

## Упражнение 1 – Запознаване със средата за разработване на програми на Python интерпретатор, настройка и изпълнение на програма

---

### Какво е Python ?

Python е обектно-ориентиран език за програмиране на високо ниво с динамична семантика. Вградените в него структури от данни на високо ниво, съчетани с динамично въвеждане и динамично свързване, го правят много привлекателен за бързото разработване на приложения, както и за използване като скриптов език. Опростеният и лесен за изучаване синтаксис на Python подчертава четливостта и следователно намалява разходите за поддръжка на програмата. Python поддържа различни модули и библиотеки, което дава възможност за повторно използване на кода. Интерпретаторът на Python и стандартната библиотека към него са със свободен достъп.

Python осигурява повишена производителност. Тъй като липсва етапа за компилация, цикълът редактиране (edit)-тестване(test)-отстраняване на грешки (debug) е невероятно бърз. Отстраняването на грешки в програмите на Python е лесно: сгрешени или некоректни входни данни никога няма да причинят грешка в сегментирането. Вместо това, когато интерпретаторът открие грешка, той поражда изключение. Когато програмата не засече изключението, интерпретаторът отпечатва проследяване на стека. Отстраняването на грешки на ниво източник позволява проверка на локални и глобални променливи, оценка на произволни изрази, задаване на точки на прекъсване (breakpoints). Програмата, занимаваща се с отстраняването на грешки в Python(debugger) е написана на самия Python език.

Програмите на Python са файлове с разширение .py. В езика Python не се слага точка и запетая (;) в края на командите. Командите се разделят една от друга чрез нов ред или интервал.

## Какво е интерпретатор (Python interpreter)?

Какво представлява една програма на Python?- програмата е последователност от команди, които указват на компютъра какво да прави, за да се постигне даден резултат.

Докато пишем код в интерпретатора на Python, компютърът всъщност не разбира езика Python на най-ниско ниво. За това интерпретаторът на Python превежда кода на език, който компютърът разбира- нули и единици, битове и байтове, с които компютрите действително работят. Този процес се извършва всеки път, когато се стартира скрипт на Python.

Има редица имплементации на Python- CPython- референтната имплементация на езика Python, Jython- написан на Java, PyPy- написан на Python, който предлага подобрена скорост и запознае на паметта, IronPython- насочен към .NET платформата на Microsoft и MicroPython, който работи на микроконтролери.

## Как функционира Python?

Първо, той обработва изявленията (statements) на скрипта и проверява за правилен синтаксис. След това компилира сорс код в байт код - междинна, независима от платформата, версия на кода, който е написан. След това изпраща този код за изпълнение към виртуалната машина Python, която всъщност изпълнява кода на най-ниското ниво.

## Какво представлява “IDE” и “Code Editor”?

IDE (Integrated Development Environment) е програма, предназначена за разработването на софтуер. Разполага с различни инструменти, специално предназначени за разработването на софтуер. Тези инструменти обикновено включват:

- Редактор, предназначен за обработка на код (подчертаване на синтаксиса и автоматично попълване);
- Инструменти за създаване, изпълнение и отстраняване на грешки

Повечето поддържат много различни езици за програмиране и съдържат много функции. Следователно те могат да бъдат с големи размери и да отнемат време за изтегляне и инсталиране.

За разлика от интегрираната среда за разработване, отделният редактор на код може да бъде толкова прост, колкото текстов редактор с възможност за

отбелязване на синтаксиса и форматиране на кода. Повечето добри редактори могат да изпълняват код и да използват дебъгер (debugger). Най-добрите взаимодействат и със “source control” системи. В сравнение с “IDE”, добрият редактор на код обикновено е по-малък като размер и по-бърз, но често с по-малко на брой функции.

### ➤ Изисквания за добра среда за писане на код на Python

Основни функции, които улесняват кодирането:

- Запазване и презареждане на файлове;
- Изпълнение на код в програмната среда;
- Отстраняване на грешки (Debugger);
- Подчертаване на синтаксис;
- Автоматично форматиране на код;

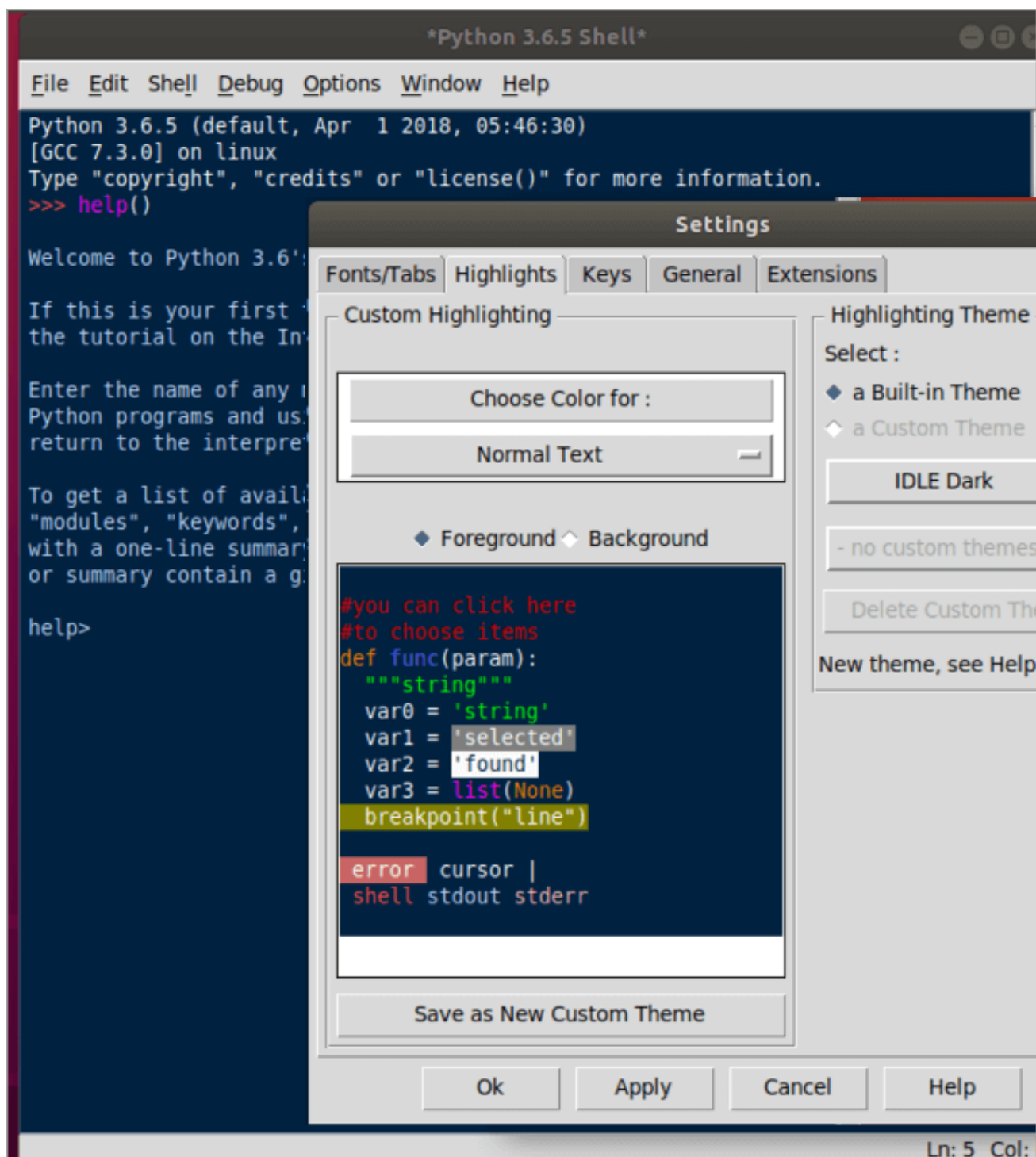
За нуждите на лабораторните упражнения ще се запознаем със следната среда за разработване (IDE) и редактор на код (Code Editor).

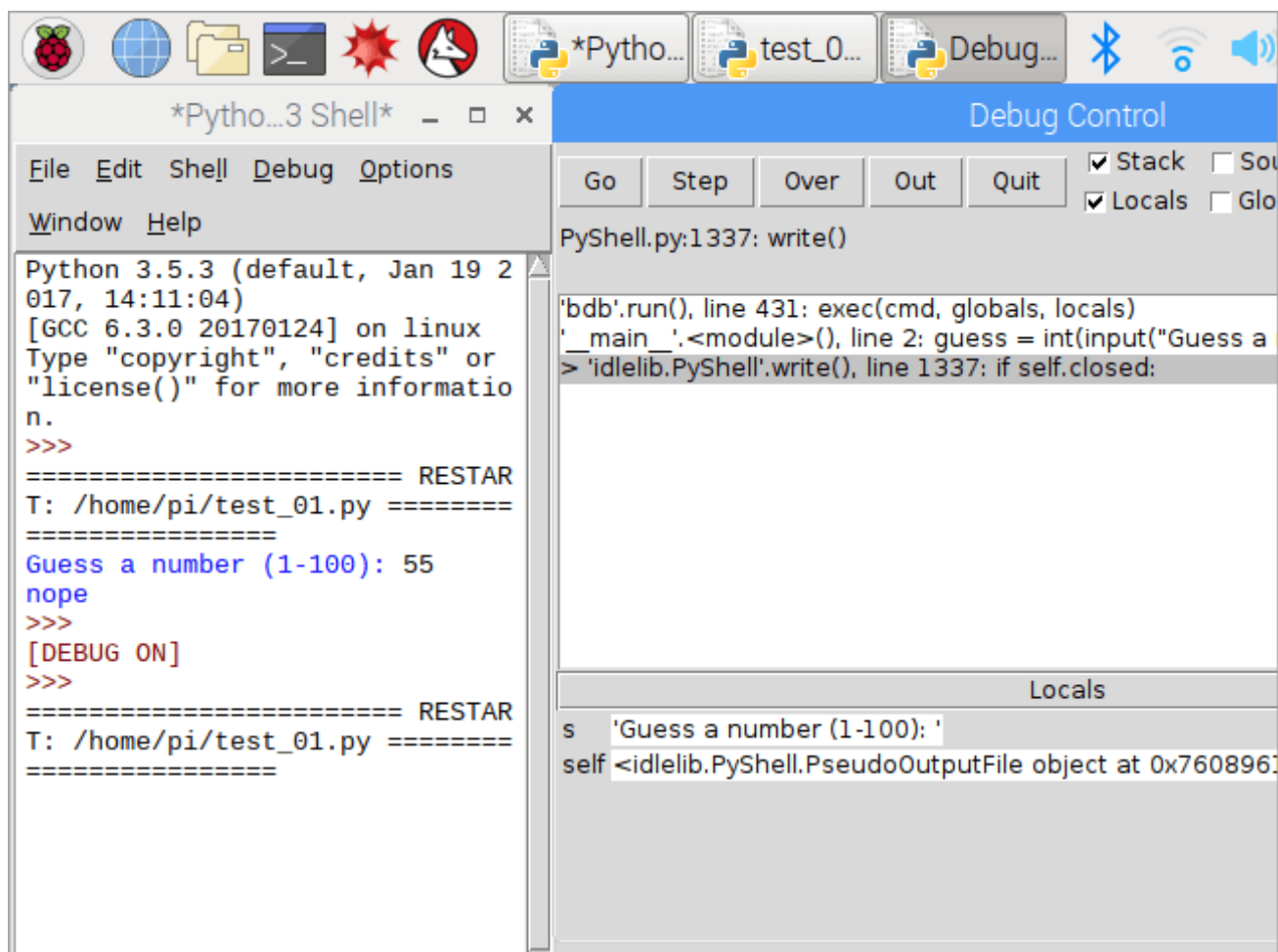
### ➤ PYTHON 3, IDLE

Когато се инсталира Python 3, IDLE също се инсталира по подразбиране. Това улеснява започването на работа с Python. IDLE е популярна интегрирана среда за разработка. Разработен е изцяло в Python с използването на Tkinter GUI. Това е една от най-добрите IDE за Python. Използва се главно от начинаещи разработчици, които искат да практикуват разработването на код на Python.

Основните му характеристики включват „shell window“ (интерактивен интерпретатор), автоматично довършване, подчертаване на синтаксиса. Разполага с текстов редактор с много прозорци, съвети, интелигентен отстъп, отмяна на последна редакция (undo), дебъгер с непрекъснати точки на прекъсване (breakpoints), глобален изглед и локални пространства. Той също така поддържа диалогови прозорци, браузъри и редактируеми конфигурации.

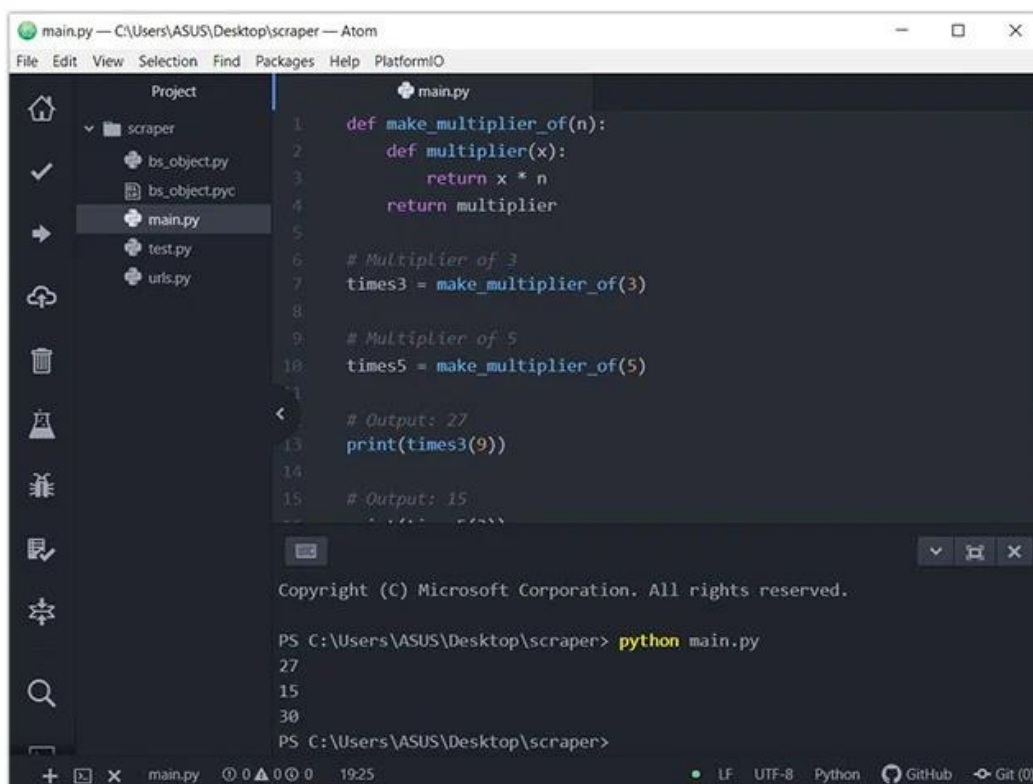
Не е оптимизиран за по-големи проекти. За това се използват програми като PyCharm, Visual Studio Code, Vim и др.





### ➤ ATOM (code editor)

Atom е редактор на код с отворен код, разработен от Github, който може да се използва за разработване на програми на Python. Atom дава възможност за персонализиране. Могат да се инсталират пакети според нуждите на потребителя. Някои от често използваните пакети в Atom за разработка на програми на Python са „autocomplete-python“, „linter-flake8“, „python-debugger“ и др.



```
main.py — C:\Users\ASUS\Desktop\scraper — Atom
File Edit View Selection Find Packages Help PlatformIO

Project
  scraper
    bs_object.py
    bs_object.pyc
    main.py
    test.py
    uris.py

1 def make_multiplier_of(n):
2     def multiplier(x):
3         return x * n
4     return multiplier
5
6 # Multiplier of 3
7 times3 = make_multiplier_of(3)
8
9 # Multiplier of 5
10 times5 = make_multiplier_of(5)
11
12 # Output: 27
13 print(times3(9))
14
15 # Output: 15
16 print(times5(3))

Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\ASUS\Desktop\scraper> python main.py
27
15
30
PS C:\Users\ASUS\Desktop\scraper>
```

## ➤ GitHub

### Какво е GitHub?

GitHub е уеб-базиран интерфейс, който използва Git (софтуер за контрол на версиите с отворен код, който позволява на множество хора да правят отделни промени в уеб страници едновременно). Позволява сътрудничество в реално време и насърчава екипите да работят заедно, за да изграждат и редактират съдържанието на сайта си.

### Как GitHub може да бъде полезен?

GitHub позволява на множество разработчици да работят по един проект по едно и също време, намалява риска от дублираща се или противоречива информация и може да помогне за намаляване на времето за разработване. С GitHub разработчиците могат едновременно да създават код, да проследяват промените и да въвеждат иновативни решения на проблеми, които могат да възникнат по време на процеса на разработка на сайта. Тези, които не участват в разработване на проекти, а влизат в сайта като гости могат също да го използват за създаване, редактиране и актуализиране на съдържанието на уебсайтове.

## Основни термини и понятия в GitHub?

- Хранилище (repo) - папка, в която се съхраняват всички файлове и техните истории на версиите.
- Клон (Branch) - работно пространство, в което можете да правите промени, които няма да повлияят на активния сайт;
- Markdown (.md) - начин за писане в Github, който преобразува обикновен текст в GitHub код. Сайтове като Atom и Sublime Text са примери за безплатни ресурси за разработчици, използващи Markdown;
- Commit Changes - запазен запис на промяна, направена във файл;
- Pull Request (PR) - начинът да поискате промените, направени в един клон, да бъдат обединени в друг клон, който също позволява на множество потребители да виждат, обсъждат и преглеждат извършената работа;
- Обединяване (Merge)- след одобряване на заявка за изтегляне, ангажиментът ще бъде изтеглен (или обединен) от един клон в друг и след това ще бъде поставен на сайта на живо;
- Проблеми (Issues)- как се проследява работата при използване на git. Issues позволяват на потребителите да отчитат нови задачи и корекции на съдържанието, както и да проследяват напредъка на даден проект от началото до края.
- Federalist - платформа, която безопасно и бързо извежда уебсайт от хранилище на GitHub и позволява на потребителите да визуализират предложените и публикувани промени.

## Задачи :

1. Да се инсталира Python 3/ IDLE- <https://www.python.org/downloads/> .
2. Да се инсталира Atom (code editor)- <https://atom.io/> .
3. Студентите да се запознаят с потребителския интерфейс и функциите на двете програми.
4. Всеки студент да си направи регистрация в сайта на GitHub и да се запознае с основните функционалности- <https://github.com/>