

Principles of Computer Science II

Algorithms for Bioinformatics

Ioannis Chatzigiannakis

Sapienza University of Rome

Lecture 3



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:

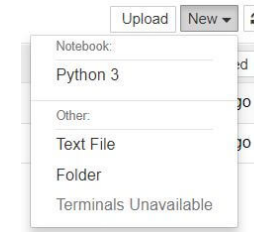
```
pip3 install jupyter
```

Execution:

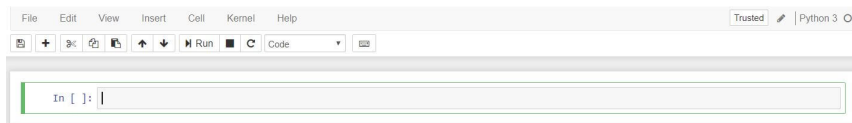
- ▶ For Windows - via Anaconda.
- ▶ For Mac / Linux from the command line:

```
jupyter notebook
```

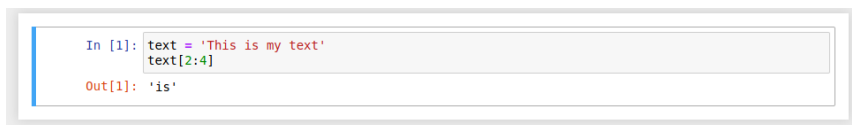
- ▶ The jupyter interface is available at <http://localhost:8888/tree>



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 [*]:**



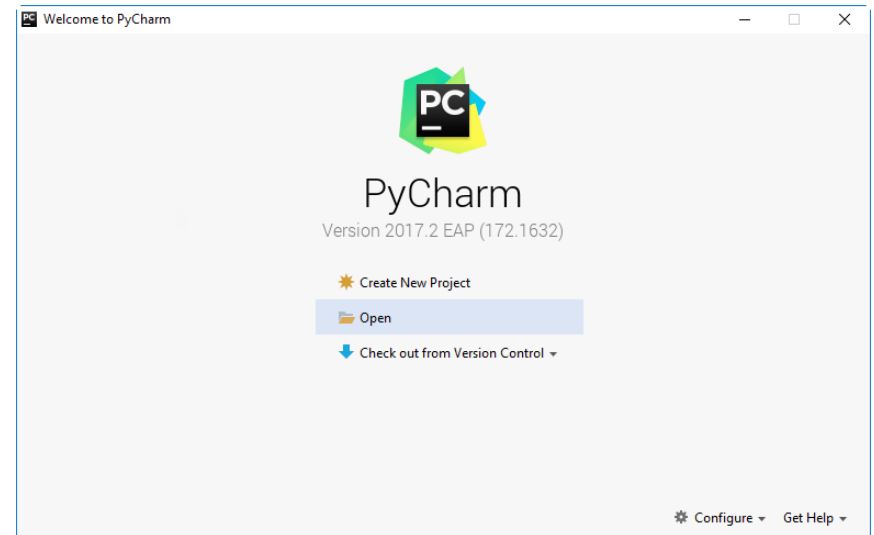
Navigating the Notebook with the Keyboard

- ▶ 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.

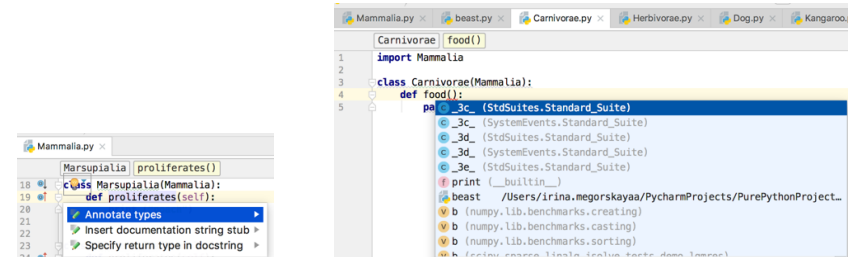
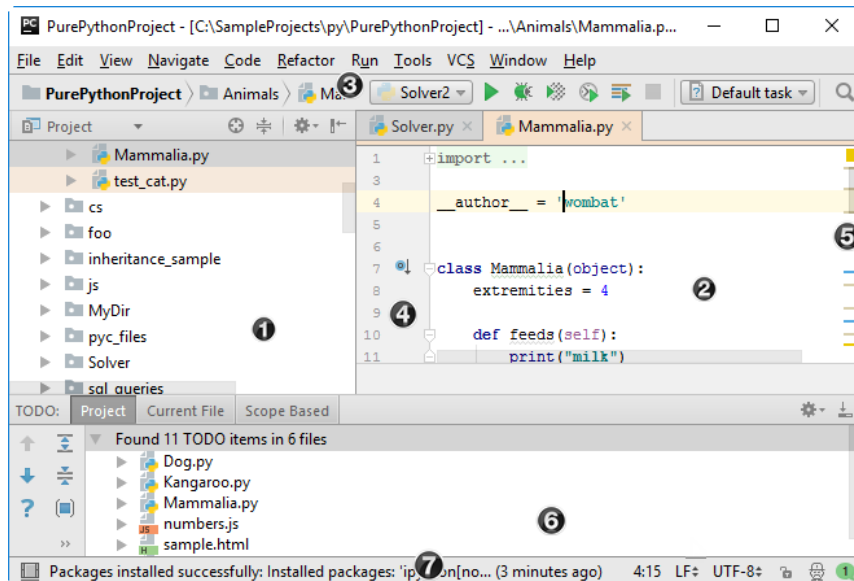


pyCharm: Python IDE for Professional Developers

- ▶ 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

```
Mammalia.py x
Mammalia
1 import Carnivorae
2 import Herbivorae
3
4 class Mammalia(object):
5     extremities = 4
6
7
8
9
10
11
12
13
14
15
```

Surround With

1. if
2. while
3. try / except
4. try / finally
5. <editor-fold...> Comments
6. region...endregion Comments

```
Mammalia.py x
Mammalia
3
4 class Mammalia(object):
5     if True:
6         extremities = 4
7
```

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



Search for Usages

Find Usages of Mammalia in All Places

Mammalia(object)

Found usages 2 usages

- Usage in superclass list 2 usages
 - PurePythonProject 2 usages
 - Animals 2 usages
 - Mammalia.py 2 usages
 - Marsupialia 1 usage

19 class Marsupialia(Mammalia):

- ▶ 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

Enter file name: Include non-project files (⌘O)

Search:

beast.py (PurePythonProject/Animals)

Herbivorae.py (PurePythonProject/Animals)

- ▶ 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

Enter action or option name:

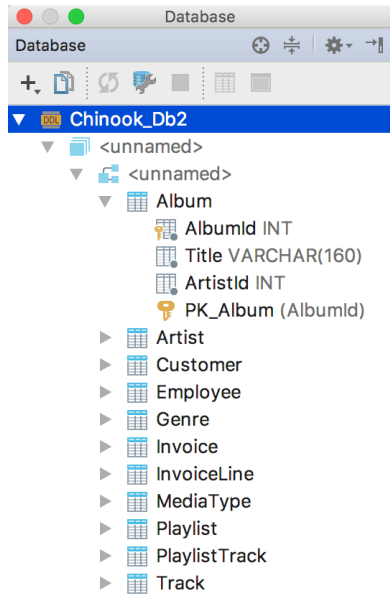
Search:

File

- File... (⌘O) Goto by Name Actions
- File Encoding File
- File Open Actions
- File Structure (⌘F12) Goto by Reference Actions
- Actions on Pairs of Files
- Configure Current File Analysis... (⌘⇧H) Code
- Find Usages in File (⌘F7) Find
- Highlight Usages in File (⌘⇧F7) Find
- HTML File
- JavaScript File
- Open Log file Help
- Recent Files (⌘E) View Recent Actions Group
- File Associations
- File Path (⌘⇧F12) Goto by Reference Actions

Press ⌘↑ or ⌘↓ to navigate through the history





- ▶ Code Hosting Platform
 - ▶ Version Control, Bug Tracking & Todo list, Wiki, **Collaboration**, ...
- ▶ Public + Private Projects
- ▶ Cloud-based or Private Storage
- ▶ Alternatives:
 - ▶ BitBucket, SourceForge, 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

Owner: PUBLIC hubot / Repository name: hello-world ✓

Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

Description (optional): Just another repository

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

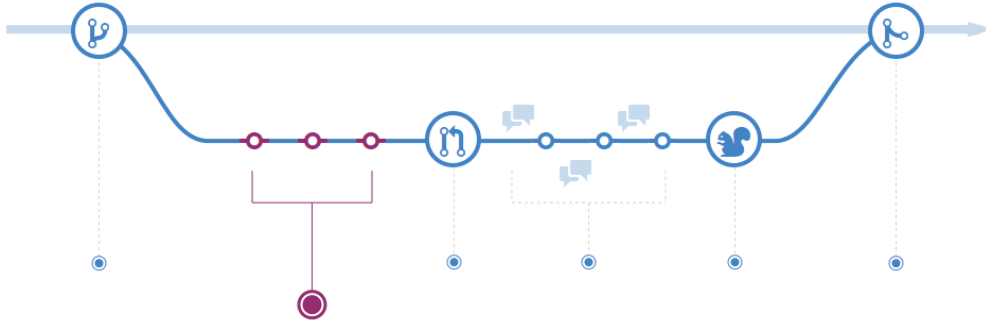
Initialize this repository with a README
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository



Make and commit changes



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



```
1- # hello-world
2
3 Hi Humans!
4
5 Hubot here, I like Node.js and Coffeescript (that's what I'm made of!).
6 I've had tacos on the moon and find them far superior to Earth tacos.
7
```

Commit changes

Finish README

And mention moon tacos

Commit directly to the `readme-edits` branch

Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

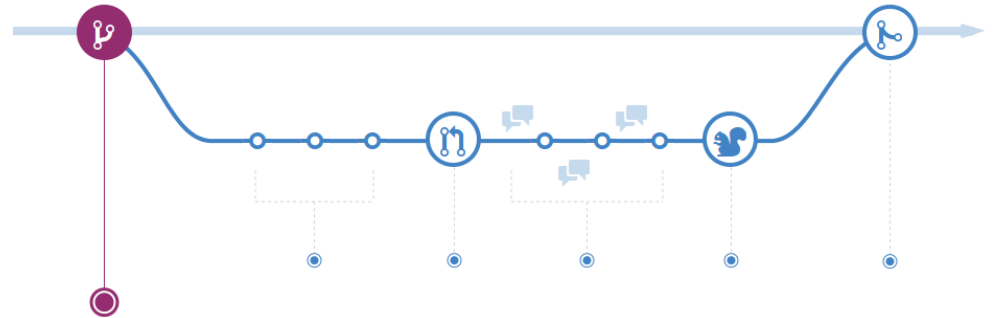


Real power of Github: Branching

- ▶ 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**.



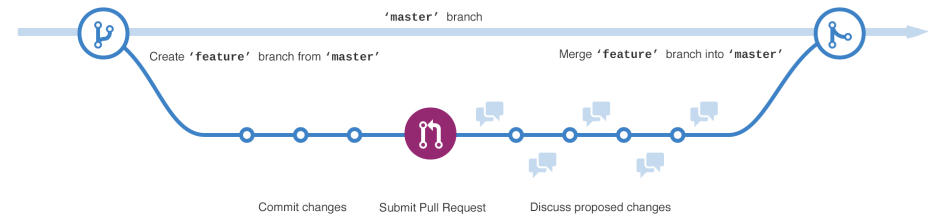
Just another repository — Edit

1 commit 2 branches

branch: `readme-edits` hello-world /

This branch is 0 commits ahead and 0 commits behind master

Fetching latest commit...



- ▶ 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 branches.
- ▶ After confirming the merge, other branches can be deleted.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Pull request successfully merged and closed
You're all set—the `readme-edits` branch can be safely deleted.

Delete branch

