Principles of Computer Science II

Algorithms for BioInformatics

Ioannis Chatzigiannakis

Sapienza University of Rome

Lecture 2

Development Tools

Programming Tool

A programming tool or software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications.

- Source Code Editor
- Debugger or Profiler
- Bug Tracking System
- Documentation Generators
- Revision Control
- Performance Analysis
- Collaborative Programming
- Cloud-based IDEs

Integrated Development Environment (IDE)

A programming tool or software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications. The IDE is meant to make programming a more productive process.

- Organize project files
- Searching
- Source Code Editor
- Debugger
- Tasks & Annotations related to code
- Documentation Generators
- Revision Control
- Code Analysis

Jupyter Notebook

- Interactively developing and presenting data science projects.
- A single document integrates: code and its output, visualizations, narrative text, mathematical equations, and other rich media.

Installation & Execution

Installation:

- For Windows - make sure you first install Anaconda, then use pip.
- For Mac / Linux use directly pip:

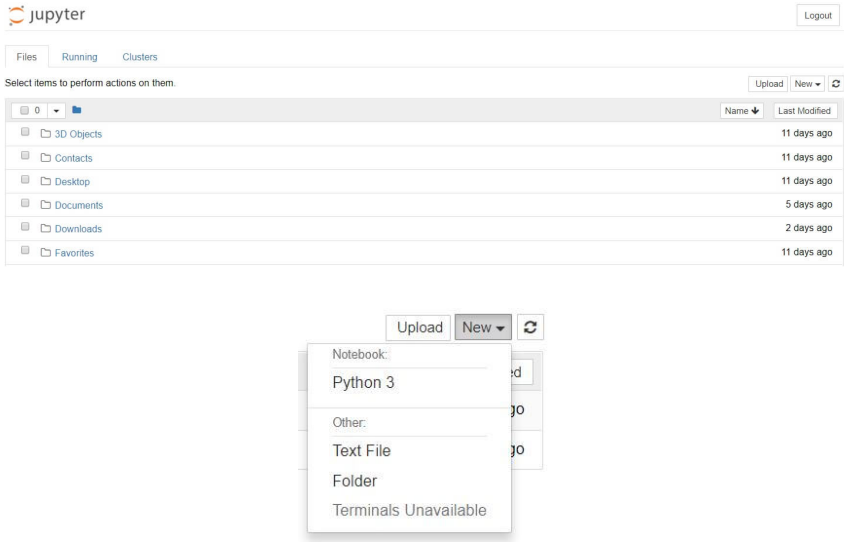
```
pip3 install jupyter
```

Execution:

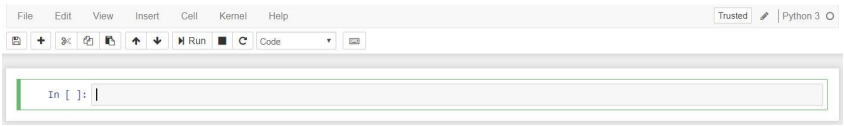
- For Windows - via Anaconda.
- For Mac / Linux from the command line:

```
jupyter notebook
```

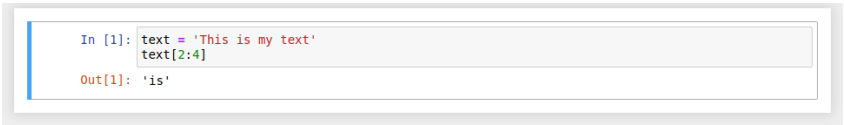
- The jupyter interface is avalable at `http://localhost:8888/tree`



The Notebook Interface



- A Cell can be either Code or Markdown
- Use the ▶| Run button or `CTRL+ENTER` to execute the Code or present the Markdown.



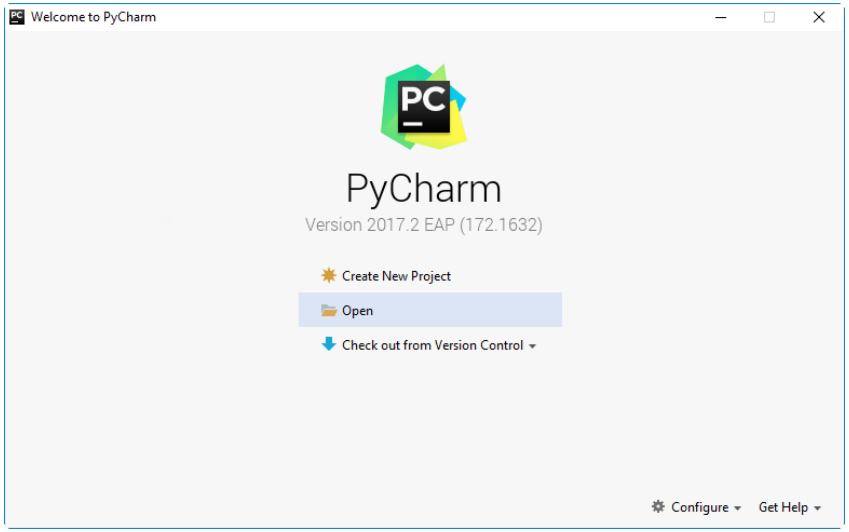
- Check out that In [ ]: has changed to In [1]:
- When python is processing the code we get In [*]:

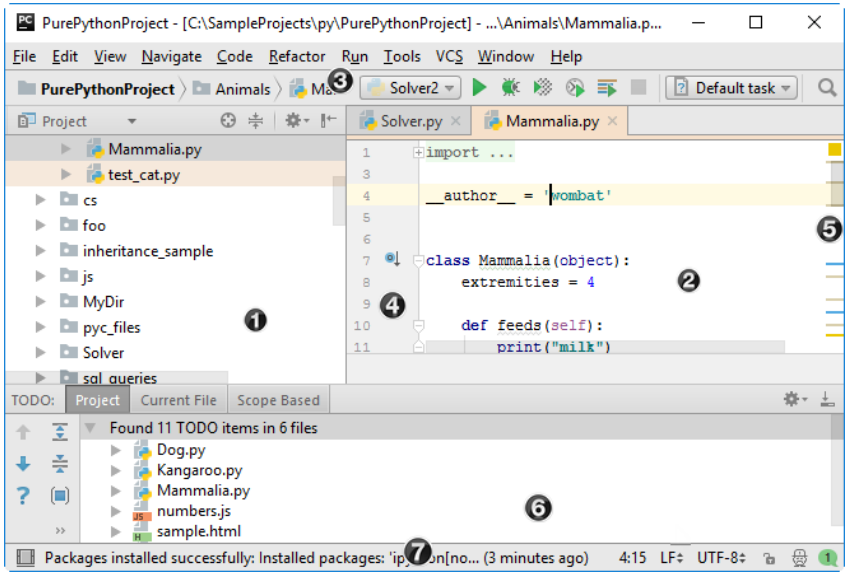
Navigating the Notebook with the Keyboard

- There is always a Cell Active.
- You can **Stop Editing** by using the ESC key.
- You can **Start Editing** by using the ENTER key.
- When **NOT** in Editing mode:
  - We can go up/down the cells using the Up and Down keys.
  - To change the Cell type to Markdown use the M key.
  - To change the Cell type to Code use the Y key.
  - To insert a new Cell above the current Cell use the A key.
  - To insert a new Cell below the current Cell use the B key.
  - To delete a Cell use the D key **twice**.
  - To UNDO a delete command use the Z key.

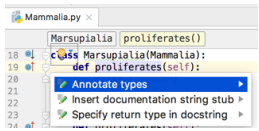
pyCharm: Python IDE for Professional Developers

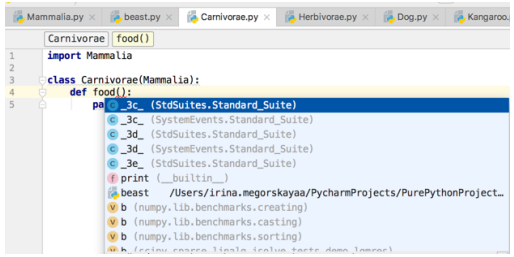
- Keyboard-centric approach
- Smart assistance
- Code quality tools
- Cross technology development
- Navigation and Refactoring
- Database support
- Scientific tools





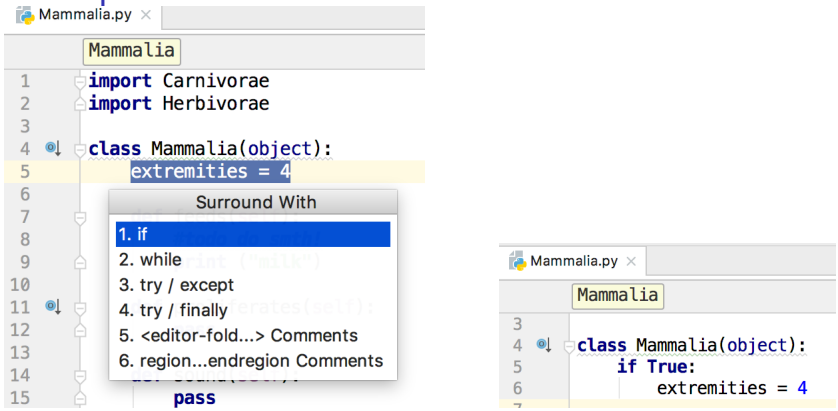
Code with smart assistance





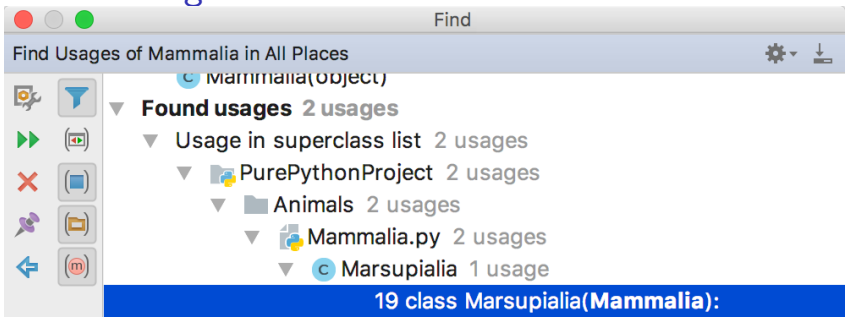
- Intention Action – indicated with a bulb ALT+Enter
  - Suggestions based on the action that you do that intend to save time.
  - Remark that the code needs to be correct for this feature to work.
- Code completion
  - Auto-complete function/variable names.

Live Templates



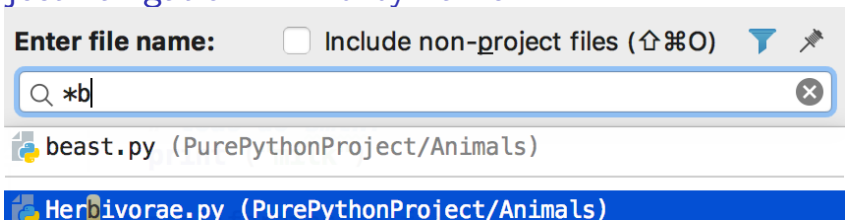
- Live Template ALT+J produce entire code constructs.
- A library of ready-to-use templates.

Search for Usages



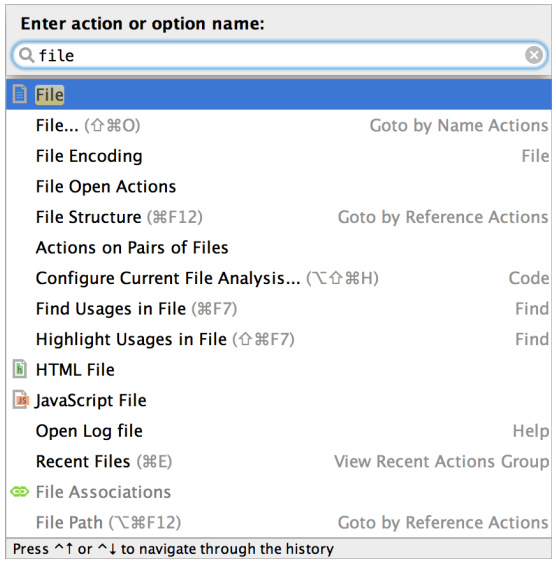
- As the project grows, or when you work with someone else's code.
- To find where a particular symbol is used, ALT+F7
  - All files are searched.

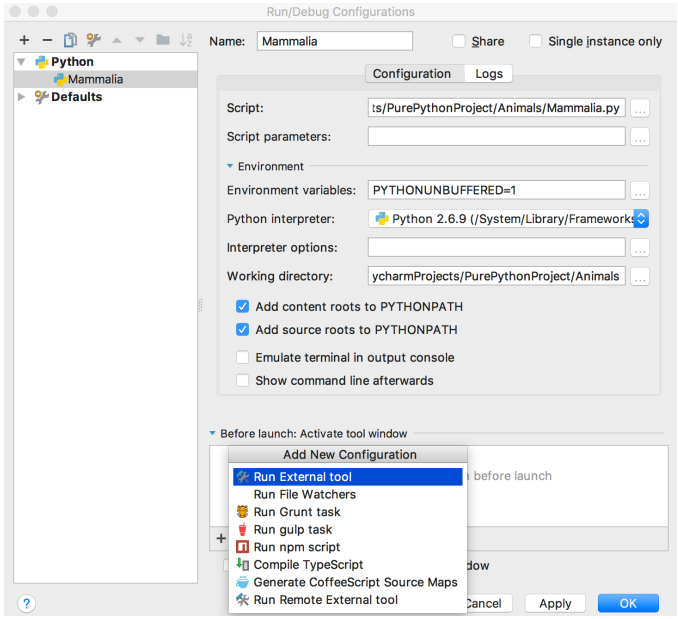
Project navigation – Find by name

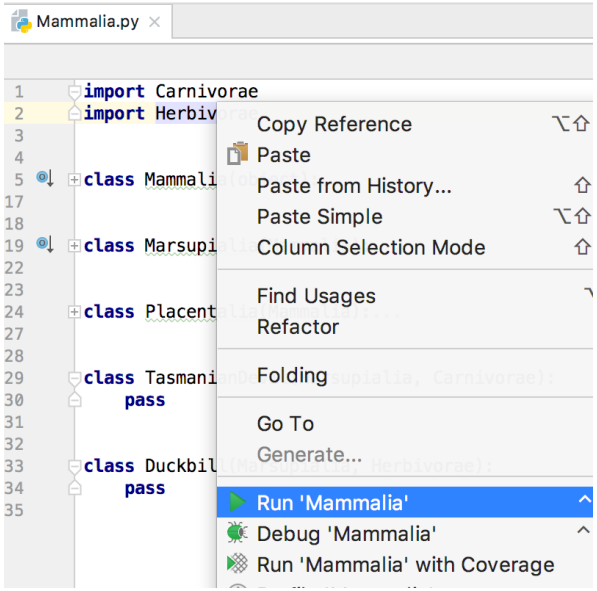


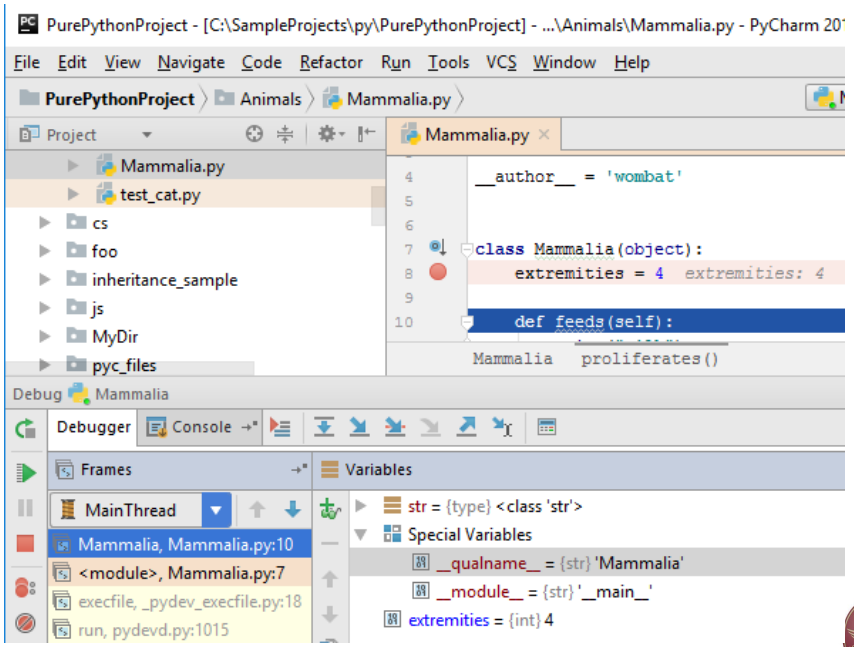
- Search only Classes by name, CTRL+N
- Search only based on filenames, CTRL+Shift+N
- Search Variable, CTRL+Shift+ALT+N
- Search Declaration, CTRL+B
- Search Class/Function, CTRL+U

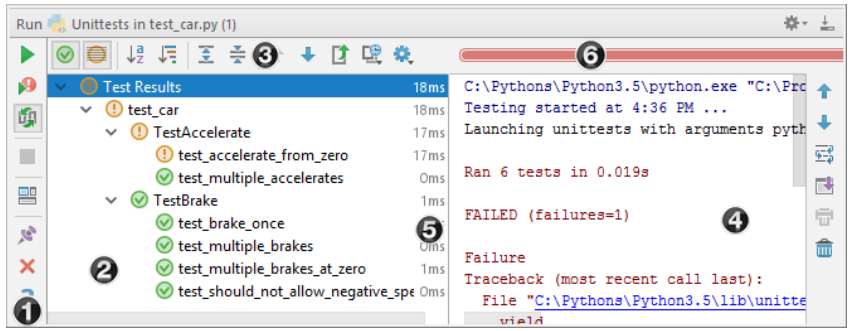
Find Action – CTRL+Shift+A

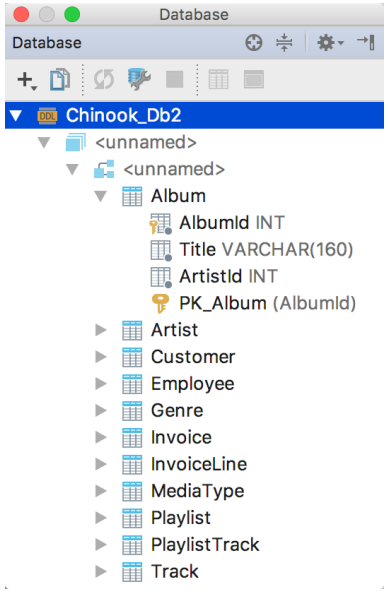



Run/Debug Configurations
Name: Mammalia
Share
Single instance only
Python
Mammalia
Defaults
Configuration
Logs
Script: :s/PurePythonProject/Animals/Mammalia.py
Script parameters:
Environment
Environment variables: PYTHONUNBUFFERED=1
Python interpreter: Python 2.6.9 (/System/Library/Frameworks
Interpreter options:
Working directory: ycharmProjects/PurePythonProject/Animals
Add content roots to PYTHONPATH
Add source roots to PYTHONPATH
Emulate terminal in output console
Show command line afterwards
Before launch: Activate tool window
Add New Configuration
Run External tool
Run File Watchers
Run Grunt task
Run gulp task
Run npm script
Compile TypeScript
Generate CoffeeScript Source Maps
Run Remote External tool
before launch
Cancel
Apply
OK


Mammalia.py
import Carnivorae
import Herbiv
class Mammali
class Marsupi
class Placent
class Tasmani
pass
class Duckbil
pass
Copy Reference
Paste
Paste from History...
Paste Simple
Column Selection Mode
Find Usages
Refactor
Folding
Go To
Generate...
Run 'Mammalia'
Debug 'Mammalia'
Run 'Mammalia' with Coverage


PurePythonProject - [C:\SampleProjects\py\PurePythonProject] - ...\Animals\Mammalia.py - PyCharm 20
File Edit View Navigate Code Refactor Run Tools VCS Window Help
PurePythonProject
Animals
Mammalia.py
Project
Mammalia.py
test_cat.py
cs
foo
inheritance_sample
js
MyDir
pyc_files
__author__ = 'wombat'
class Mammalia(object):
extremities = 4 extremities: 4
def feeds(self):
Mammalia proliferates()
Debug Mammalia
Debugger
Console
Frames
Variables
MainThread
Mammalia, Mammalia.py:10
<module>, Mammalia.py:7
execfile, _pydev_execfile.py:18
run, pydevd.py:1015
str = {type} <class 'str'>
Special Variables
__qualname__ = {str} 'Mammalia'
__module__ = {str} '__main__'
extremities = {int} 4

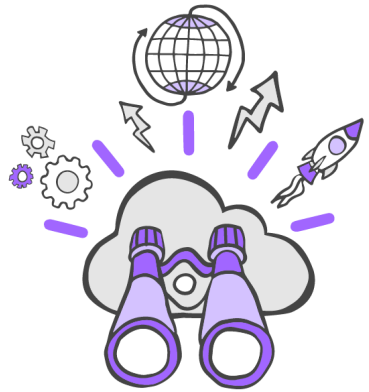

Run
Unittests in test_car.py (1)
Test Results 18ms
test_car 18ms
TestAccelerate 17ms
test_accelerate_from_zero 17ms
test_multiple_accelerates 0ms
TestBrake 1ms
test_brake_once
test_multiple_brakes
test_multiple_brakes_at_zero 1ms
test_should_not_allow_negative_spe 0ms
C:\Pythons\Python3.5\python.exe "C:\Pro
Testing started at 4:36 PM ...
Launching unittests with arguments pyth
Ran 6 tests in 0.019s
FAILED (failures=1)
Failure
Traceback (most recent call last):
File "C:\Pythons\Python3.5\lib\unitte
1
2
3
4
5
6



aws educate

AWS Educate Student Registration

How to create an AWS Educate account and access the AWS portal



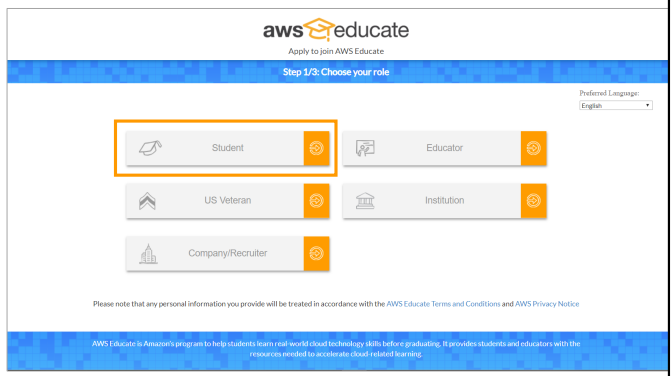
aws educate

Registration Process

aws educate

Step 1

- Visit www.awseducate.com
- Click **Join AWS Educate**
- Select **Student**

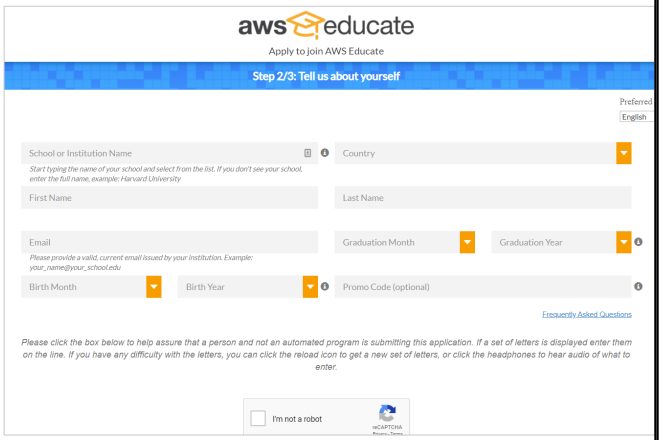


aws educate

Step 2

- Complete all fields
- Click **Next**

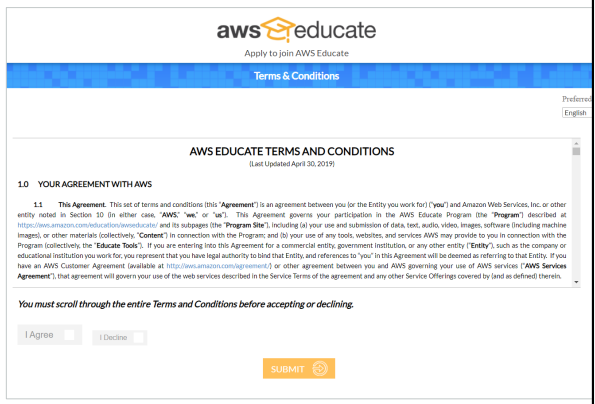
It's important to select your institution to receive increased credits in the AWS Educate Starter Account.



aws educate

Step 3

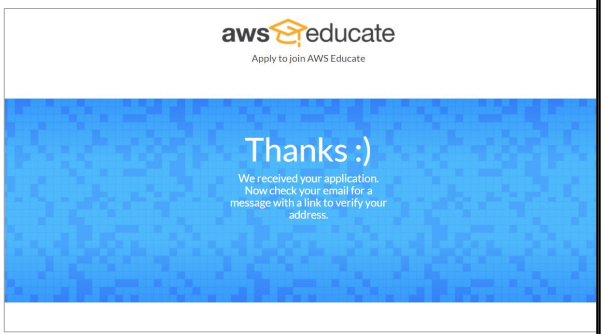
- Review AWS Educate Terms and Conditions
- Select **I Agree**
- Click **Submit**



aws educate

Step 4

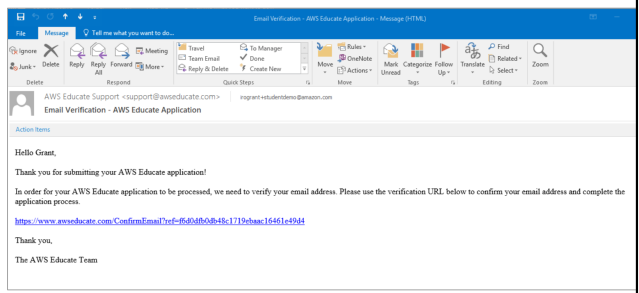
- Confirmation page



aws educate

Step 5

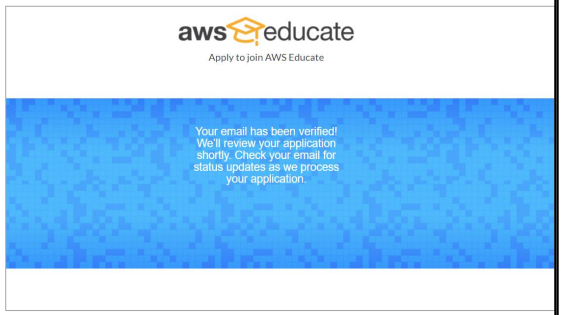
- **Check inbox** for confirmation email
- **Click link** to validate email address



aws educate

Step 6

- Verification page

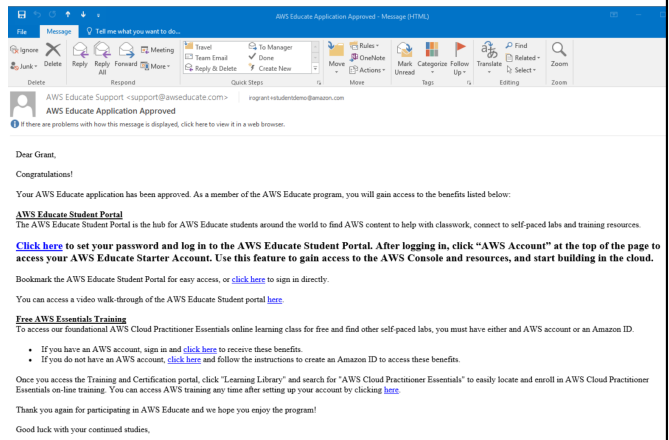


aws educate

Step 7

- **Check inbox** for welcome email
- Select **Click Here** link to activate account and enter portal

Students may receive an email with request for more information. Follow directions.



aws educate

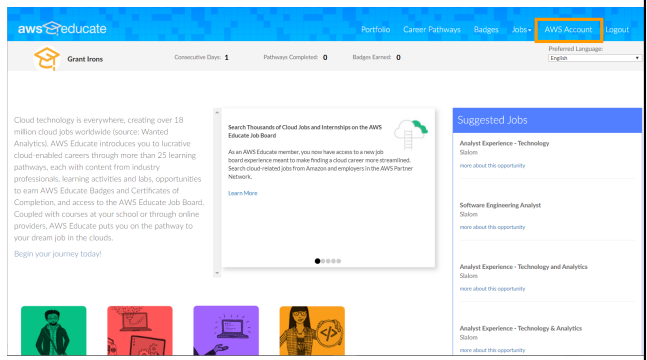
Step 8

- **Set password** at first sign-in

aws educate

Step 9

- From the Student Portal Homepage, click **AWS Account**



aws educate

Step 10

- Select **Create Starter Account** to access the AWS Console

Students at **member institutions** receive **$100** credit annually.

Students at **non-member institutions** receive **$30** credit annually.

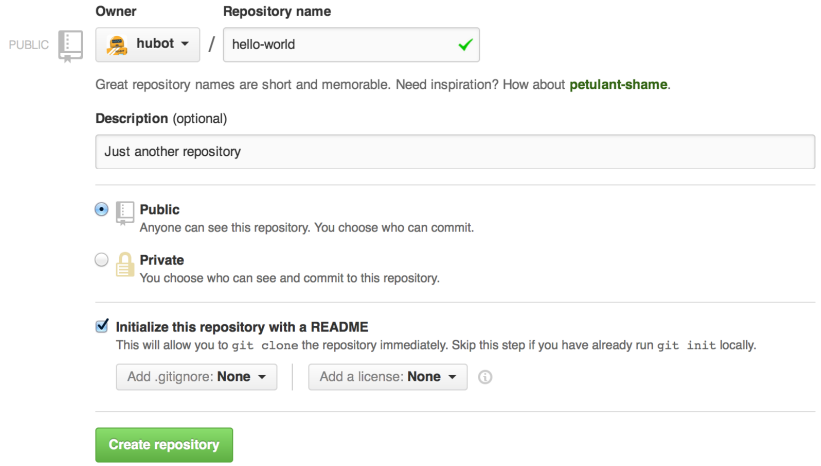
github SOCIAL CODING

- Code Hosting Platform
  - Version Control, Bug Tracking & Todo list, Wiki, Collaboration, ...
- Public + Private Projects
- Cloud-based or Private Storage
- Alternatives:
  - BitBucket, SourceFourge, Team Foundation Server, SVN, CVS

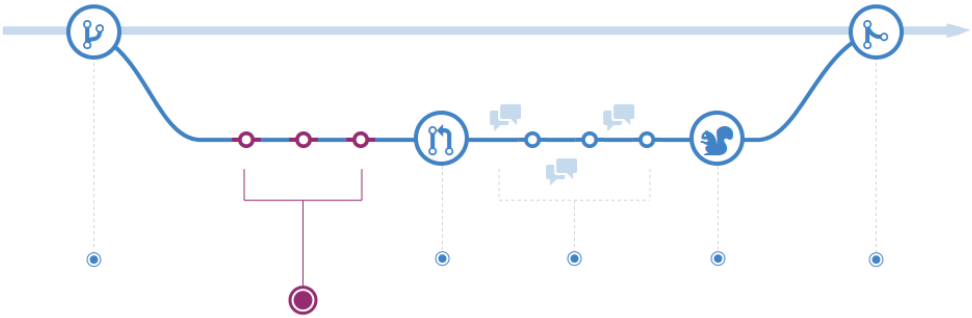
First steps on Github

- Repository-oriented Family of Services
  - Repository: group of files relevant to a specific project.
  - Not necessarily related to coding.
- Each member of the project needs a separate account.
- Repositories are owned by an account.
  - Organizations are also allowed to own repositories.
- Repositories are created via the Website.
- Repositories can be browsed/modified via the Web or via broad range of client applications.

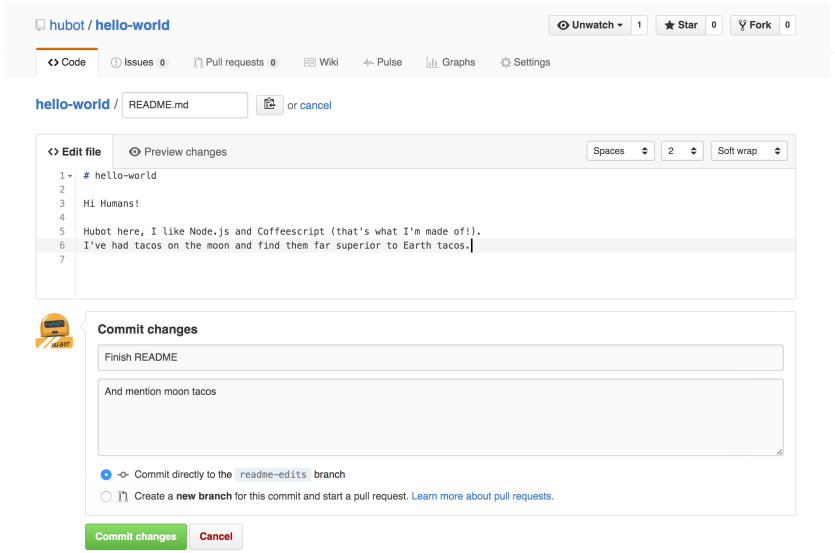
Creating a new Repository



Make and commit changes



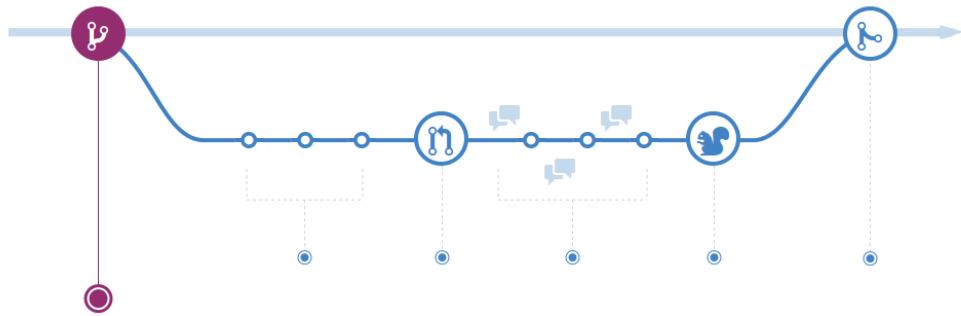
- Whenever you add, edit, delete.
- Keeps track of progress.
- Easy to roll-back to previous states.



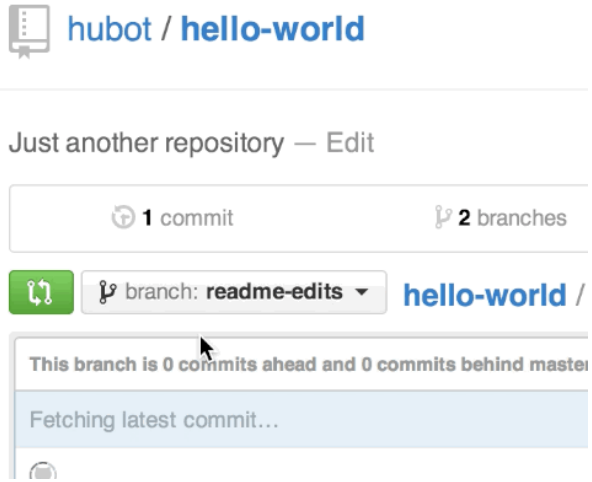
Real power of Github: Branching

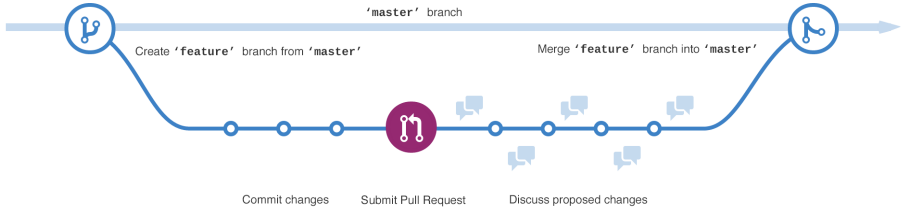
- The most over-stressed functionality.
- Branching: work on different versions of a repository at one time.
- By default each repository has 1 branch: master
- When create a new branch off the master:
  - Make a copy of all contents.
  - Changes on new repository are separated.
  - Can pull changes from master at any point.
  - Can push changes to master at any point.

Branching



- Starting from the MASTER branch.
- We create the FEATURE branch.
- The new branch progresses independently.
- Eventually, it MERGES into MASTER.





- Communicating changes to the other members of the team is done via PULL REQUESTS.
- Pull Requests are the heart of collaboration on GitHub.
- As soon as you make a commit:
  - open a pull request,
  - start a discussion!

Merge Pull Requests

- The final step of bringing changes together.
- Merging 2 brunches.
- After confirming the merge, other branches can be deleted.