

Come sviluppo le applicazioni web

Andrea Lazzarotto – andrealazzarotto.com

Mi presento

Sviluppatore software e
consulente informatico

Autore di vari software open
source (script, RecuperaBit, ...)

Socio fondatore GrappaLUG

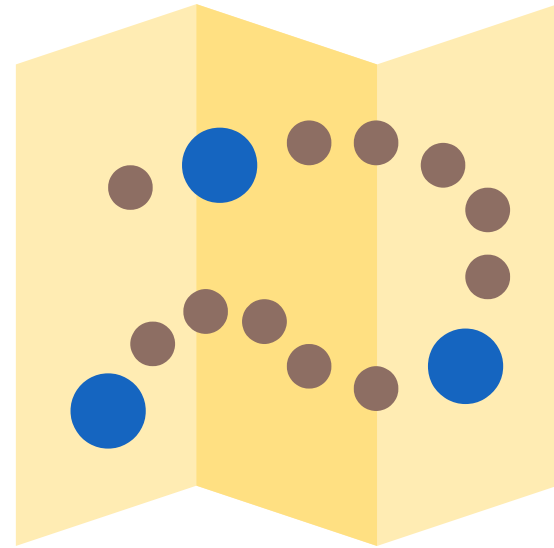


Percorso

A proposito di PHP

Sviluppare il software

Fare il deploy



A proposito di PHP

Sviluppo web

Tantissimi usano PHP (e MySQL)

WordPress, Wikipedia, Digg, Facebook

Parliamone...

Inconsistenza dei nomi

strpos

str_rot13

bin2hex

strptime

usleep

microtime

«`htmlspecialchars` fu una tra le prime funzioni.
Al tempo in cui PHP aveva meno di 100 funzioni e
**il meccanismo di hashing per i nomi di
funzione era `strlen()`.**»

— Rasmus Lerdorf, creatore di PHP

Riguardo a `htmlspecialchars`

I'm sick of this s*, don't deny
You're a waste of time
I'm sick of this s***, don't ask why**

— Avril Lavigne, "I Can Do Better"

Tipi e operatori

NULL < -1

NULL == 0

2000 == "2e3"

Tutti questi confronti
danno come risultato il
valore **TRUE**

«Non so come fermarlo, non c'era nessuna intenzione di scrivere un linguaggio di programmazione. [...] **Non ho idea** di come si scriva un linguaggio di programmazione»

— Rasmus Lerdorf, creatore di PHP

Ah ecco...

You're so full of s*
I can't stand the way you act
I just can't comprehend**

— Avril Lavigne, "I Can Do Better"

Messaggi di errore

```
<?php $x = ?>
```

syntax error, unexpected ';'

```
function foo(string $s){}  
foo("hello world");
```

Argument 1 passed to foo() must be an instance of string, string given

«Ero molto, molto scarso nello scrivere i parser.
Sono **ancora** molto scarso nello scrivere i parser.»
— Rasmus Lerdorf, creatore di PHP

Chi l'avrebbe mai detto!

**I will drink as much limoncello as I can
And I'll do again and again**

– Avril Lavigne, “I Can Do Better”

Sviluppare il software

Oltre a bere per dimenticare...



Andrea Lazzarotto 😊 produttivo.



27 settembre alle ore 0:21 ·  ▼

Two things can solve all problems in life:
Python and limoncello, of course.



Mi piace



Commenta



Condividi

Python

I tipi di dato hanno senso

I messaggi di errore hanno senso

Facilissimo e “batterie incluse”

**It's a state of bliss, you think you're dreaming
It's the happiness inside that you're feeling
It's so beautiful it makes you wanna cry**

— Avril Lavigne, "Innocence"

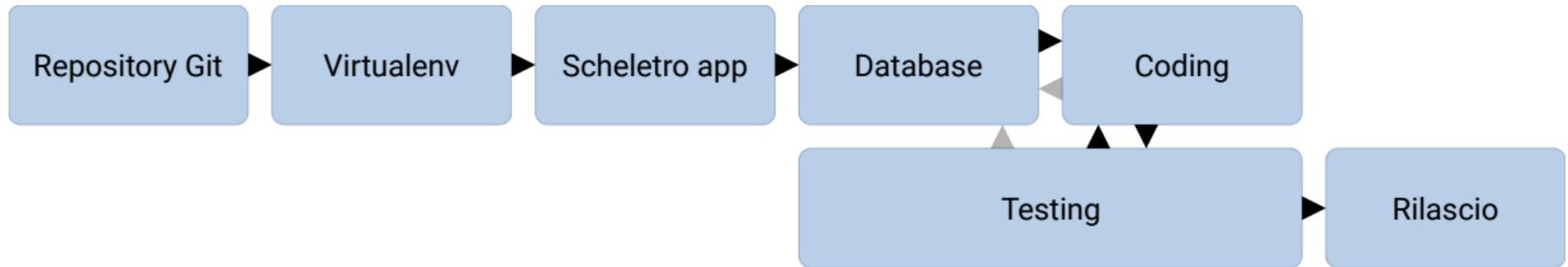
Flask

Micro-framework web per Python

Poco invasivo, veloce da imparare

Estendibile con altri moduli (es. Flask-Login)

Il workflow



Git

Creazione repository:

```
$ mkdir banana  
$ cd banana  
$ git init  
$ git remote add origin [...]
```

In remoto si può usare
GitHub o Bitbucket

File `.gitignore`:

```
__pycache__/  
*.py[co]  
*$py.class  
*.so  
*.db  
*-journal  
config.py  
venv
```

Virtualenv

Racchiude tutte le dipendenze in una directory!

```
$ pip install virtualenv
```

Nel nostro progetto:

```
$ virtualenv venv  
$ source venv/bin/activate.fish
```

Per chi usa Bash, c'è lo script **activate**

Scheletro del progetto

Struttura:

banana/	
├── banana.py	Applicazione Flask
├── banana.wsgi	Configurazione per Apache
├── config.py	File di configurazione (escluso da Git)
├── config.py.example	File di esempio (parametri fittizi)
├── requirements.txt	Moduli Python da installare (Flask e altro)
├── static	
│ ├── favicon.ico	Favicon (in realtà una PNG 128x128)
│ └── robots.txt	Indicazioni per i motori di ricerca
└── templates	
└── home.html	Template Jinja

File principale

```
from flask import Flask, render_template, request, send_from_directory

app = Flask(__name__)

@app.route("/")
def index():
    return render_template('home.html')

@app.route('/favicon.ico')
@app.route('/robots.txt')
def static_from_root():
    return send_from_directory(app.static_folder, request.path[1:])

if __name__ == '__main__':
    app.run(port=5000, debug=True)
```


File per Apache WSGI

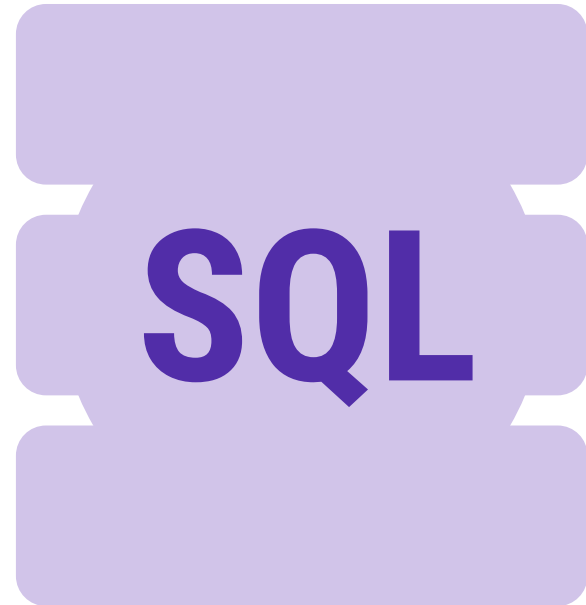
```
import os
import sys

wd = os.path.dirname(os.path.realpath(__file__))
sys.path.insert(0, wd)

from banana import app as application
```

Esempio di DB

```
CREATE TABLE `guestbook` (  
    `date`    TEXT NOT NULL,  
    `name`    TEXT NOT NULL  
);  
  
$ sqlite3 data.db < data.sql
```

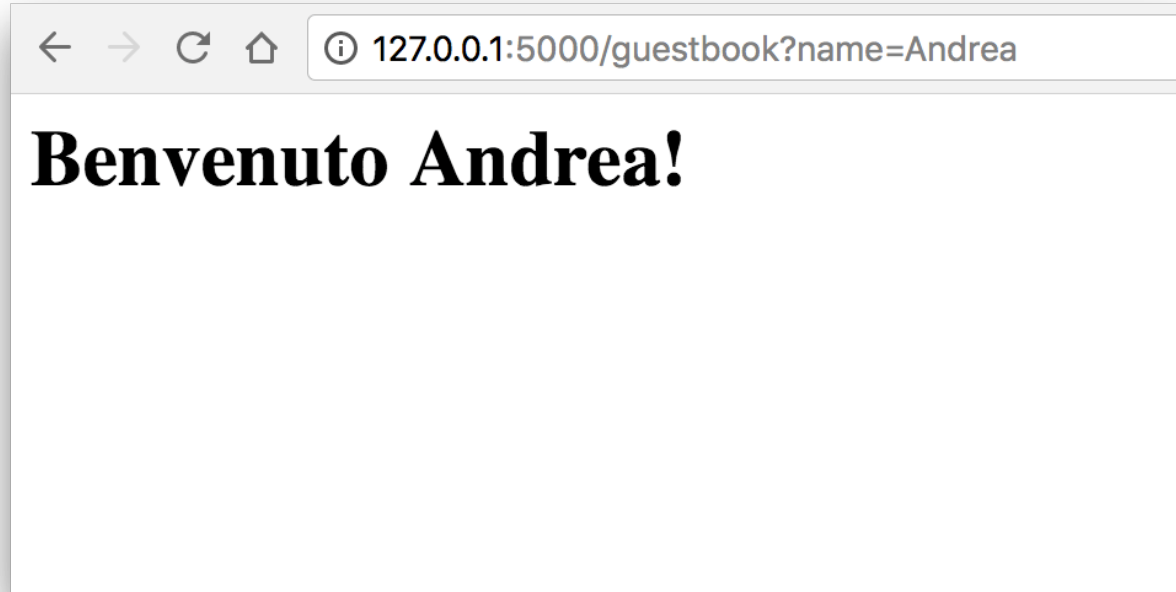


Aggiungiamo una rotta

```
@app.route("/guestbook")
def guestbook():
    name = request.args.get('name')
    if name:
        message = 'Benvenuto {}'.format(name)
        # TODO: inserire visita nel database con timestamp
    else:
        message = 'Mmm... chi sei?'
    return render_template('guestbook.html', message=message)
```

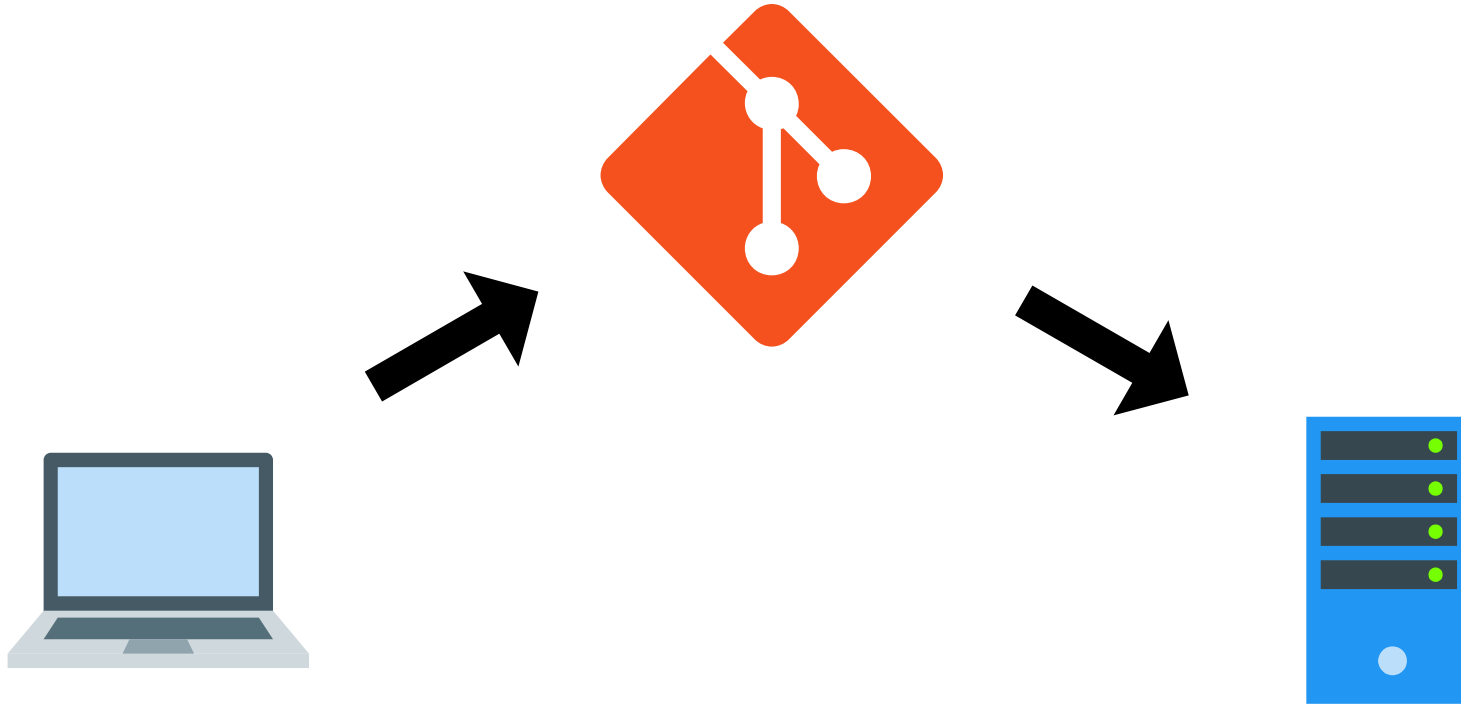
Il template `guestbook.html` contiene: `<h1>{{ message }}</h1>`

Avviando ./banana.py



Fare il deploy

Flusso del codice



Profilo utente sul server

```
# cd /var/www/  
# mkdir Banana  
# adduser banana  
# chown banana:banana Banana  
# su banana
```



Clone del repository

```
$ git clone git@github.com:Lazza/flask-banana.git /var/www/Banana
$ cd /var/www/Banana
$ pip install -u -r requirements.txt
$ sqlite3 data.db < data.sql
$ cp config.py.example config.py
$ nano config.py
```


Configurazione di Apache

```
<VirtualHost *:443>
  ServerName example.com

  ErrorLog ${APACHE_LOG_DIR}/banana_error.log
  CustomLog ${APACHE_LOG_DIR}/banana.log combined

  Alias /.well-known /var/www/Banana/.well-known

  WSGIDaemonProcess banana user=banana group=banana threads=20
  WSGIScriptAlias / /var/www/Banana/banana.wsgi

  <Directory /var/www/Banana>
    WSGIProcessGroup banana
    WSGIApplicationGroup %{GLOBAL}
    Order deny,allow
    Allow from all
  </Directory>

  SSLEngine On
  SSLCertificateFile /path/fullchain.pem
  SSLCertificateKeyFile /path/privkey.pem
</VirtualHost>
```

Salvato in `sites-available/banana.conf`

`*:80` fa solo redirect a HTTPS

`# a2ensite banana`

Successivi aggiornamenti

```
dev@laptop $ git push
```

```
banana@server $ git pull
```

```
root@server # service apache2 reload
```

Grazie per l'attenzione

Andrea Lazzarotto

@thelazza

<https://andrealazzarotto.com>