EE 551 A
Programming in Python
Spring 2021
Lecture 1: Introduction
02/5/2021

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Engineering Programming: Python EE 551A
Electrical and Computer Engineering

About me

- Adjunct Assistance Prof at ECE department since 2018
- Adjunct Assistance Prof at TCNJ, FDU, CUNY – other universities
- Distinguish member of technical staff – Verizon (2015 - Present)
- Principal member of technical staff – Alcatel-Lucent (2009 - 2015)
- PhD at Ad hoc mobile networks from City University of New York - 2009
- BS and MS in Electrical and Computer Engineering from Alexandria University - Egypt

My Interests

- Data analytics -5G RAN and Core
- Operating systems - cloud computing
- Wireless communication 4G
- Network architecture for messaging and voice
- ..

About the course

- Will use canvas for announcements, lectures files, and grades
- Will github for assignments- repository

- Tools that students will need to install:
  - github, Pycharm, Jubiter

Survey

- Distribute survey at canvas
- Students submit survey at canvas by date Feb 10

Course General Information

• Lecture time:	Friday at 6:30pm – 9:00pm
• Classroom:	Main campus room BC 210 # and virtual via zoom https://stevens.zoom.us/j/96810001862
• Instructor:	Yousef Abdelmalek
• Teaching Assistant:	Hatim Alhazmi halhazm1@stevens.edu
• Contact info:	yabdelma@stevens.edu 908-239-1378
• Office hours:	Friday 5:30pm-6:30pm
• Course listed with:	CPE 551A

Course Description

- This course presents tool, techniques, algorithms, and programming techniques using the Python programming language for data intensive applications and decision making.
- The course formally introduces techniques to: (i) gather,(ii) store, and (iii) process large volumes of data to make informed decisions.
- Such techniques find applicability in many engineering application areas, including communications systems, embedded systems, smart grids, robotics, Internet, and enterprise networks, or any network where information flows and alters decision making.

Student Learning Outcomes

After successful completion of this course, students will be able to:

- learn how to design and program python applications
- learn how to use lists, tuples, and dictionaries in python programs
- learn how to write loops and decision statements in python
- learn how to read and write files in python.
- extract and analyze data in python
- learn how to use indexing and slicing to access data in python programs.
- prepare for their future career in Technology related fields

Course Methodology

- I believe that students learn best by doing, so writing python programs, and classes will include dynamic illustrations of the concepts. Students are expected to bring their laptops and run the programs in their own laptops.
- It is also helpful to study incomplete and incorrect solutions
- There will be weekly coding assignments. The students will return assignments via github which will be tested using continuous integration tool Travis-CI.
- There will be pop quizzes at the end of some classes

Course Materials

- All other materials (code) and slides will be uploaded to course canvas/github website
- Think Python, by Allen B. Downey, second edition, O'Reilly, Sebastopol, California.
- How to think like a Computer Scientist by Downey, Elkner, Meyers
- Canvas will be used for sharing assignment links and grades

Course Requirements

- **Attendance:** Attendance is crucial for an effective learning but will not be graded. However, there will be pop quizzes.
- **Participation:** During the lectures, randomly selected students might be asked questions about the concepts taught in the lectures, but these will not be graded.
- **Homework:** The assignments will be submitted via github and graded and tested in the cloud.
- **Quizzes:** Pop quizzes will be given at the beginning/end of the lectures.
- **Project(s):** Students are welcome to work on a project they desire.
- **Exams** There will be a mid-term and final hand-written exam.

Grading Policy

Grading Procedures

• Homework & attendance (40 %)	400 points
• Project/Presentation (20 %)	200 points
• Mid-term Exam (20 %)	200 points
• Final Exam (20 %)	200 points

Late Policy:

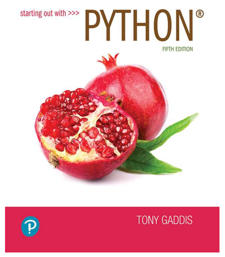
- Assignments submitted after the deadline will be 10% deduced each day after the due date.
- There will be no make-ups for the quizzes.

Code of Academic Integrity

- *All Stevens graduate students promise to be fully truthful and avoid dishonesty, fraud, misrepresentation, and deceit of any type in relation to their academic work. A student's submission of work for academic credit indicates that the work is the student's own. All outside assistance must be acknowledged. Any student who violates this code or who knowingly assists another student in violating this code shall be subject to discipline.*
- *All graduate students are bound to the Graduate Student Code of Academic Integrity by enrollment in graduate coursework at Stevens. It is the responsibility of each graduate student to understand and adhere to the Graduate Student Code of Academic Integrity. More information including types of violations, the process for handling perceived violations, and types of sanctions can be found at www.stevens.edu/provost/graduate-academics.*

Compilers and Interpreters (1 of 3)

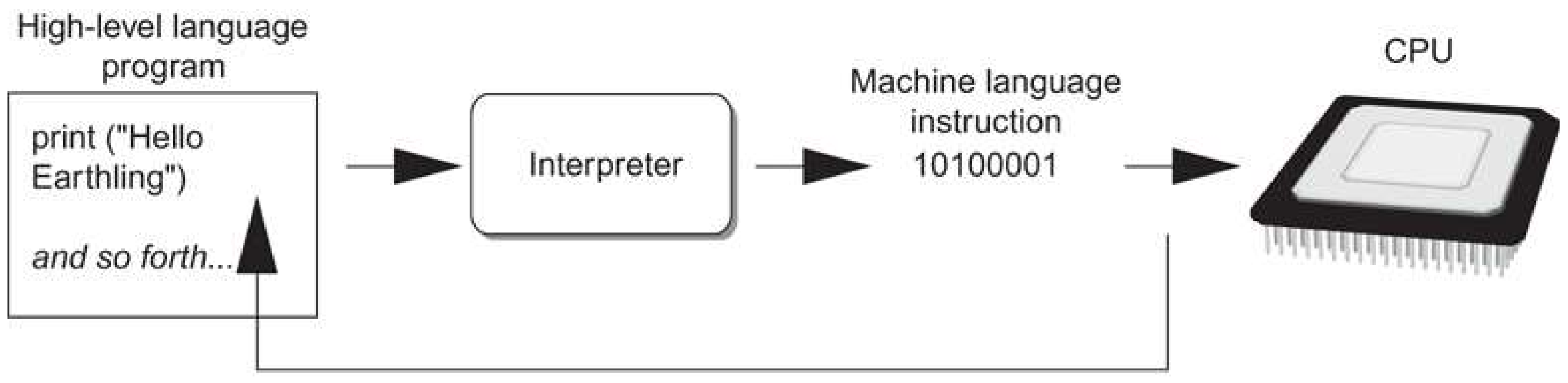
- Programs written in high-level languages must be translated into machine language to be executed
- Compiler: translates high-level language program into separate machine language program
  - Machine language program can be executed at any time



Compilers and Interpreters (2 of 3)

- Interpreter: translates and executes instructions in high-level language program
  - Used by Python language
  - Interprets one instruction at a time
  - No separate machine language program
- Source code: statements written by programmer
  - Syntax error: prevents code from being translated

Compilers and Interpreters (3 of 3)


High-level language program
print ("Hello Earthling")
and so forth...
Interpreter
Machine language instruction
10100001
CPU

The interpreter translates each high-level instruction to its equivalent machine language instructions then immediately executes them.

This process is repeated for each high-level instruction.

Using Python

- Python must be installed and configured prior to use
  - One of the items installed is the Python interpreter
- Python interpreter can be used in two modes:
  - Interactive mode: enter statements on keyboard
  - Script mode: save statements in Python script

Interactive Mode

IDLE

- When you start Python in interactive mode, you will see a prompt
    - Indicates the interpreter is waiting for a Python statement to be typed
    - Prompt reappears after previous statement is executed
    - Error message displayed If you incorrectly type a statement
- Good way to learn new parts of Python

Writing Python Programs and Running Them in Script Mode

- Statements entered in interactive mode are not saved as a program
- To have a program use script mode
  - Save a set of Python statements in a file
  - The filename should have the `.py` extension
  - To run the file, or script, type

    `python` *`filename`*

    at the operating system command line

IDlE

- IDLE (Integrated Development Program): single program that provides tools to write, execute and test a program
  - Automatically installed when Python language is installed
  - Runs in interactive mode
  - Has built-in text editor with features designed to help write Python programs

Why Python (1)

Software Quality

- Simple and readable syntax; hence reusable and maintainable
- Python seems to `fit your brain` -features of the language interact in consistent and limited ways and follow naturally from a small set of core concepts.
- Easier to learn, understand and remember
- Python adopts a somewhat minimalist approach. Explicit is better than implicit, and simple is better than complex.

Why Python (2)

Extensive Support Libraries

- It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python

Why Python (3)

Integration Feature

- Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other markup languages as it can run on all modern operating systems through same byte code.

Limitations or Disadvantages of Python (1)

Weak in Mobile Computing

- Python has made its presence on many desktop and server platforms, but it is seen as a weak language for mobile computing. This is the reason very few mobile applications are built in it.

Limitations or Disadvantages of Python (2)

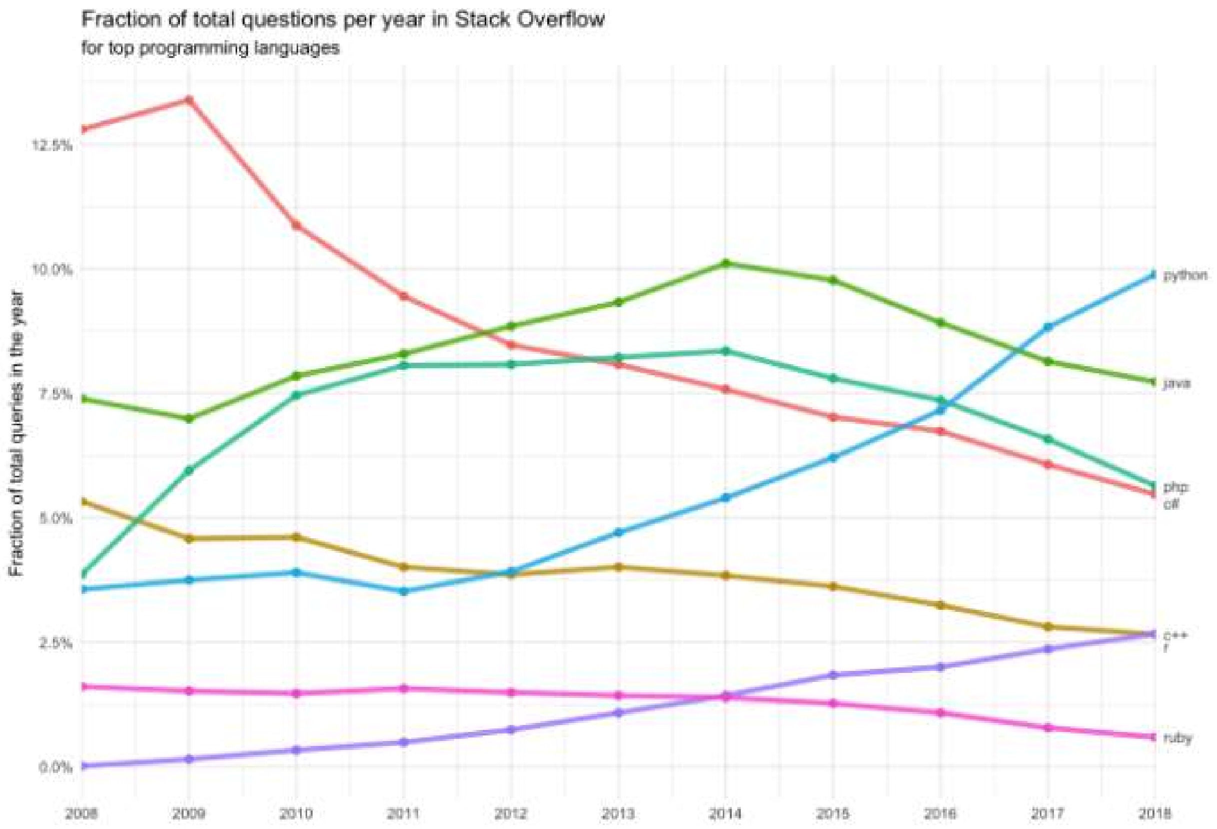
Gets Slow in Speed

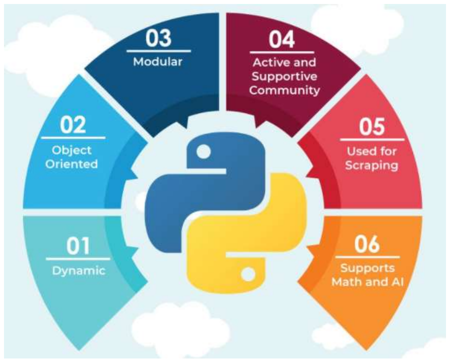
- Python executes with the help of an interpreter instead of the compiler, which causes it to slow down because compilation and execution help it to work normally. On the other hand, it can be seen that it is fast for many web applications too.

Is Python a “Scripting language”?

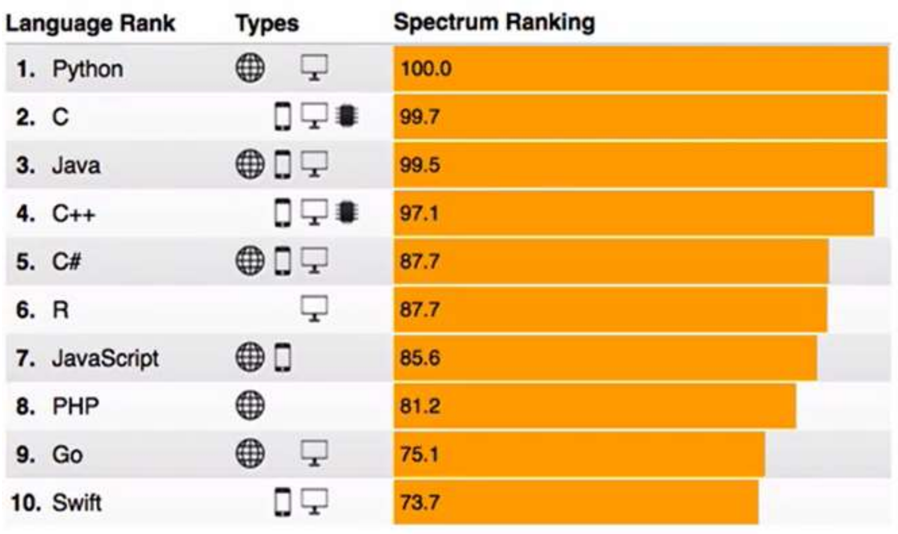
- Python is a general-purpose programming language that is often applied in scripting roles.
- The term "scripting" often implies languages that don't require an explicit compilation step.
- Python is often used without a compilation step but can be compiled, too.
- So, in general, the term “scripting” is probably best used to describe the rapid and flexible mode of development that Python supports.

Why Python


Fraction of total questions per year in Stack Overflow
for top programming languages
Fraction of total queries in the year
0.0%
2.5%
5.0%
7.5%
10.0%
12.5%
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
python
java
php
c#
c++
r
ruby

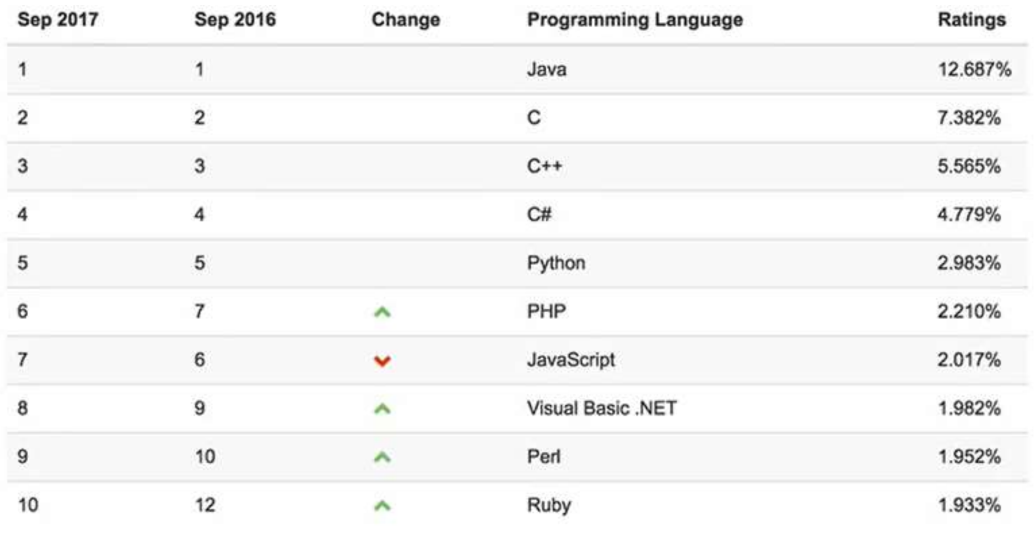

01
Dynamic
02
Object Oriented
03
Modular
04
Active and Supportive Community
05
Used for Scraping
06
Supports Math and AI

Growing in Popularity



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥	100.0
2. C	📱 🖥 ▦	99.7
3. Java	🌐 📱 🖥	99.5
4. C++	📱 🖥 ▦	97.1
5. C#	🌐 📱 🖥	87.7
6. R	🖥	87.7
7. JavaScript	🌐 📱	85.6
8. PHP	🌐	81.2
9. Go	🌐 🖥	75.1
10. Swift	📱 🖥	73.7

IEEE Spectrum (http://bit.ly/2wWgUaB)



Sep 2017	Sep 2016	Change	Programming Language	Ratings
1	1		Java	12.687%
2	2		C	7.382%
3	3		C++	5.565%
4	4		C#	4.779%
5	5		Python	2.983%
6	7	▲	PHP	2.210%
7	6	▼	JavaScript	2.017%
8	9	▲	Visual Basic .NET	1.982%
9	10	▲	Perl	1.952%
10	12	▲	Ruby	1.933%

TIOBE (http://bit.ly/2mPBq5f)

What can you do using Python

- Data science/analytics
- Web development
- Scripting
- Database programming
- Machine learning
- Gaming
- Robotics
- And more

Getting started!

Command Line:

- https://philchodrow.github.io/cos_2017/1_terminal_and_git/Introduction%20to%20terminal.pdf

Windows users:

Linux users:

https://www.youtube.com/watch?v=uTeH7vm8JZU

Getting started!

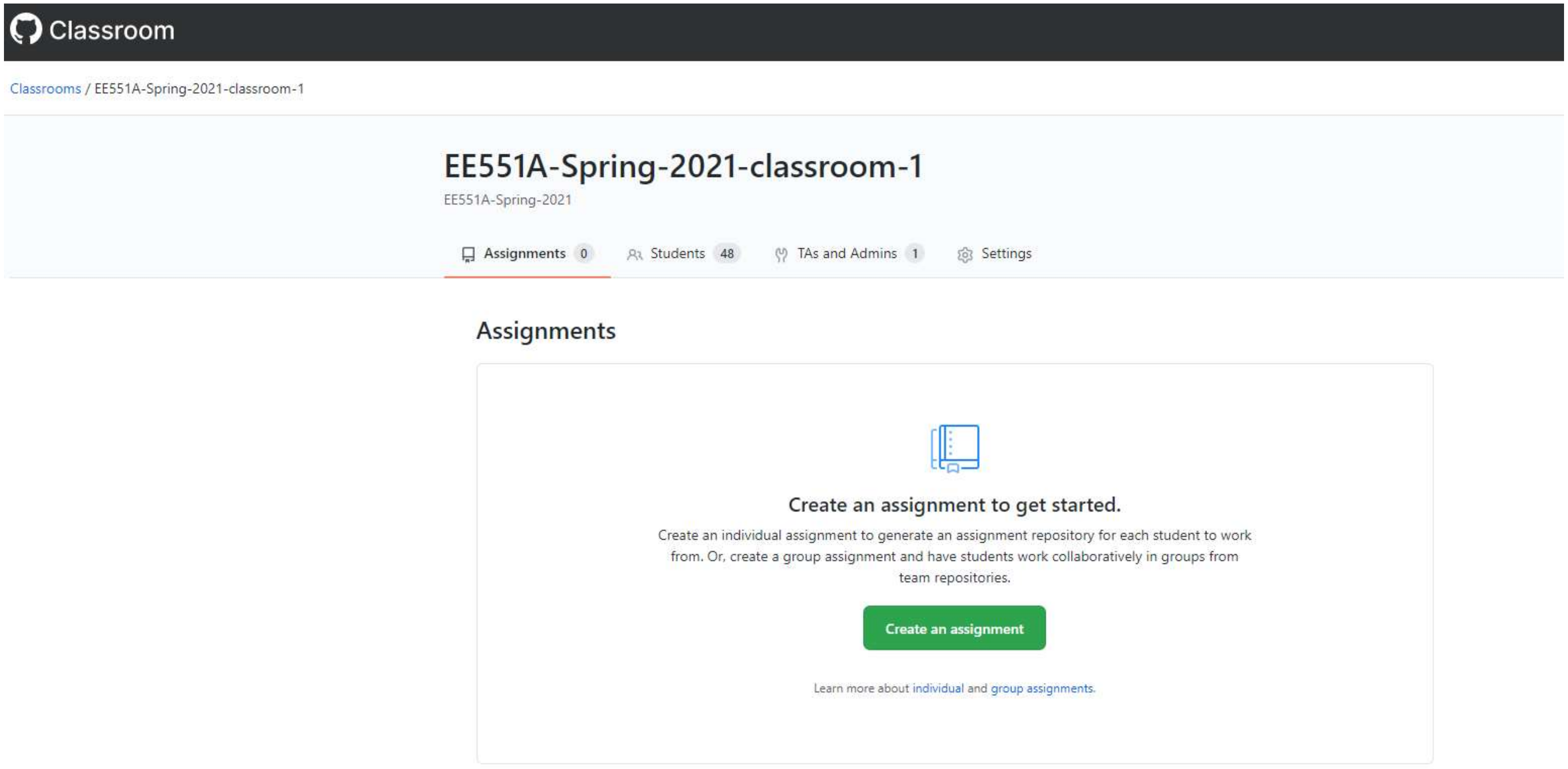
Git –version control

Github account creation

If you are not familiar with git and Github

Watch Git Github essential training course by Kevin Skoglundat Lynda

Classroom.github.com


Classroom

Classrooms / EE551A-Spring-2021-classroom-1

EE551A-Spring-2021-classroom-1

EE551A-Spring-2021

Assignments

Create an assignment to get started.

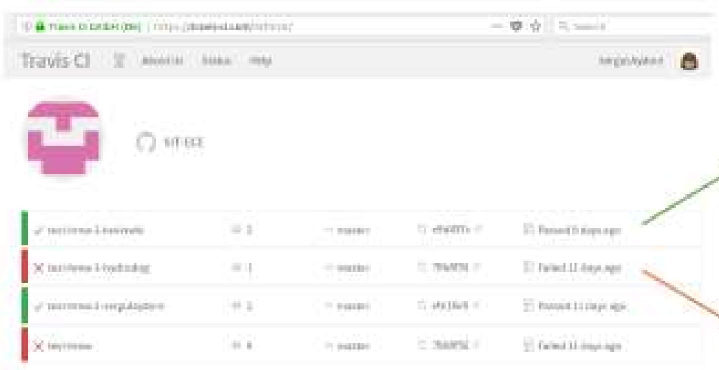
Create an individual assignment to generate an assignment repository for each student to work from. Or, create a group assignment and have students work collaboratively in groups from team repositories.

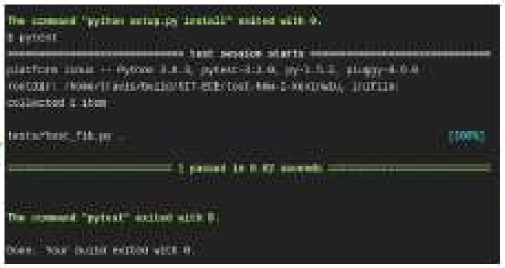
Learn more about individual and group assignments.

A private repository will be created for each student

Travis CI

- Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub. (wikipedia)





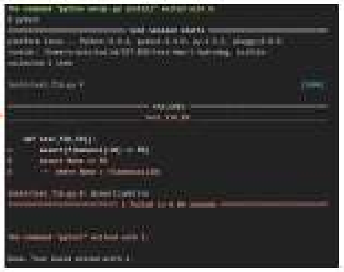
```
The command "python setup.py install" exited with 0.
$ pytest
============================= test session starts ==============================
platform linux -- Python 3.6.3, pytest-3.3.0, py-1.5.2, pluggy-0.6.0
rootdir: /home/travis/build/SIT-ECE/[illegible], inifile:
collected 1 item

tests/test_fib.py .                                                      [100%]

=========================== 1 passed in 0.02 seconds ===========================

The command "pytest" exited with 0.

Done. Your build exited with 0.
```



```
The command "python setup.py install" exited with 0.
$ pytest
============================= test session starts ==============================
[illegible]
collected 1 item

tests/test_fib.py F                                                      [100%]

=================================== FAILURES ===================================
                                    test_fib

    def test_fib_50():
>       assert[illegible]
E       [illegible]

tests/test_fib.py:[illegible]: AssertionError
=========================== 1 failed in 0.[illegible] seconds ===========================

The command "pytest" exited with 1.

Done. Your build exited with 1.
```

When you "push" your code, it will automatically run the test I wrote for you. And I can see whose code passed the tests or not

Install Python

- If you don't have it, install the recent version from https://www.python.org/

  An interpreter is a kind of program that executes other programs.

  To enable that, you must install a Python interpreter to your computer.

  Make sure to have Python 3.4 or later installed in your system by typing python –V in your shell.

A private repository will be created for each student

- pip is a tool for installing Python packages. It comes with Python 3.x installation but make sure you have upgraded it https://pip.pypa.io/en/stable/installing
- A Virtual Environment enables us to keep the dependencies required by different projects in separate places, by creating virtual Python environments.
- virtualenvwrapper is a set of extensions to virtualenv tool. The extensions include wrappers for creating and deleting virtual environments and otherwise managing our development workflow, making it easier to work on more than one project at a time without introducing conflicts in their dependencies.
- Setup an environment with virtual environment wrapper based on your system: http://virtualenvwrapper.readthedocs.io/en/latest/

A private repository will be created for each student

- Interactive environments are useful for fast prototyping but a better text editor is required for more professional development.
- You can write your Python code in any text editor.
- There are special editors designed for Python software development: IDE (Integrated Development Environment)
- One of the most popular IDEs for Python is PyCharm
- Download PyCharm

https://www.jetbrains.com/pycharm-edu/download

PyCharm

- Configure PyCharm
- Change the appearances in Settings to Darcula
- Change the interpreter to the location where your virtual Environment is located

Demo

- Task1: Install Python
  - Few notes:
    - Idle is chosen
    - Check the environment variable
    - Make sure that installed directory is unique
  - Check cmd has python version
  - Check IDLE is installed
- Task2: Install Pycharm
  - Few notes
    - Make sure that Dracule is chosen
    - Make sure that you choose pip to be installed
    - Make sure that you choose interpreter
- Task3: get familiar with github
  - Create github account
  - Watch the tutorial about github
    - Learn how to download source
    - Learn how to commit changes to code
    - Learn basics of github commends

Summary

- This lecture covered:
  - Course outlines and objectives
  - Why Python
  - What software are needed to start using Python
  - Advantages of Python
  - Limitations of Python
  - Installing Python and the Python interpreter modes