



[www.inmagik.com](http://www.inmagik.com)

# FULLSTACK WEB DEVELOPMENT



introduzione ad uno stack opensource basato su django e react

Ing. Mauro Bianchi  
[mauro.bianchi@inmagik.com](mailto:mauro.bianchi@inmagik.com)

# COSA VEDREMO

Sviluppo di software per il web con strumenti e framework opensource

Gestione del codice sorgente con git

Progettazione di una base dati per gestire informazioni dinamiche

Implementazione di un servizio web (API) per interagire con la base dati

Sviluppo di un sito pubblico con React (Javascript + HTML)

<https://github.com/inmagik/workshop-its>

# IL WEB MODERNO

## Prodotti Software

Siti statici

Siti dinamici

PWA

CMS

Ecommerce

Applicativi web

API

## Tecnologie

HTML

CSS

Javascript

SVG

JSON

websocket

...

# SVILUPPARE CON STRUMENTI OPENSOURCE

## Pro

Accesso a strumenti avanzati senza costi di ingresso

Approccio aperto alla conoscenza

Community aperte

## Attenzione a

Verificare la compatibilità delle licenze

Mancanza di supporto a pagamento

Difficoltà nell'individuare gli strumenti giusti

Documentazione spesso solo in inglese

# STRUMENTI PER LO SVILUPPO WEB

Un editor di testo:

- visual studio code: <https://code.visualstudio.com>
- atom: <https://atom.io/>

Una shell unix-like

- bash, zsh, sh o simili su sistemi linux/mac
- windows: mingw, git bash, tmux

Un browser adatto allo sviluppo:

- Google Chrome
- Mozilla Firefox

# STRUMENTI (2)

Git

- <https://git-scm.com/>
- Windows: <https://gitforwindows.org/>

Interprete python

- <https://www.python.org/>

Runtime javascript (node)

- <https://nodejs.org/it/>

# CASO PRATICO

Costruiremo un sito dinamico per un museo

BACKEND (python/django)

Database per la memorizzazione delle informazioni

Sito di amministrazione per la gestione dei contenuti

API per esporre i dati

FRONTEND (javascript/react)

Sito pubblico

# APPLICATIVO SERVER: tecnologie

Linguaggio: python 3

<https://www.python.org/>

Ambiente di sviluppo: virtualenv/pipenv

<https://virtualenv.pypa.io/en/latest/>

Framework: django

<https://www.djangoproject.com/>



# APPLICATIVO SERVER

django

“The Web framework for perfectionists with deadline”

Punti di forza

- Modello MVC
- ORM
- admin
- applicazioni riusabili
- python (leggibilità ed ecosistema)

# APPLICATIVO SERVER

```
cd museum-server
```

```
# creiamo un ambiente python isolato
```

```
# e installiamo i requisiti
```

```
virtualenv -p python3 env
```

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

```
pip install -r requirements.txt
```

```
# creiamo il progetto django
```

```
django-admin.py startproject museum_server
```

# APPLICATIVO SERVER

Proviamo il web server di sviluppo

```
cd museum_server
```

```
python manage.py runserver
```

Browse:

<http://localhost:8000>

# APPLICATIVO SERVER

Qualche comando di management di django:

```
python manage.py migrate
```

```
python manage.py createsuperuser
```

```
python manage.py runserver
```

# APPLICATIVO SERVER - modello dati

- App riusabili
- Modello dati dichiarativo
- ORM
- django-admin

<https://docs.djangoproject.com/en/2.1/topics/db/models/>

# APPLICATIVO SERVER - API REST

- django rest framework

<https://www.django-rest-framework.org>

- Implementiamo web services per ottenere i dati da altri applicativi (API REST)
  - lista artisti
  - dettaglio artista e relative opere

# APPLICATIVO SERVER - API REST

django-rest-framework

E un framework per realizzare API REST integrato con ORM django

Serializer:

oggetto python -> rappresentazione “serializzabile”

ViewSet:

Insieme di viste (=endpoints,urls)

# FRONTEND: tecnologie

Javascript

ECMAScript 6 (+Babel\*)

<https://nodejs.org/it/> (raccomandato anche yarn: <https://yarnpkg.com/>)

React

“A JavaScript library for building user interfaces”

<https://reactjs.org/>

\*Create React App

Set up a modern web app by running one command.

<https://facebook.github.io/create-react-app/>



# REACT

Sito ufficiale

<https://reactjs.org/tutorial/tutorial.html>

<https://reactjs.org/docs/getting-started.html>

<https://overreacted.io/>

<https://github.com/enaqx/awesome-react>

# REACT

Approccio dichiarativo: decido cosa voglio vedere

React manipola il DOM in modo efficiente per mantenere il documento (DOM) sincronizzato con lo stato desiderato (espresso in javascript/jsx).

Differente da un approccio “imperativo” (tipo jQuery)

React si basa sul concetto di componenti e mette a disposizione delle API per fare operazioni durante i vari momenti del “ciclo di vita” di un componente.

<https://reactjs.org/docs/design-principles.html>

# REACT: concetti

## Componenti

E' l'unità di composizione delle funzionalità in react, possono essere espressi come classi o come funzioni  
“Components let you split the UI into independent, reusable pieces, and think about each piece in isolation.”

## JSX

E' un modo di scrivere HTML all'interno di javascript (e viceversa) su cui si basa fortemente il codice scritto in react

## props & state

Sono le proprietà di un componente (rispettivamente passate dall'esterno e gestite dall'interno) che determinano come dovrebbe essere visualizzato. Cambiandole l'html del componente viene rivalutato

$$\text{HTML} = f(\text{props}, \text{state})$$

# REACT: ecosistema

react-router-dom (client-side routing)

<https://reacttraining.com/react-router>

lodash (functional helpers)

<https://lodash.com/>

reactstrap (bootstrap components)

<https://reactstrap.github.io/>

...

# FRONTEND: creazione progetto

```
npx create-react-app museum-frontend
```

```
cd museum-frontend
```

```
yarn start
```

```
# (or npm start)
```

Browse:

<http://localhost:3000>

# FRONTEND: aggiunta dipendenze

```
#our preferred css framework  
yarn add bootstrap
```

```
# sass  
yarn add node-sass
```

```
# abort controller.. (for stopping fetch crossbrowser)  
yarn add abort-controller
```

```
#... any other dependency
```

# CREATE-REACT-APP

Sintassi js avanzata (es6 + proposals)

Configurazione automatica webpack + babel

Server di sviluppo con autoreload

Supporto SASS

Comando di build

Proxy http

# FRONTEND: DESIGN SYSTEM, STYLE, WIDGETS

Bootstrap

<https://getbootstrap.com/>

SASS

<https://sass-lang.com/>

LIGHTBOX

<https://github.com/frontend-collective/react-image-lightbox>



# FRONTEND: strategie di sviluppo

Pagine

Componenti riusabili

Caricamento dati

Widget riusabili

Framework CSS/Stile CSS/conflitti namespaces