

django

configurazione

configurazione

per riuscire ad utilizzare correttamente **django** abbiamo bisogno di alcune tecnologie base:

python → È un linguaggio **general-purpose, multi-paradigma** e **open source** che supporta il paradigma object oriented, la programmazione strutturata e molte caratteristiche di programmazione funzionale.

pipenv/virtualenv → tool per la creazione di **ambienti python isolati**. Cioè cartelle contenenti tutti gli eseguibili necessari per un progetto python.

IDE → Un ambiente di sviluppo integrato è un software che, in fase di programmazione, supporta i programmatori nello sviluppo del codice sorgente di un programma.

controllo di versione → è una tipologia di controllo di versione che permette di tenere traccia delle modifiche e delle versioni apportate al codice sorgente di un software.



configurazione

nello specifico di **django**:

python → versione 3

pipenv/virtualenv → opzionale, altamente consigliato pipenv per avere ambienti python equivalenti

IDE → opzionale, consigliato PyCharm (per gli studenti si può chiedere la versione Pro gratuita utilizzando la mail del proprio istituto)

controllo di versione → opzionale, consigliato GitLab, gratuito con molte funzionalità interessanti tra cui CD/CI



python

Tutta la documentazione su python è reperibile su: <https://wiki.python.org>

installazione su sistemi linux → <https://wiki.python.org/moin/BeginnersGuide/Download>

arch -> preinstallato

centos/redhat/fedora → yum python3 python3-devel

debian/ubuntu → apt install python3.x python3.x-dev

installazione su MacOS → <https://installpython3.com/mac/>

installazione su Windows → download .exe from <https://www.python.org/downloads/>

attenzione a selezionare la voce “Add Python 3.X to PATH” durante l’installazione



- ☒ Install launcher for all users (recommended)
- ☒ Add Python 3.6 to PATH

python

verifica di funzionamento:

terminale (powershell nel caso di windows) ed eseguo → python

risultato atteso:

```
cicciodev@cicciarch ~$ python
Python 3.8.2 (default, Feb 26 2020, 22:21:03)
[GCC 9.2.1 20200130] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

```
In [1]: import sys
if sys.version_info >= (3, 0):
    print('Good job!')
else:
    print('Huston we have a problem\n'
          'Prova ad eseguire python3 nella shell.')
```

Good job!



pipenv



Assicurarsi di avere pip → `python3 -m pip --version`
o installarlo → <https://pip.pypa.io/en/stable/installing/>

installazione generica pipenv → `pip install pipenv`

documentazione pipenv → <https://pipenv-fork.readthedocs.io/en/latest/>

Perchè utilizzare pipenv?

Cito dalla documentazione: pipenv è pensato per fornire agli sviluppatori un metodo facile per creare un ambiente di lavoro. Inoltre, sempre dalla documentazione, pipenv è dotato di colori molto carini da terminale.



Integrated Development Environment → in pratica un software che aiuta i programmatori a sviluppare software

Installazione pycharm → <https://www.jetbrains.com/pycharm/download/>

Jetbrains Pro for Education → <https://www.jetbrains.com/community/education/#students>

è sufficiente utilizzare la mail studenti

Perchè utilizzare PyCharm?

PyCharm ha un tipo di progetto dedicato a Django, supporto per i Django templates, run/debug per Django server, completamento codice per django (basterebbe solo questo come motivo), il prof lo utilizza e quindi possiamo utilizzare le sue stesse shortcut.



Jupyter - opzionale -

Jupyter Notebook → un'applicazione (di tipo server-client che possiamo utilizzare sui nostri desktop) che permette di creare dei documenti con un “kernel” ovvero un unità computazionale che ci permette di eseguire codice python

Installazione Jupyter → `pip install jupyterlab`

documentazione → <https://jupyter.org/install.html>

Perchè utilizzare Jupyter?

Un'interfaccia efficace e flessibile per prendere appunti ed eseguire nel mentre codice python. Inoltre è un modo “sicuro” per utilizzare alcune librerie python come *numpy* o *pandas* senza dover installare *anaconda* a livello di sistema (che su sistemi linux non è una bellissima idea).



controllo di versione

Git → sistema free, open-source di controllo di versione distribuito

Installazione di git →

Linux → (apt|yum) install git

MacOs e Windows → <https://git-scm.com/downloads>

documentazione → <https://git-scm.com/doc>

account online → <https://gitlab.com> <https://bitbucket.org>

Perchè **NON** github?

Perchè è della Micros**t

Perchè **NON** la Micros**t?

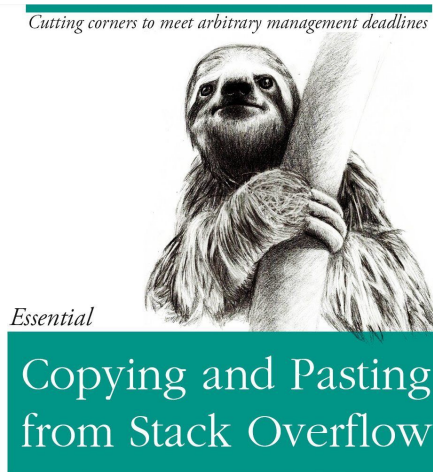
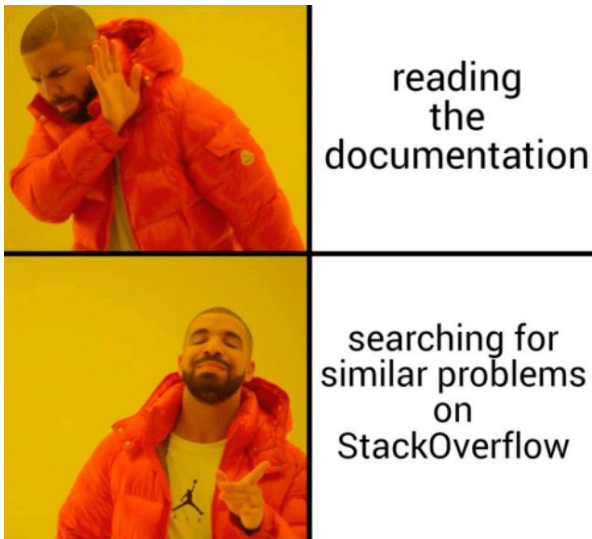
https://en.wikipedia.org/wiki/Criticism_of_Microsoft



documentazione utile

python → documentazione online <https://docs.python.org/3/>
“Python in a nutshell” di Alex Martelli (<http://www.aleax.it/>) è il miglior manuale per iniziare con python, viene citato da numerosi manuali python avanzati

django → uno dei framework meglio documentati in circolazione
<https://docs.djangoproject.com/en/2.2>



bisogno di aiuto?

django

francesco.faenza@unimore.it