Programming fundamentals with Python

Modules and packages

Pepe García jgarciah@faculty.ie.edu

Plan for today

- Errata & answer questions about previous sessions

Plan for today

- Errata & answer questions about previous sessions
- PYPI

Plan for today

- Errata & answer questions about previous sessions
- PYPI
- virtual environments

Plan for today

- Errata & answer questions about previous sessions
- PYPI
- virtual environments
- pip

Plan for today

- Errata & answer questions about previous sessions
- PYPI
- virtual environments
- pip
- Downloading & using Python libraries

Errata

The last day I wasn't able to run cells that contained imports of local python packages. This demonstrated to be a bug in VSCode's python plugin.

Q&A

- How recursion works?
- How mergesort recursion works?

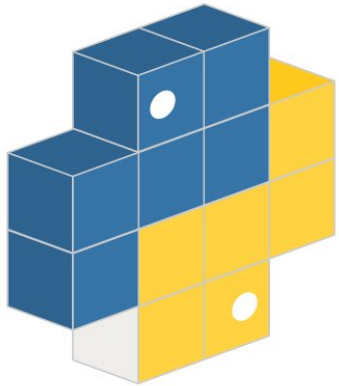
Python Package Index

In order to install libraries using pip we will first need to know which library to install. To make this search easier, the Python community maintains **Pypi**, the Python Package Index, a website with all the available libraries and instructions to install them.

https://pypi.org

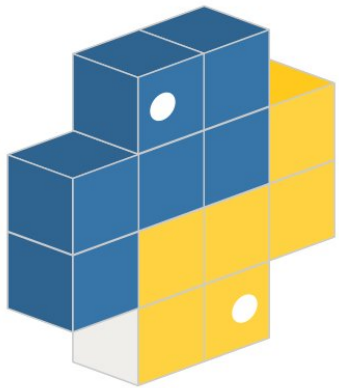
Python Package Index

The Python Package Index is a great place to find libraries you may be interested in.



Python Package Index

The Python Package Index is a great place to find libraries you may be interested in.



Searching in the Python Package Index

Let's search for some libraries in the Python Package Index...

Virtual environments

Virtual environments are a way to create a project with its own dependencies.

Virtual environments

Hint

In order to follow today's class from windows, run commands in **Anaconda Prompt**.

Creating virtual environments

In order to create a virtual environment, we'll use the `python -m venv` tool:

```
$ mkdir testing-virtualenvs
$ cd testing-virtualenvs
$ python -m venv env
```

This last command creates a subdirectory in the current directory that will contain all the meta information about the virtualenv. It's common to check that into `git`.

Activating a virtual environment

We can *activate* a virtual environment by running:

```
$ source env/bin/activate
```

pip

`pip` is a package manager for Python. We will use it through the command line:

```
$ <venv>/bin/pip
```

Or, in our case:

```
$ env/bin/pip
```

pip

There are a lot of commands available from pip

pip

There are a lot of commands available from pip

- install packages **`env/bin/pip install whatever`**

pip

There are a lot of commands available from pip

- install packages **`env/bin/pip install whatever`**
- uninstall packages **`env/bin/pip uninstall whatever`**

pip

There are a lot of commands available from pip

- install packages **`env/bin/pip install whatever`**
- uninstall packages **`env/bin/pip uninstall whatever`**
- show all installed packages **`env/bin/pip freeze`**

pip

There are a lot of commands available from pip

- install packages **`env/bin/pip install whatever`**
- uninstall packages **`env/bin/pip uninstall whatever`**
- show all installed packages **`env/bin/pip freeze`**

pip

There are a lot of commands available from pip

- install packages **`env/bin/pip install whatever`**
- uninstall packages **`env/bin/pip uninstall whatever`**
- show all installed packages **`env/bin/pip freeze`**

And... what can this **`whatever`** be? Anything we want to search a library for.

pip

```
$ python3 -m pip install mergesort
Collecting mergesort
  Downloading mergesort-0.0.1.tar.gz (779 bytes)
Building wheels for collected packages: mergesort
  Running setup.py clean for mergesort
Failed to build mergesort
Installing collected packages: mergesort
    Running setup.py install for mergesort ... done
Successfully installed mergesort-0.0.1
```

Troubleshooting

It's possible that on your windows computer `python` is not behaving as you expect. This tutorial is handy for these cases.

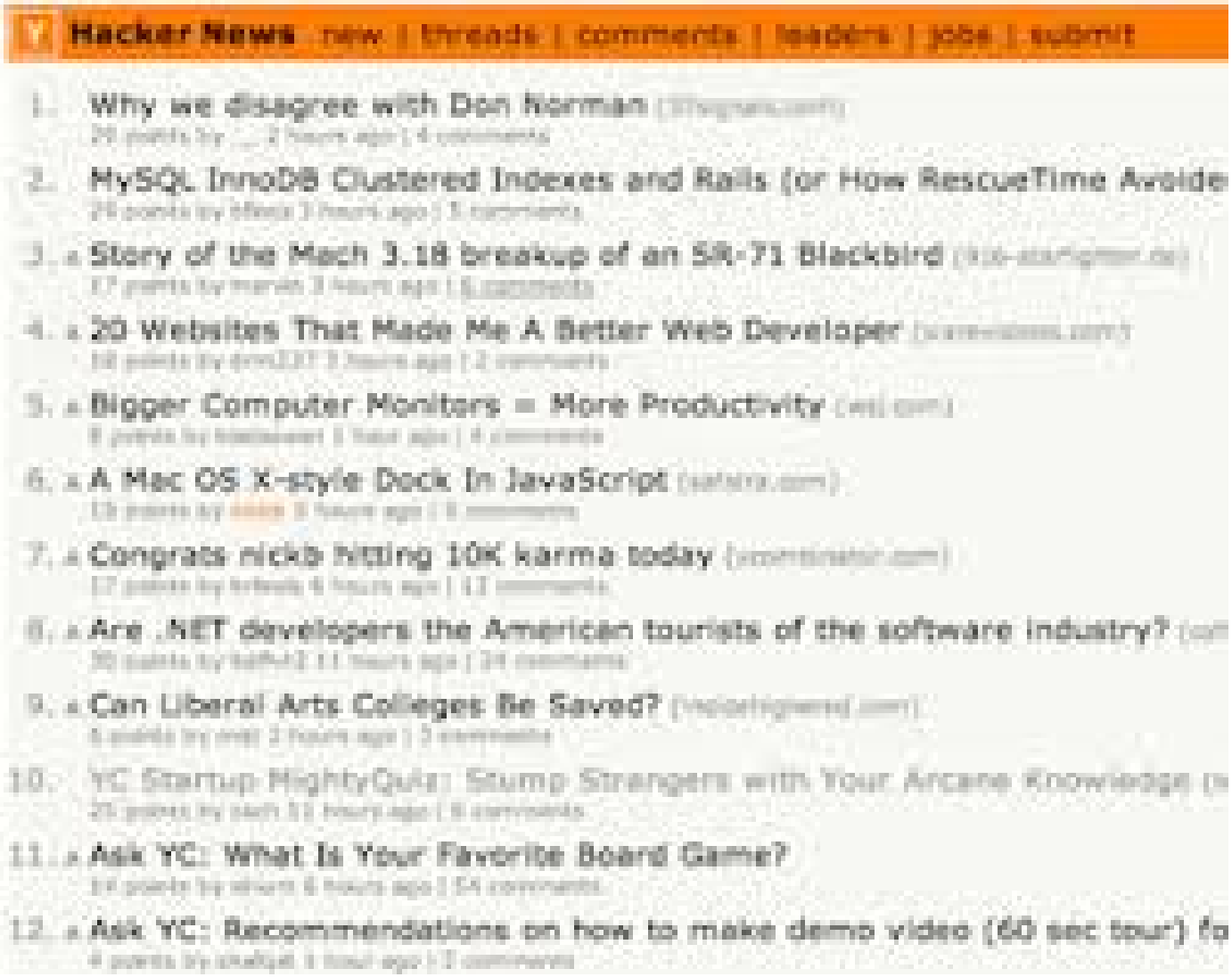
https://www.datacamp.com/community/tutorials/installing-anaconda-windows

Using third party libraries

Once we've found the library we want to use in our project, it's time to see how it's used. Usually libraries will have some documentation explaining how to use them.

The place where this doc should be is in the homepage of the library.

Using third party libraries



Using third party libraries

`hackernews-client` is a library for getting HackerNews (`http://news.ycombinator.com`) data programatically.

Exercise

Let's have 10 minutes to create a virtual environment that uses hackernews-client and a Python program that gets the 10 best stories from it (look in the docs how to get it).

Using third party libraries - HN

```
$ python3 -m pip install hackernews-client

Collecting hackernews-client
  Using cached hackernews-client-0.1.2b1.tar.gz (8.1 kB)
Building wheels for collected packages: hackernews-client
  Building wheel for hackernews-client (setup.py) ... error
  Running setup.py clean for hackernews-client
Failed to build hackernews-client
Installing collected packages: hackernews-client
    Running setup.py install for hackernews-client ... done
Successfully installed hackernews-client-0.1.2b1
```

Using third party libraries - HN

```
from hackernews import hn # import the hn module from hackernews packa
```

Using third party libraries - HN

```
from hackernews import hn

client = hn.NewsClient() # create a client
```

Using third party libraries - HN

```
from hackernews import hn

client = hn.NewsClient()

for story in client.get_best_story(10): # get 10 high score stories
    print(story.title)
```

Getting all dependencies for a project

When we're working on a project, we want to make it easy for people to get up to speed. We can do that for dependencies, so that new developers in the project don't need to install dependencies one by one.

Remember **`pip freeze`**? We'll use it to generate a file called **`requirements.txt`** that contains all library dependencies.

```
$ env/bin/pip freeze > requirements.txt
```

Getting all dependencies for a project

Finally, when we clone a repository that contains some Python dependencies, we can install them all in one go, with:

```
$ env/bin/pip install -r requirements.txt
```

Demo

Let's demo it with session-15 repository

Recap

- Answer questions about previous sessions

Recap

- Answer questions about previous sessions
- PYPI

Recap

- Answer questions about previous sessions
- PYPI
- virtual environments

Recap

- Answer questions about previous sessions
- PYPI
- virtual environments
- pip

Recap

- Answer questions about previous sessions
- PYPI
- virtual environments
- pip
- Downloading & using Python libraries