סיכום שיעור ראשון:

1. **? What is DevOps**

- First of all DevOps is not a title/job/profession/set of tools or software, DevOps is a culture!
- A culture where Dev and Ops teammates cooperate with each other.
- DevOps culture is talking about:
  - Engineers empowerment - by giving engineers more responsibility over the whole application lifecycle process. (dev -> test -> deploy -> monitor)
  - Test Driven Development - write tests before you write code. Unit tests, integration tests, system tests. This will help increase the quality of your service and give you more confidence to release faster and more frequent.
  - Automation - automate everything that can be automated. Test automation, infrastructure automation, deployment automation, etc..
  - Monitoring - monitor your apps, build monitoring alerts well. It should save your time, don't flood with metrics and alerts.
  - Self service - provide a self service for any framework that you build or anything that you do. Don't be a bottleneck.

Benefits:

- Speed
  - Ensure faster time-to-market/delivery times that improves ROI.
  - DevOps is basically the application of Agile principles, and thus the end result is faster development of software, ensuring more frequent delivery.
- Reliability
  - Early detection and faster correction of defects that helps provide the best services and robust features that must be delivered to customers.
- Cultural
  - Happier, more productive teams
  - Higher employee engagement
  - Improved communication and collaboration

Methodologies .2

Waterfall:

- Gather and document requirements
- Design
- Code and unit test
- Perform system testing
- Perform user acceptance testing
- Fix any issues
- Deliver the finished product
- The biggest problem with this approach is that the customer's needs usually change during the development phase which causes delivering software that doesn't meet the revised needs, or you spend a lot of time and money in changing plans midway.

Agile:

- The idea in Agile is to develop software in small iterations (known as iterations) and be able to adapt to the changing customer needs better than in Waterfall.
- However, some of this model disadvantages are:
- Budget goals and deadlines are often missed.
- New features break old functions.

Agile & DevOps

- DevOps brings more flexibility on top of the Agile model.
- With continuous integration (CI) and continuous delivery (CD) pipelines, you can make sure that you can release often and the releases actually work and meet the customer needs.
- Cooperation between the development team and IT operations ensures that also the used tools are streamlined and do not form bottlenecks.
- With the effective tools, repetitive work can be automated and transparency is improved.

CI (Continuous integration) .3

- Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day.
- Each integration can then be verified by an automated build and automated tests.
- One of the key benefits of integrating regularly is that you can detect errors quickly and locate them more easily.
- As each change introduced is typically small, pinpointing the specific change that introduced a defect can be done quickly.
- In recent years CI has become a best practice for software development and is guided by a set of key principles. Among them are revision control, build automation and automated testing.
- Some popular tools used in CI which will be discussed later are:
  - Jenkins
  - Travis CI
  - Bamboo

CD(Continuous Delivery) .4

- Continuous delivery is an extension of continuous integration to make sure that you can release new changes to your customers quickly in a sustainable way.
- This means that on top of having automated your testing, you also have automated your release process and you can deploy your application at any point of time by clicking on a button.
- In theory, with continuous delivery, you can decide to release daily, weekly, or whatever suits your business requirements.
- However, if you truly want to get the benefits of continuous delivery, you should deploy to production as early as possible to make sure that you release small batches, that are easy to troubleshoot in case of a problem.

CD(Continuous Deployment) .5

- Continuous deployment goes one step further than continuous delivery.
- With this practice, every change that passes all stages of your production pipeline is released to your customers.
- There's no human intervention, and only a failed test will prevent a new change to be deployed to production.
- Continuous deployment is an excellent way to accelerate the feedback loop with your customers and take pressure off the team as there isn't a Release Day anymore.
- Developers can focus on building software, and they see their work go live minutes after they've finished working on it.

6. מבוא לPython :

Python is a clear and powerful object-oriented multi-purpose programming language. Some features of Python are:

a. Python supports object-oriented programming with classes and multiple inheritance.
b. Code can be grouped into modules and packages.
c. Python's automatic memory management frees you from having to manually allocate and free memory in your code.
d. Runs anywhere, including Mac OS X, Windows, Linux, and Unix, with unofficial builds also available for Android and iOS.
e. It is free open-source language

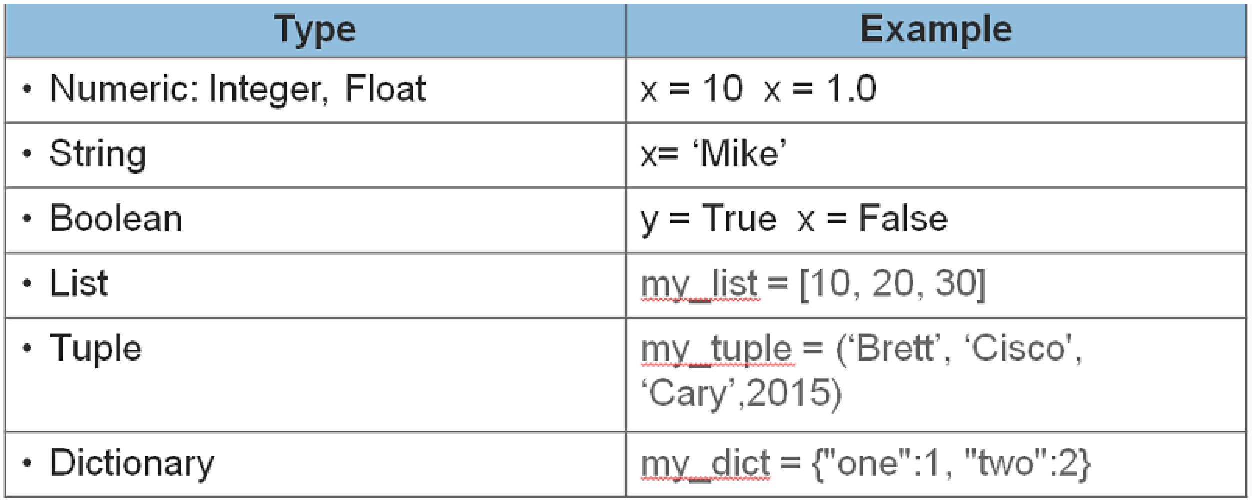
7. Python Interpreter

- Python is an interpreted language which means that our code works as such:
  - Instructions (code) are read
  - Code evaluated
  - Result is returned for each instruction in a sequence.
- And this is the interpreter job.
- Unlike compiled languages, in which whole context is passed through a compiler, which in return gives a translational unit.
- The Standard interpreter for Python is written in C.
- There is also a version of Python written in Python, which instead of Python bytecode it generates Python byte code (Jython).
- One of the advanteages is the increased productivity it provides, since there is no compilation step unlike many other languages.

8. :Data Types

data type הוא סיווג נתונים, אשר יכול לאחסן סוג מסוים של מידע.

- Primitive data types
- להלן טבלת המשתנים הפרימיטיביים הקיימים ב Python



Type	Example
• Numeric: Integer, Float	x = 10 x = 1.0
• String	x= ‘Mike’
• Boolean	y = True x = False
• List	my_list = [10, 20, 30]
• Tuple	my_tuple = (‘Brett’, ‘Cisco’, ‘Cary’,2015)
• Dictionary	my_dict = {"one":1, "two":2}

- Numbers –
  - int – whole number: **X = 2**
  - float – a floating point number: **y = 2.2**
  - long – are large whole numbers (happens automatically) when a number is big.
- Boolean –
  - Can hold one value out of the following two values: True/False **isHere = True**
- String –
  - A string is a sequence of characters, which is marked in a single/ double quotes:

    Name = “John”
- Data structures-
  - List, Tuple, Dictionary will be discussed later.

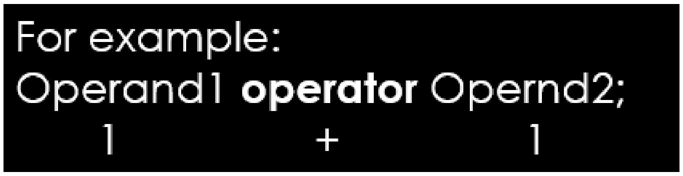
    https://en.wikibooks.org/wiki/Python_Programming/Variables_and_Strings

9. Data Types:

- Operators are special symbols that perform specific operations on one, two, or three operands, and then return a result.
- Operator Types:

a. Arithmetic operators

b. Comparison operators



For example:
Operand1 **operator** Opernd2;
1 + 1

c. In Python each **operator** should have space **before** and **after**

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html

10.**אופרטורים:**

אופרטורים הם סמלים מיוחדים המבצעים פעולות ספציפיות על אחד, שניים או שלושה אופרנדים, ולאחר מכן מחזירים תוצאה.

Simple Assignment Operator

```
=       Simple assignment operator
```

Arithmetic Operators

```
+       Additive operator (also used
        for String concatenation)
-       Subtraction operator
*       Multiplication operator
/       Division operator
%       Remainder operator
```

Equality and Relational Operators

```
==      Equal to
!=      Not equal to
>       Greater than
>=      Greater than or equal to
<       Less than
<=      Less than or equal to
```

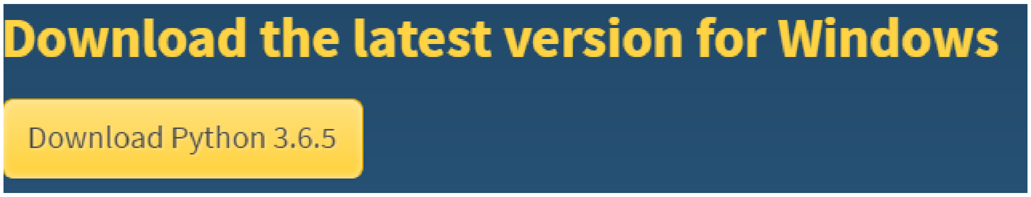
https://www.w3schools.com/python/python_operators.asp

11.**התקנות:**

- התקנת Python.
- התקנת IDE (PyCharm).

Python Installation

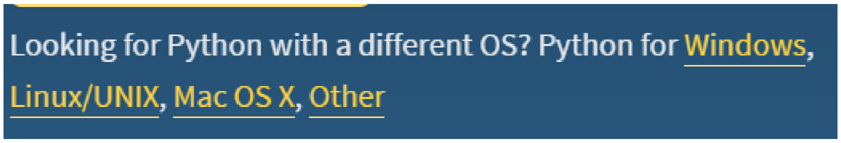
1. Enter Python download page:
   https://www.python.org/downloads/
2. Click yellow button "Download Python.." for windows



Download the latest version for Windows

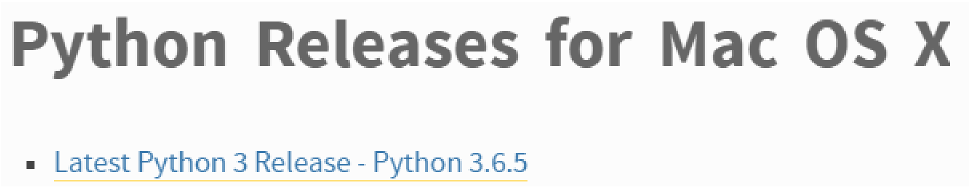
Download Python 3.6.5

For other OS's, click the corresponding link:



Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

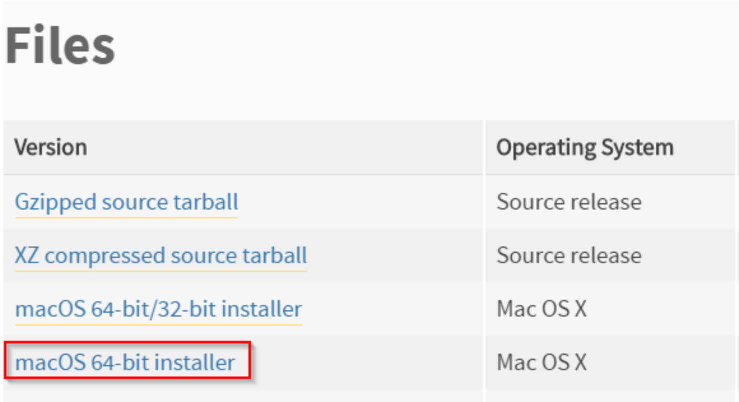
Pick latest release:



Python Releases for Mac OS X

- Latest Python 3 Release - Python 3.6.5

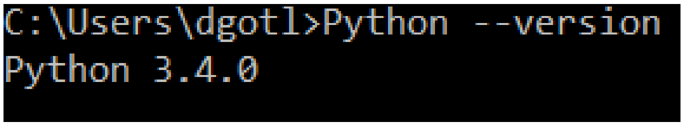
Then scroll down, until you find the version:



Files

Version	Operating System
Gzipped source tarball	Source release
XZ compressed source tarball	Source release
macOS 64-bit/32-bit installer	Mac OS X
macOS 64-bit installer	Mac OS X

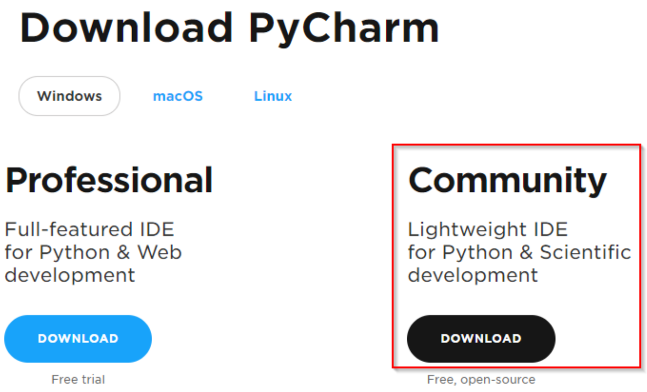
3. Validate Python is installed by doing the following steps:
   A. Open CMD / Terminal.
   B. Type python --version
   C. Python version will appear at command line.



```
C:\Users\dgotl>Python --version
Python 3.4.0
```

12. IDE (PyCharm) download and setup:

1. Go to PyCharm download page: https://www.jetbrains.com/pycharm/download/#section=windows
2. Press download under Community:



Download PyCharm

Professional

Full-featured IDE for Python & Web development

Free trial

Community

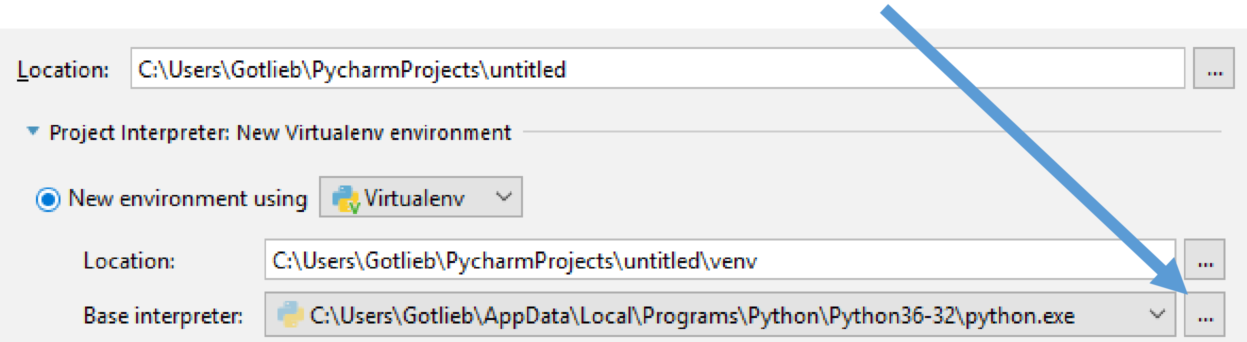
Lightweight IDE for Python & Scientific development

Free, open-source

3. Follow the installation with defaults.
4. You will be asked to agree the terms, and send statistics if you want
5. Mark the installer based on your system 32/64 bit
6. Mark Download and install JRE X86..

7. Once Pycharm will start, press on “**Create New Project**” in the welcome screen, and give it a name (does not matter):

8. Next you will be asked to choose the Python interpreter.
9. Simply press the 3 dots at the end of the line of “**Base Interpreter**”



Location: C:\Users\Gotlieb\PycharmProjects\untitled ...

Project Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: C:\Users\Gotlieb\PycharmProjects\untitled\venv ...

Base interpreter: C:\Users\Gotlieb\AppData\Local\Programs\Python\Python36-32\python.exe ...

10. And choose the file **Python.exe** from Python installation path

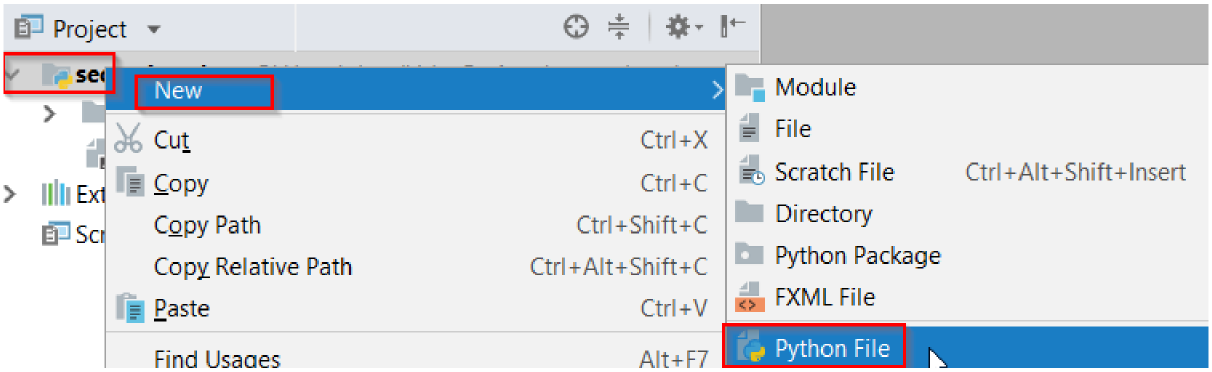
11. The default is:
C:\users\<USER_NAME>\Appdata\Local\Python\Python36-32\Python.exe

12. That’s it.. We are officially ready to start Python development!

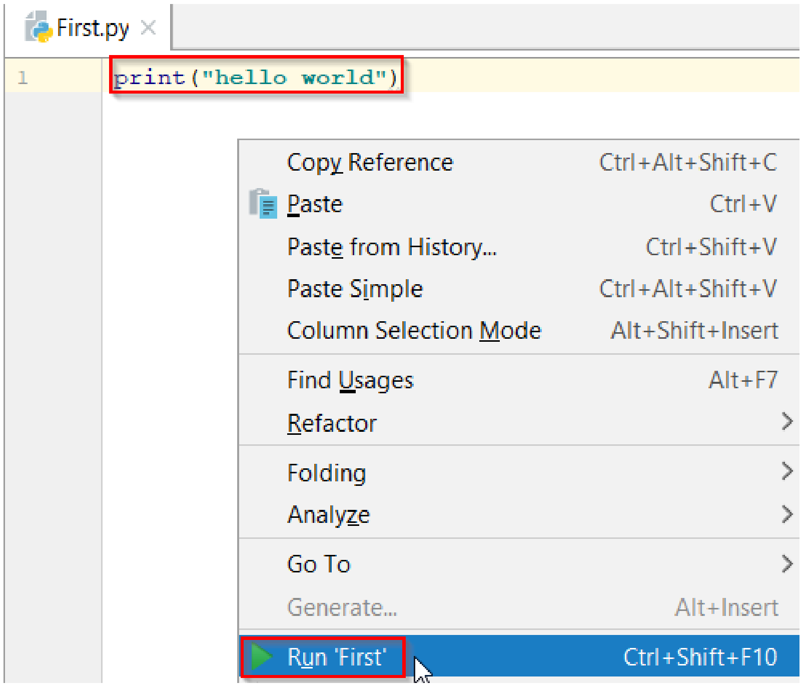
13. **קוד:**

- יצירת Python file
- הדפסת נתונים למסך.

1. To create a new Python file: Right click on the project folder→ New→ Python file:



2. Type a name to your file(any)
3. Type: print("your_name")
4. Right click → Run <your_file_name>
5. Let's start coding…


First.py
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Paste from History... Ctrl+Shift+V
Paste Simple Ctrl+Alt+Shift+V
Column Selection Mode Alt+Shift+Insert
Find Usages Alt+F7
Refactor
Folding
Analyze
Go To
Generate... Alt+Insert
Run 'First' Ctrl+Shift+F10

```
print("hello world")
```

תזכורת ממה שעשינו בכיתה:

1. הכרזה על משתנים.

2. שינוי ערך של משתנה y

3. פעולות אריתמטיות

4. הדפסת ערכים(פלט)

5. הוספת הערות לקוד

```
y=12              1

y=45              2

d = 44.3+12       3

c = d

t = 1+2*6

print(d+y)        4

#print(c*y)       5
```