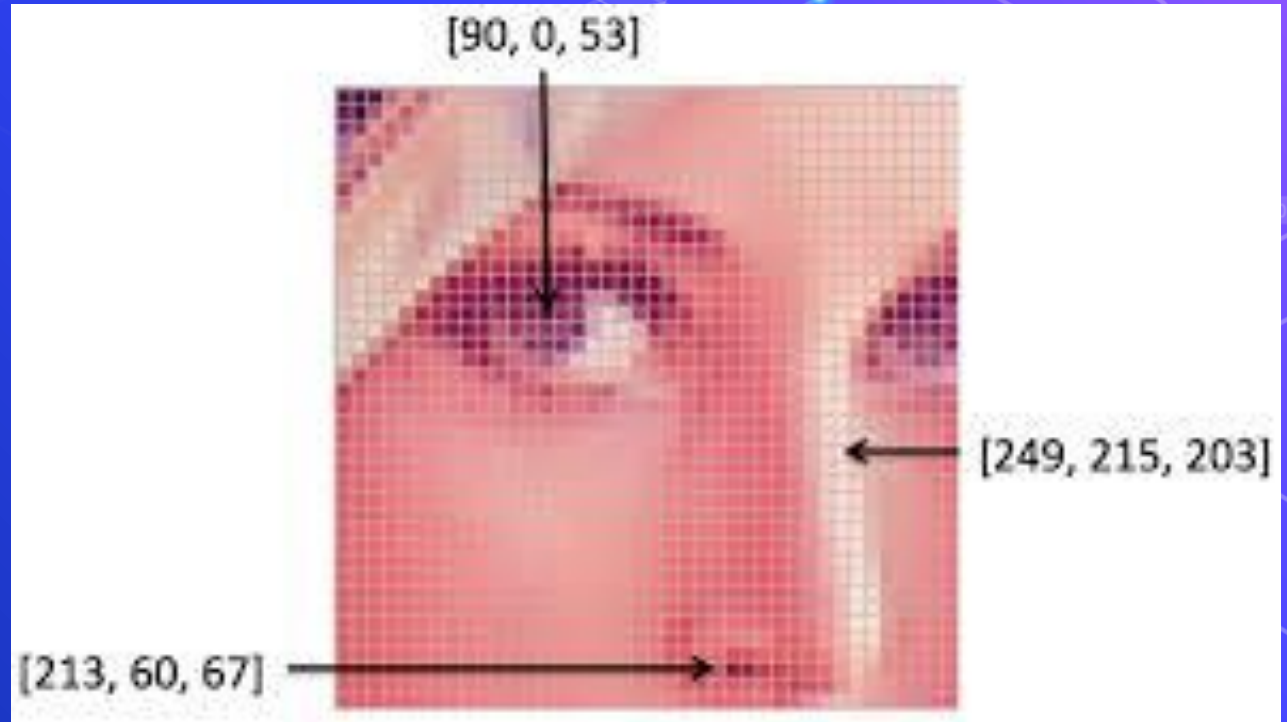
How computer stores Images (Yellow)



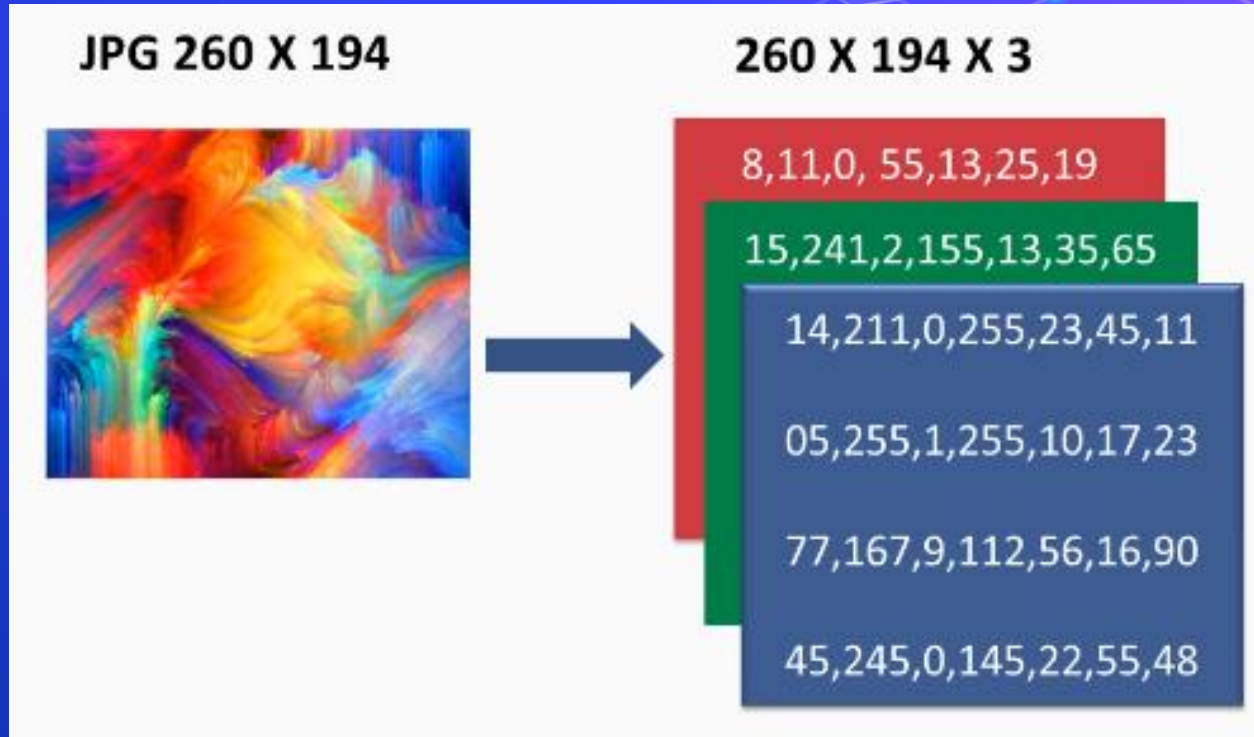
How computer stores Images

Resolution: 100x100

Width : 100 pixels
Height: 100 pixels



How computer stores Images



Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



How computer stores Videos

30 fps



10 fps



1 fps

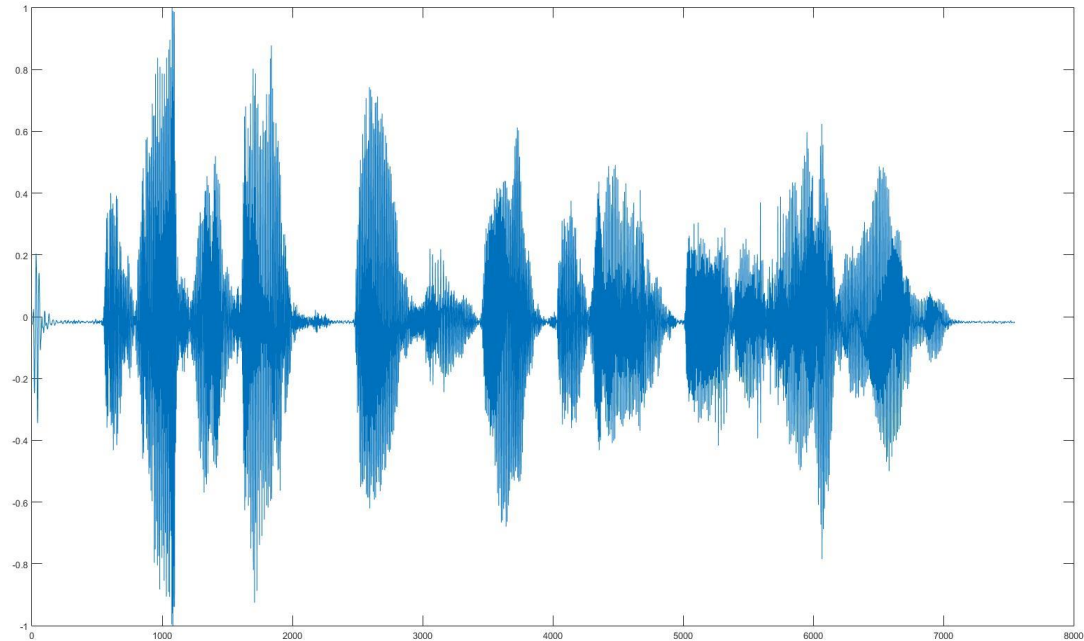


Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



How computer stores Audio



Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



Programming ?!

Just writing code for executing some sequential instructions to perform various tasks.

Computers are **FAST** but **DUMB**, they need to know what to do.



Technology Tree

- ⬡ Embedded Systems
- ⬡ Operating Systems
- ⬡ Desktop Applications
- ⬡ Web Applications
- ⬡ Mobile Applications
- ⬡ Database Systems
- ⬡ Networking & Server administration
- ⬡ Internet of Things
- ⬡ Game Development
- ⬡ AR / VR
- ⬡ Compression
- ⬡ Encryption

- ⬡ Security & Ethical Hacking
- ⬡ Machine & Deep Learning
- ⬡ Data Science
- ⬡ Computer Vision
- ⬡ Speech Processing
- ⬡ Natural Language Processing
- ⬡ Autonomous
- ⬡ Blockchain
- ⬡ Big Data
- ⬡ Computer Graphics
- ⬡ Compiler Design
- ⬡ ...

Intro to Computer Science

- ⬡ How Computers Work
- ⬡ Decimal & Binary numbering systems
- ⬡ How computer stores Words (ASCII)
- ⬡ How computer stores Images
- ⬡ How computer stores Videos
- ⬡ How computer stores Audio
- ⬡ Technology Tree
- ⬡ Why python



Why Python ?!

>_ Easy to Learn, Read, Maintain.

>_ Very Big Community so you will find a lot of Libraries to use.



Python 2 vs 3 ?!



Introduction to Python programming Course Outline

- Intro to Computer Science
- Environment Setup (Anaconda)**
- Command Line
- Conda & pip package managers
- Jupyter Notebook
- Input & Output
- Variables
- Data types
 - Numbers & Math
 - Boolean & Comparison and Logic
 - Strings
 - Lists
 - Tuples
 - Sets
 - Dictionaries

- File Handling
- If Conditions
- For Loops
- Built-in functions & Operators (zip, enumerate, range, ...)
- List Comprehensions
- Functions
- Lambda Expressions
- Map, Filter, Reduce
- Variables Scope
- Modules & Packages

ANACONDA

Awesome Python Distribution.

Free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications) . Anaconda distribution includes data-science packages suitable for Windows, Linux, and macOS.

<https://www.anaconda.com>



Visual Studio Code

Awesome Code Editor.

<https://code.visualstudio.com>



Introduction to Python programming Course Outline

- Intro to Computer Science
- Environment Setup (Anaconda)
- Command Line
- Conda & pip package managers
- Jupyter Notebook
- Input & Output
- Variables
- Data types
 - Numbers & Math
 - Boolean & Comparison and Logic
 - Strings
 - Lists
 - Tuples
 - Sets
 - Dictionaries

- File Handling
- If Conditions
- For Loops
- Built-in functions & Operators (zip, enumerate, range, ...)
- List Comprehensions
- Functions
- Lambda Expressions
- Map, Filter, Reduce
- Variables Scope
- Modules & Packages

Command Line

>_ *cd*

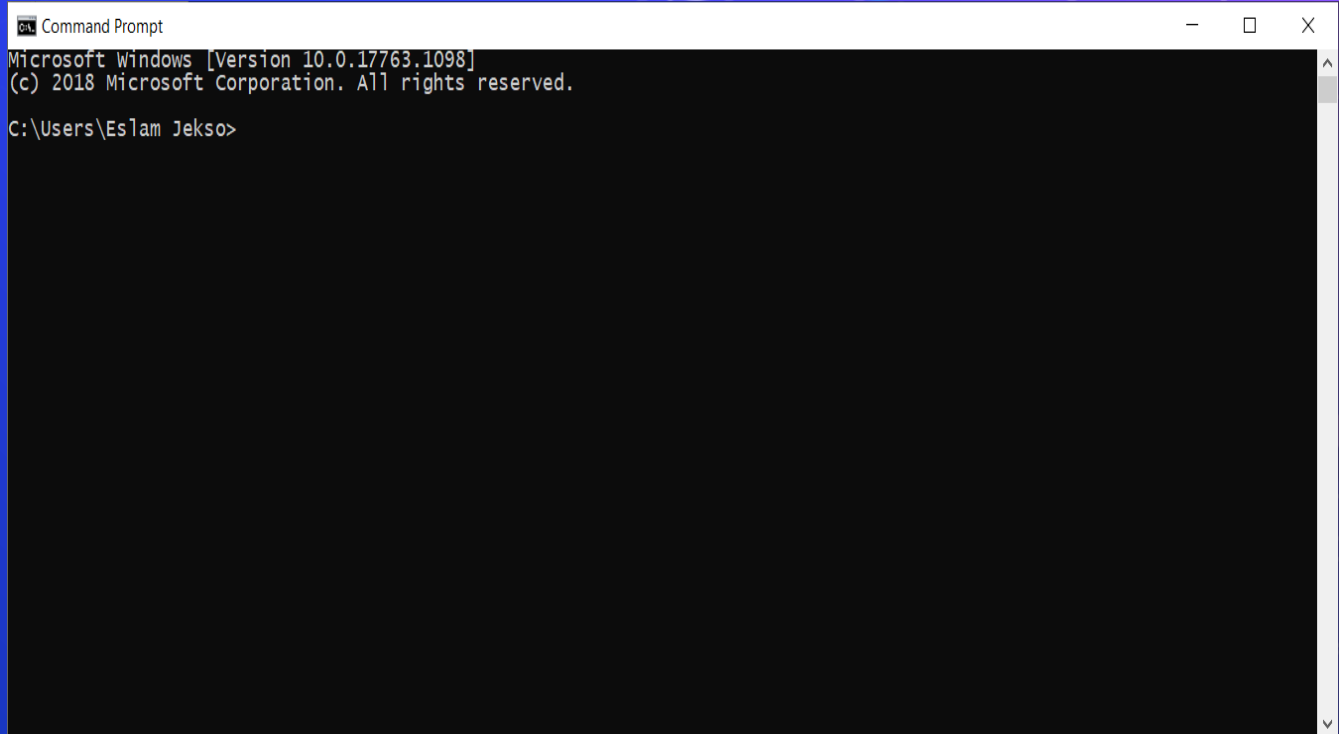
>_ *dir*

>_ *copy*

>_ *del*

>_ *move*

...

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the following text: "Microsoft Windows [Version 10.0.17763.1098]", "(c) 2018 Microsoft Corporation. All rights reserved.", and "C:\Users\Eslam Jekso>". The prompt is at the end of the line, ready for input. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Command Prompt
Microsoft Windows [Version 10.0.17763.1098]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Eslam Jekso>
```


Introduction to Python programming Course Outline

- Intro to Computer Science
- Environment Setup (Anaconda)
- Command Line
- Conda & pip package managers**
- Jupyter Notebook
- Input & Output
- Variables
- Data types
 - Numbers & Math
 - Boolean & Comparison and Logic
 - Strings
 - Lists
 - Tuples
 - Sets
 - Dictionaries

- File Handling
- If Conditions
- For Loops
- Built-in functions & Operators (zip, enumerate, range, ...)
- List Comprehensions
- Functions
- Lambda Expressions
- Map, Filter, Reduce
- Variables Scope
- Modules & Packages

conda & pip package managers

<https://anaconda.org/>

```
1 conda install --package name--
```

<https://pypi.org/>

```
1 pip install --package name--
```



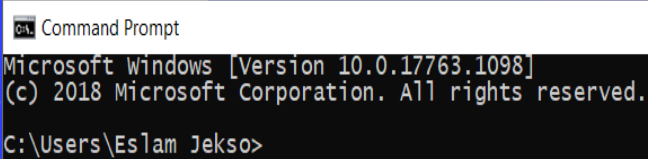
Introduction to Python programming Course Outline

- Intro to Computer Science
- Environment Setup (Anaconda)
- Command Line
- Conda & pip package managers
- Jupyter Notebook
- Input & Output
- Variables
- Data types
 - Numbers & Math
 - Boolean & Comparison and Logic
 - Strings
 - Lists
 - Tuples
 - Sets
 - Dictionaries

- File Handling
- If Conditions
- For Loops
- Built-in functions & Operators (zip, enumerate, range, ...)
- List Comprehensions
- Functions
- Lambda Expressions
- Map, Filter, Reduce
- Variables Scope
- Modules & Packages

Run Python Script via Command Line

- 1- Make a .py file
- 2- Write code & Save it
- 3- Open cmd
- 4- >_ *python file.py*

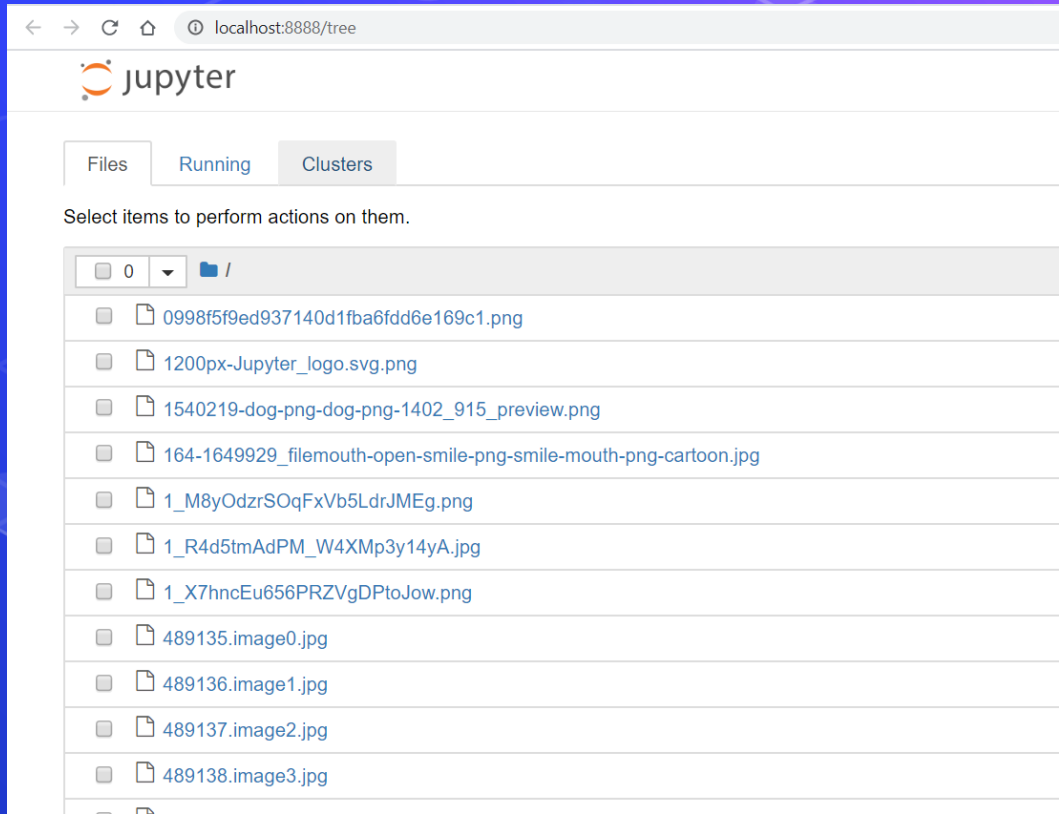
A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the standard Windows boot-up text: "Microsoft Windows [Version 10.0.17763.1098]" and "(c) 2018 Microsoft Corporation. All rights reserved." Below this, the current directory is shown as "C:\Users\Eslam Jekso>".

```
Command Prompt
Microsoft Windows [Version 10.0.17763.1098]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Eslam Jekso>
```

Run Python via Jupyter Notebook

1- Open cmd in a folder

2- `>_ jupyter notebook`



Jupyter Notebook

- ⬡ Create new .ipynb file
- ⬡ Naming the notebook
- ⬡ Menu buttons (Run, Insert, Delete cells, etc...)
- ⬡ Move Cell up or down
- ⬡ Copy, Paste and Cut Cells
- ⬡ Merge Cells
- ⬡ Saving the notebook for checkpoints
- ⬡ Code and Markdown Cells
- ⬡ Export .py file
- ⬡ Kernel
- ⬡ Use command line in Jupyter using '!' operator



Jupyter Notebook (Shortcuts)

- ⬡ *Ctrl + Enter* --- > Execute Cell
- ⬡ *Shift + Enter* --- > Execute Cell then go to the next cell
- ⬡ *Alt + Enter* --- > Execute Cell then insert new cell below
- ⬡ *A and B* --- > Insert Cell Above or Below
- ⬡ *Shift + Up or Down* --- > Select Cells Above or Below
- ⬡ *C and V and X* --- > Copy, Paste and Cut Cells inside Notebook
- ⬡ *Ctrl + C or V or X* --- > Copy, Paste and Cut Cells outside Notebook
- ⬡ *Double D* --- > Delete Cells
- ⬡ *Shift + M* --- > Merge Cells
- ⬡ *Y and M* --- > Make Cell type Code or Markdown
- ⬡ *S* --- > Save Notebook



Questions ?!



Thanks!

>_ Live long and prosper

