

<https://tutorial.djangogirls.org/it/installation/>

**Se stai facendo il tutorial a casa**

Se stai facendo il tutorial a casa e non durante uno degli [eventi Django Girls](https://djangogirls.org/events/), puoi saltare questo capitolo e andare direttamente a [Come funziona Internet?](https://tutorial.djangogirls.org/it/how_the_internet_works/).

Copriamo questi argomenti nel tutorial completo, e questa è una pagina aggiuntiva contenente tutte le istruzioni di installazione. L'evento Django Girls include una "serata dedicata all'installazione" in cui installiamo tutto, in modo da non preoccuparcene durante il workshop, quindi questo capitolo è utile per chi partecipa ad uno dei nostri workshop.

Se lo trovi utile puoi seguire anche questo capitolo. Ma se vuoi iniziare a imparare prima di installare le cose sul tuo computer, salta questo capitolo. Ti spiegheremo l'installazione più avanti.

Buona fortuna!

**Installazione**

Nel workshop costruirai un blog, e ci sono alcuni task dedicati all'impostazione che sarebbe bello completare in anticipo, in modo che tu sia pronta a scrivere codice in giornata.

**Installare Python**

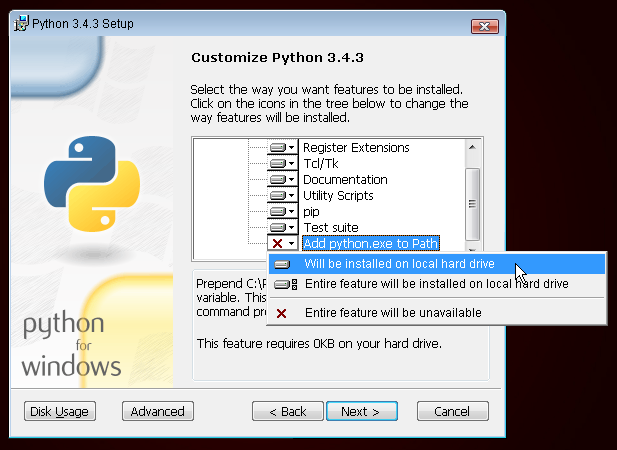
Questa sezione si basa su un tutorial fatto da Geek Girls Carrots (<https://github.com/ggcarrots/django-carrots>)

Django è scritto in Python. Abbiamo bisogno di Python per fare qualsiasi cosa in Django. Iniziamo con l'installazione! Vogliamo che sul tuo pc sia installato Python 3.4 quindi se hai una versione precedente, dovrai aggiornarlo.

**Windows**

Puoi scaricare Python per Windows dal sito web <https://www.python.org/downloads/release/python-343/>. Dopo aver scaricato il file **\*.msi**, lo dovresti eseguire (cliccaci sopra due volte) e segui le istruzioni. È importante ricordare il percorso (la directory) dove ha installato Python. Più tardi sarà necessario!

Una cosa a cui fare attenzione: sulla seconda schermata dell'installazione guidata, contrassegnata "Customize", assicurati di scorrere verso il basso e di scegliere l'opzione "Add python.exe to the Path", come illustrato qui:



**Linux**

È molto probabile che tu abbia Python già installato di default. Per controllare se ce l'hai già installato (e quale versione è), apri una console e digita il seguente comando:

$ python3 --version

Python 3.4.3

Se non hai Python installato o se vuoi una versione diversa, puoi installarla come segue:

**Debian o Ubuntu**

Digita questo comando nella tua console:

$ sudo apt-get install python3.4

**Fedora (fino a 21)**

Usa questo comando nella tua console:

$ sudo yum install python3.4

**Fedora (22+)**

Usa questo comando nella tua console:

$ sudo dnf install python3.4

**openSUSE**

Usa questo comando nella tua console:

$ sudo zypper install python3

**OS X**

Devi andare sul sito <https://www.python.org/downloads/release/python-343/> e scarica il programma d'installazione di Python:

* Scarica il file *Mac OS X 64-bit/32-bit installer*
* Fai doppio click su *python-3.4.3-macosx10.6.pkg* per eseguire il programma d'installazione.

Verifica che l'installazione si sia conclusa correttamente aprendo l'applicazione *Terminal* ed eseguendo il comando python3:

$ python3 --version

Python 3.4.3

Se hai dubbi o se qualcosa è andato storto e non hai idea di cosa fare dopo - chiedi al tuo insegnante! A volte le cose non vanno come dovrebbero ed è meglio chiedere aiuto a qualcuno con più esperienza.

**Preparare virtualenv e installare Django**

Una parte di questo capitolo si basa sui tutorial delle Geek Girls Carrots (<https://github.com/ggcarrots/django-carrots>).

Una parte di questo capitolo di basa sul [django-marcador tutorial](http://django-marcador.keimlink.de/" \t "_blank) sotto licenza Creative Commons Attribution-ShareAlike 4.0 International License. Il tutorial di django-marcador è protetto da copyright di Markus Zapke-Gründemann et al.

**Ambiente virtuale**

Prima di installare Django, ti vogliamo far installare uno strumento estremamente utile per tenere in ordine l'ambiente in cui programmerai sul tuo computer. Potresti saltare questo passaggio, ma è caldamente consigliato soffermarsi. Se inizi con la migliore configurazione possibile, ti risparmierai un sacco di problemi per il futuro!

Per cui, creiamo ora un **ambiente virtuale** (chiamato anche un *virtualenv*). Virtualenv isolerà la tua configurazione di Python/Django in base ai diversi progetti. Questo significa che qualunque modifica farai su un sito non avrà alcun effetto su tutti gli altri che stai sviluppando. Chiaro ora?

Tutto quello che devi fare è trovare una cartella in cui vuoi creare il virtualenv; la tua home directory, ad esempio. Su Windows potrebbe essere C:\Users\Name (dove Name è il nome del tuo login).

Per questo tutorial useremo una nuova directory djangogirls dalla tua home directory:

mkdir djangogirls

cd djangogirls

Ora creeremo un virtualenv dal nome myvenv. Questo è il formato del comando generale:

python3 -m venv myvenv

**Windows**

Per creare un nuovo virtualenv è necessario aprire la console (ti abbiamo spiegato come fare nei capitoli precedenti, ricordi?) ed esegui C:\Python34\python -m venv myvenv. Il risultato somiglierà a questo:

C:\Users\Name\djangogirls> C:\Python34\python -m venv myvenv

dove C:\Python34\python è la directory in cui precedentemente hai installato Python e myvenv è il nome del tuo virtualenv. Puoi utilizzare qualsiasi altro nome, ma attieniti a utilizzare le minuscole, a non usare spazi, accenti o caratteri speciali. È meglio usare un nome breve dal momento che dovrai digitarlo spesso!

**Linux e OS X**

Creare un virtualenv su Linux e OS X è semplice, basta digitare: python3 -m venv myvenv. Il risultato sarà simile a questo:

~/djangogirls$ python3 -m venv myvenv

myvenv è il nome del tuo virtualenv. Puoi usare qualsiasi altro nome, ma utilizza solo minuscole e niente spazi. Sarebbe meglio se il nome fosse corto perchè lo dovrai digitare molte volte!

**NOTA:** Avviare il virtualenv su Ubuntu 14.04 in questa maniera al momento darà il seguente errore:

Error: Command '['/home/eddie/Slask/tmp/venv/bin/python3', '-Im', 'ensurepip', '--upgrade', '--default-pip']' returned non-zero exit status 1

Per aggirare il problema utilizza, invece del precedente, il comando virtualenv.

~/djangogirls$ sudo apt-get install python-virtualenv

~/djangogirls$ virtualenv --python=python3.4 myvenv

**Lavorare con virtualenv**

Il comando sopra specificato, creerà una cartella dal nome myvenv (o col nome che hai scelto) che contiene il tuo virtual environment (ovvero un mucchio di files e cartelle).

**Windows**

Avvia il tuo virtualenv digitando:

C:\Users\Name\djangogirls> myvenv\Scripts\activate

**Linux e OS X**

Avvia il tuo virtualenv digitando:

~/djangogirls$ source myvenv/bin/activate

Ricordati di sostituire myvenv con il nome virtualenv che hai scelto!

**Nota:** a volte il comando source potrebbe non essere disponibile. In quel caso prova ad usare questo:

~/djangogirls$ . myvenv/bin/activate

Saprai con certezza che hai avviato virtualenv quando vedrai che il prompt dei comandi nella console si presenta così:

(myvenv) C:\Users\Name\djangogirls>

oppure:

(myvenv) ~/djangogirls$

Fai caso al prefisso (myvenv)!

Quando si lavora all'interno di un ambiente virtuale, python farà automaticamente riferimento alla versione corretta da utilizzare. Per cui puoi digitare python invece python3.

OK, abbiamo tutte le dipendenze importanti pronte. Finalmente possiamo installare Django!

**Installare Django**

Ora che hai iniziato ad utilizzare il tuo virtualenv, puoi installare Django usando pip. Nella console, esegui pip install django==1.8 (nota che usiamo un doppio simbolo di uguale: ==).

(myvenv) ~$ pip install django==1.8

Downloading/unpacking django==1.8

Installing collected packages: django

Successfully installed django

Cleaning up...

Su Windows

Se ottieni un errore quando chiami pip sulla piattaforma Windows controlla se il pathname del tuo progetto contiene spazi, accenti o caratteri speciali (i.e. C:\Users\User Name\djangogirls). Se è così ti conviene spostarlo in un altro path senza spazi, accenti o caratteri speciali (il suggerimento è: C:\djangogirls). Dopo averlo spostato, prova ad eseguire di nuovo il comando di cui sopra.

su Linux

Se ottieni un errore quando esegui il comando pip su Ubuntu 12.04, prova ad eseguire python -m pip install -U --force-reinstall pip per risolvere il problema.

Questo è tutto! Sei (finalmente) pronto/a a creare un'applicazione Django!

**Installare un editor di codice**

Sono disponibili diversi editor e la scelta di uno piuttosto che un altro dipende principalmente dal gusto personale. La maggior parte dei programmatori Python usa complessi ma estremamente potenti IDE (ambienti di sviluppo integrati), come PyCharm. Tuttavia, dal momento che sei ancora agli inizi non è l'editor più appropriato; quelli che ti suggeriremo noi sono ugualmente potenti ma molto più semplici da utilizzare.

I nostri suggerimenti sono riportati qui di seguito, ma sentiti libero/a di chiedere al tuo coach quali sono le sue preferenze in materia di editor, in questo modo sarà più semplice per il tuo coach aiutarti.

**Gedit**

Gedit è un editor open-source e gratuito, disponibile per tutti i sistemi operativi.

[Scaricalo qui](https://wiki.gnome.org/Apps/Gedit#Download)

**Sublime Text 3**

Sublime Text è uno tra gli editor più utilizzati. Ha un periodo di prova gratuito. È molto facile da installare e da utilizzare ed è disponibile per tutti i sistemi operativi.

[Scaricalo qui](https://www.sublimetext.com/3)

**Atom**

Atom è un nuovo editor di codice creato da [GitHub](https://github.com/" \t "_blank). È gratuito, open-source, facile da installare e da usare. È disponibile per Windows, OSX e Linux.

[Scaricalo qui](https://atom.io/)

**Perché installiamo un editor di codice?**

Forse ti stai chiedendo per quale motivo installiamo questo editor di codice invece di usare un applicazione come Word or Blocco Note.

Il primo motivo è che il codice deve essere **testo semplice**, e il problema con programmi come Word e Textedit è che in realtà non producono testo semplice. Producono testo RTF (con caratteri e formattazione), utilizzando formati personalizzati come [RTF (Rich Text Format)](https://en.wikipedia.org/wiki/Rich_Text_Format).

La seconda ragione è che i code editor sono specializzati per programmare, perciò hanno molte funzionalità utili, ad esempio diversi colori per evidenziare frammenti di codice con diversi significati, o l'inserimento automatico del secondo paio di virgolette.

Vedremo tutto ciò più tardi. Il tuo fidato code editor sarà presto uno dei tuoi strumenti preferiti :)

**Installare Git**

**Windows**

È possibile scaricare Git da [git-scm.com](https://git-scm.com/" \t "_blank). Puoi saltare tutti i passaggi tranne uno. Nel quinto passaggio, dal titolo "Regolazione della variabile PATH di sistema", scegli "Esegui Git e gli strumenti Unix associati dalla riga di comando di Windows" (l'opzione in basso). A parte questo, i valori predefiniti vanno bene. 'Checkout Windows-style' e 'commit Unix-style line endings' vanno bene.

**MacOS**

Scarica Git da [git-scm.com](https://git-scm.com/) e segui le istruzioni.

**Linux**

Se non è già installato, git dovrebbe essere disponibile tramite il gestore di pacchetti, prova a cercare:

sudo apt-get install git

# oppure

sudo yum install git

# oppure

sudo zypper install git

**Creare un account GitHub**

Vai su [GitHub.com](https://www.github.com/) e iscriviti per un nuovo account gratuito.

**Creare un account PythonAnywhere**

Ora è il momento di registrarsi per un account gratuito "Beginner" su PythonAnywhere.

* [www.pythonanywhere.com](https://www.pythonanywhere.com/)

**Nota** Quando scegli il tuo nome utente, tieni conto che l'URL del tuo blog diventerà iltuousername.pythonanywhere.com, quindi scegli il tuo nickname oppure un nome ispirato a ciò su cui si basa il tuo blog.

**Inizia la lettura**

Complimenti, ora sei pronta! Se hai un po' di tempo prima del workshop, potrebbe essere utile cominciare a leggere i capitoli iniziali:

* [Come funziona Internet](https://tutorial.djangogirls.org/it/how_the_internet_works/)
* [Introduzione alla riga di comando](https://tutorial.djangogirls.org/it/intro_to_command_line/)
* [Introduzione a Python](https://tutorial.djangogirls.org/it/intro_to_command_line/)
* [Che cos'è Django?](https://tutorial.djangogirls.org/it/django/)