

# Metodologie per la Programmazione per il Web - MF0437

## *What's next?*

Docente

Giancarlo **Ruffo** [ [giancarlo.ruffo@uniupo.it](mailto:giancarlo.ruffo@uniupo.it) ]

Informazioni, materiale e risorse su:

moodle [ <https://www.dir.uniupo.it/course/view.php?id=16455> ]

- \* Panoramica
- \* Novità JS, Typescript
- \* MVC, Single Page Applications, API design, Frameworks
- \* Angular vs Vue vs Ember vs React

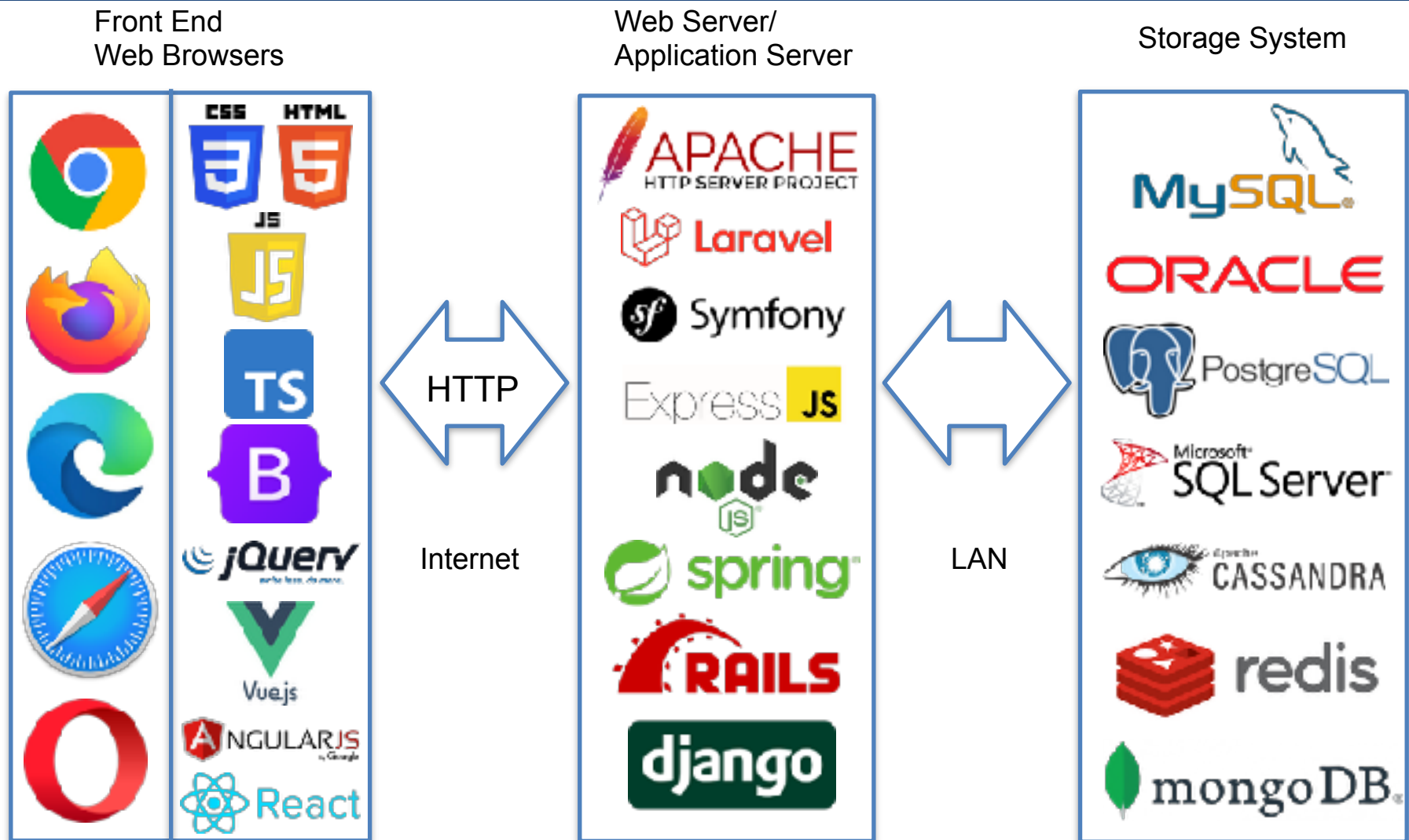
- \* **Panoramica**

- \* Novità JS, Typescript

- \* MVC, Single Page Applications, API design, Frameworks

- \* Angular vs Vue vs Ember vs React

# Architettura applicazione Web full stack



# Stack: front-end development

\*Sostanzialmente, la scelta non esiste:



\*In pratica, lo sviluppo lato client è "facilitato" da librerie, toolkit e framework



fuori moda... ma molto presente in sistemi già esistenti per cui sviluppare evolutive



per sviluppare strutture dinamiche di dati per web application



estende JS in JSX per sviluppare single page application scalabili



framework leggero per applicazioni single-page

\*Strumenti per o sviluppo veloce di pagine Web



**bootstrap:** collezione di template 'responsive'

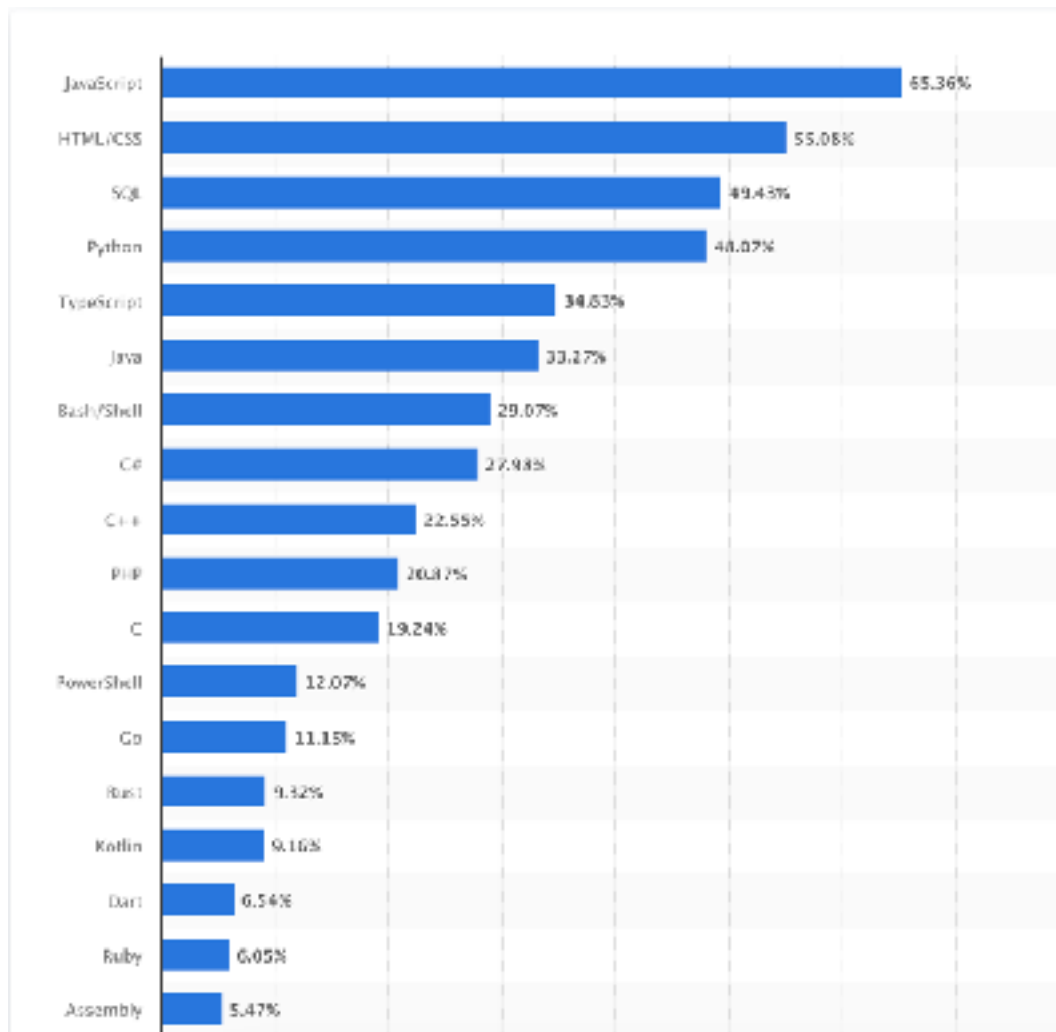


insieme di componenti html, css, js per sviluppo veloce di applicazioni 'mobile-first'

# Stack: back-end development

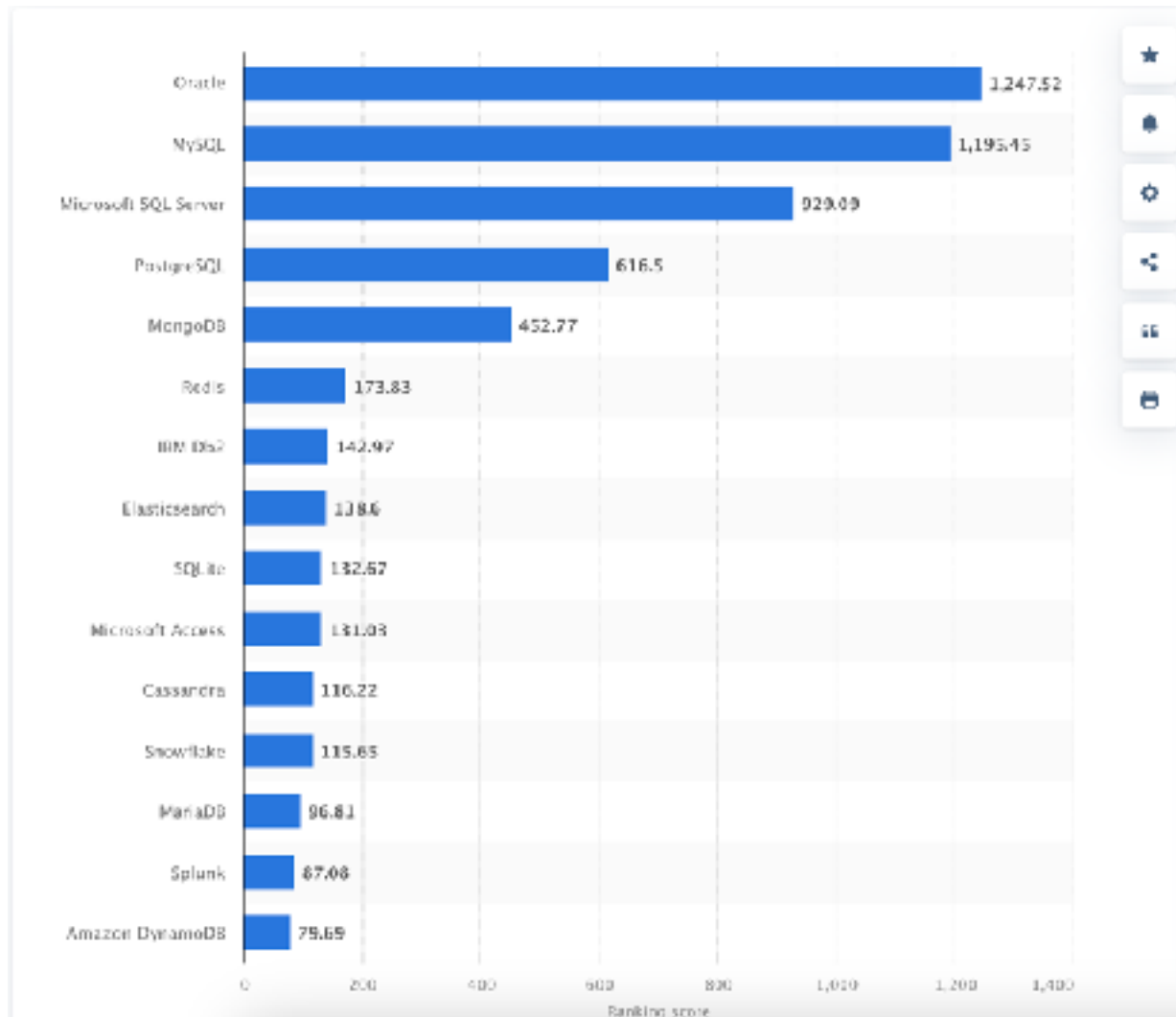
- \* Qui abbiamo un'ampia scelta, sia a livello di linguaggio (e framework), ad esempio
  - \* Python, PHP, Ruby, Java, and Scala
  - \* Django, Laravel, Ruby on Rails, Spring, and Play
- \* Ampia scelta pure per quanto riguarda i dbms (in ordine di popolarità nel 2023)
  - \* Oracle, MySQL, Microsoft SQL Server, PostgreSQL, and MongoDB, Redis, IBM Db2, Elasticsearch, SQLite, Microsoft Access, ...
- \* HTTP Server: Apache, Nginx, Cloudflare, Microsoft IIS, ... (davvero pochi altri)
- \* Sistemi operativi: Linux, Windows, MacOS - a volte questa scelta comporta una scelta quasi obbligata di altre componenti dello stack

# Programming languages used by software developers worldwide as of 2022



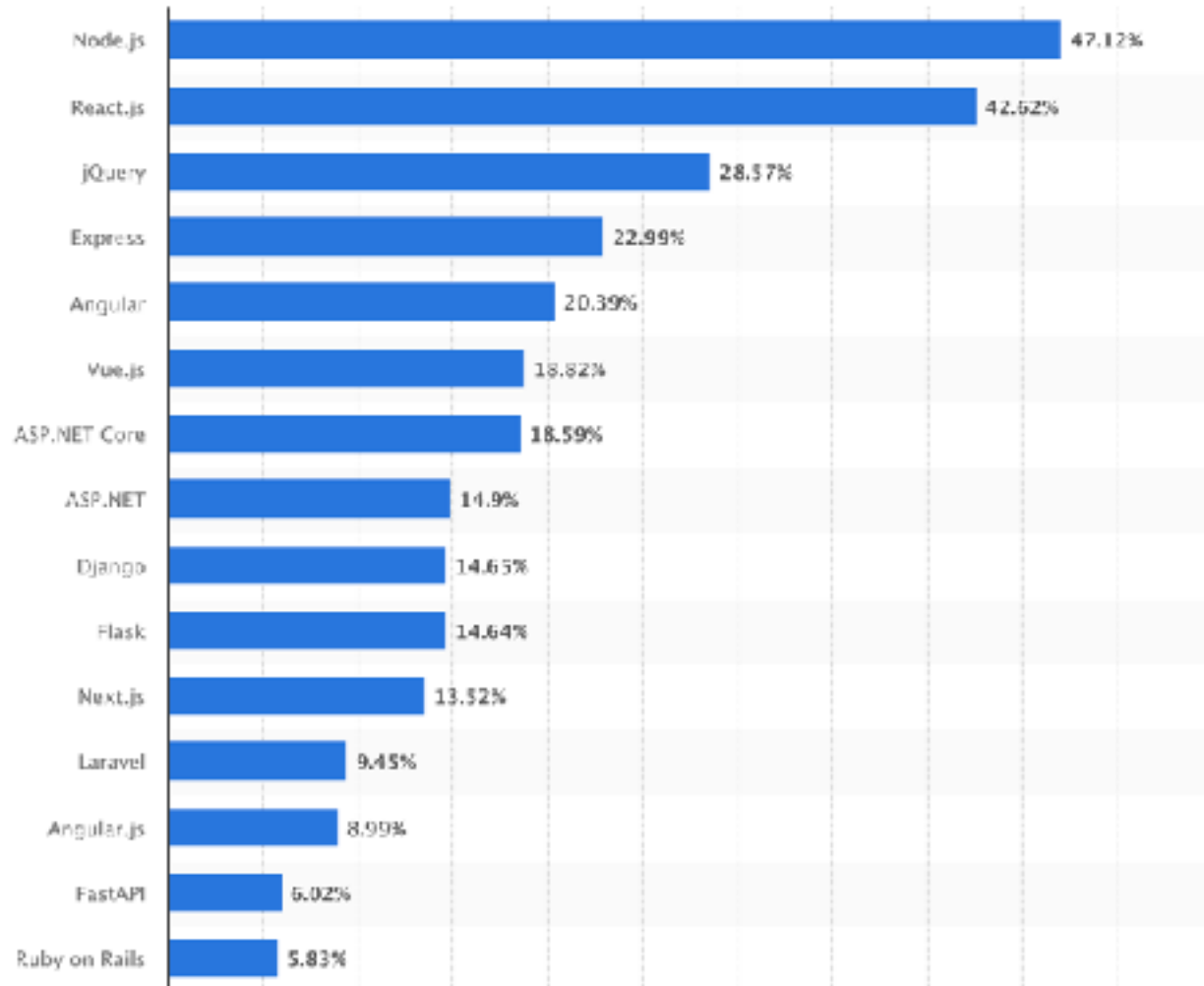
<https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>

# Ranking of the most popular database management systems worldwide, as of Feb 2023





# Most used web frameworks among developers worldwide, as of 2022



# Stack: opzioni principali

\* LAMP:



+



APACHE  
HTTP SERVER PROJECT

+



+



\* MEAN:



mongoDB<sup>+</sup>

Express



ANGULARJS<sup>+</sup>  
by Google



\* MERN:



mongoDB<sup>+</sup>

Express



React<sup>+</sup>



\* Java Full Stack:



spring<sup>+</sup>



Java<sup>+</sup>

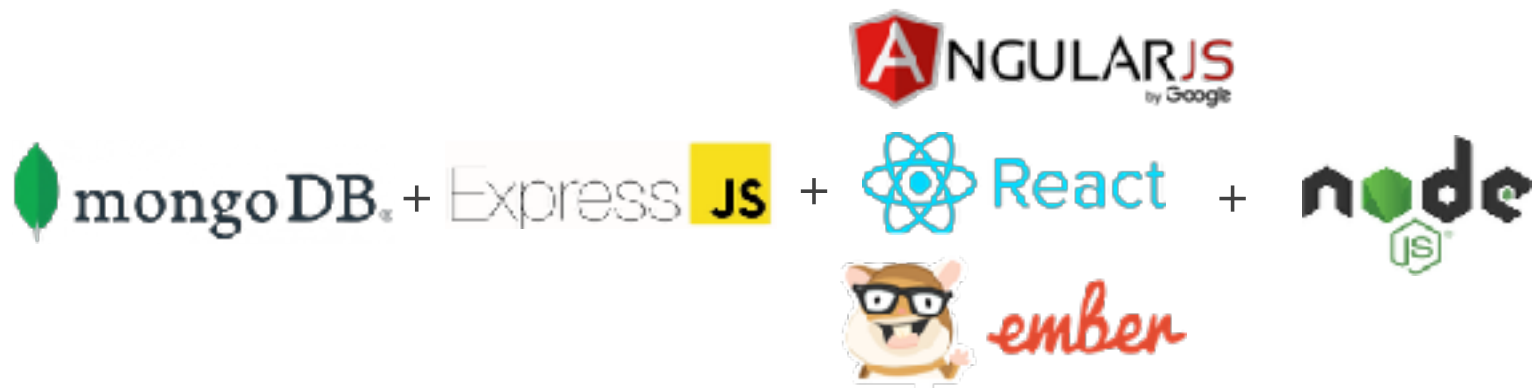
\* Ruby Stack, Django Stack, WISA, ...

# Stack: variazioni

LAMP  
WAMP  
MAMP  
WAMPP



MEAN  
MERN  
MEEN



# Stack: quale scegliere?

- \* Si tende a porsi domande di questo tipo:
  - \* Quali tecnologie ho usato in passato (esperienza, riuso del codice, etc.)?
  - \* Quali tecnologie usano i tuoi diretti concorrenti?
  - \* Con quale tecnologia preferisci lavorare?
  - \* Cosa dicono gli "esperti" online?

# Stack: quale scegliere?

- \* Le domande che invece dovresti porti:
  - \* Qual è la dimensione e la complessità del tuo progetto?
  - \* Quanto sarà estendibile l'applicazione?
  - \* Quale sarà la piattaforma finale?
  - \* Quali sono i vincoli di tempo, di budget, di personale?

# Confronti

## MEAN/MERN

- \* Unico linguaggio (js)
- \* popolare per applicazioni web moderne e ibride
- \* Supportata da grandi compagnie (e.g., Google)
- \* Migliore per le app dove la maggior parte dell'intelligenza è implementata client side
- \* Difficile da mantenere sul lungo termine dato che JS si evolve molto rapidamente
- \* Migliore per progressive web app (pwa)

## LAMP/Java+Spring

- \* Migliore per applicazioni complesse e di grandi dimensioni
- \* comunità molto grande, più matura
- \* framework molto stabili (Symfony, Laravel, Zend)
- \* Facile seguire gli standard e tenere il codice pulito ed ordinato

- \* Panoramica
- \* **Novità JS, Typescript**
- \* MVC, Single Page Applications, API design, Frameworks
- \* Angular vs Vue vs Ember vs React

# Novità JS

- \* Nuovi standard ECMAScript rilasciati annualmente
- \* Transpiling: tradurre codice nel nuovo linguaggio in JS vecchio stile
  - \* consente a tutti i software di front-end di funzionare con le "nuove" funzionalità
- \* Babel: <https://babeljs.io/en/repl>
- \* I nuovi framework usano le nuove funzionalità in modo massiccio
- \* Molto più 'object-oriented' rispetto al passato
- \* Include molte caratteristiche 'funzionali'



# "rivoluzioni" principali viste durante il corso

- \* JS Callbacks: [https://developer.mozilla.org/en-US/docs/Glossary/Callback\\_function](https://developer.mozilla.org/en-US/docs/Glossary/Callback_function)
- \* JS Closure: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>
- \* Asynchronous JS: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous>
- \* JS Promises: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise)
- \* JS Async/Await: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async\\_function](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function)

# Typescript



- \*Linguaggio sviluppato da Microsoft
- \*Sintatticamente è un sovrainsieme di JS + tipi

```
interface User {  
  name: string;  
  id: number;  
}  
  
class UserAccount {  
  name: string;  
  id: number;  
  
  constructor(name: string, id: number) {  
    this.name = name;  
    this.id = id;  
  }  
}  
  
const user: User = new UserAccount("Murphy", 1);
```

- \* Panoramica
- \* Novità JS, Typescript
- \* **MVC, Single Page Applications, API design, Frameworks**
- \* Angular vs Vue vs Ember vs React

# MVC

- \* Il pattern model-view-controller è il paradigma più usato al mondo per le applicazioni (Web) odierne
- \* E' stato usato per la prima volta con il linguaggio SmallTalk, ma reso popolare successivamente in altri ambienti
  - \* Java, J2EE, Android
  - \* Objective-C (iOS)
  - \* Ruby on Rails
  - \* ...

# AngularJS



- \* Estende HTML con **nuovi attributi**
- \* Pensato appositamente per applicazioni costituite da una sola pagina (**SPA – single page application**)
  - \* Velocizza lo sviluppo grazie ad un'estensione dinamica di HTML
  - \* Inserisce la logica MVC direttamente nel front end
- \* Nuova terminologia: directive, expression, filter, module, controller
- \* Nuove funzioni per: DOM, form, input, validazione, richieste http, etc.
- \* Creato da Miško Hevery nel 2009 (sviluppatore Google)

# AngularJS

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
```

```
<div ng-app="">
```

```
<p>Input something in the input box:</p>
```

```
<p>Name: <input type="text" ng-model="name"></p>
```

```
<p ng-bind="name"></p>
```

```
</div>
```

Input something in the input box:

Name:

Giancarlo

[http://www.w3schools.com/angular/tryit.asp?filename=try\\_ng\\_intro](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_intro)

# Direttive ed espressioni

## \* Direttive

- \* `ng-app`: dice ad AngularJS che l'elemento `<div>` è il "proprietario" dell'applicazione
- \* `ng-model`: lega il valore del campo di input alla variabile *name*
- \* `ng-bind`: lega l'innerHTML dell'elemento `<p>` alla variabile dell'applicazione *name*

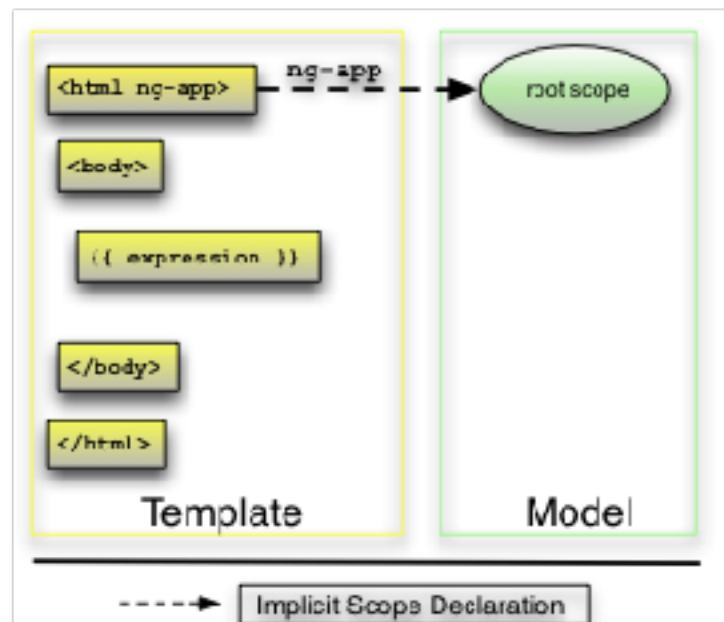
## \* Espressioni

- \* Simili alle espressioni JS
- \* Es. `{{ 5 + 5 }}` or `{{ firstName + " " + lastName }}`

```
<div ng-app="">
  <p>Name: <input type="text" ng-model="name"></p>
  <p>{{ name }}</p>
</div>
```

# Riferimenti





- \* Tutorial w3schools: <http://www.w3schools.com/angular/default.asp>
- \* Tutorial ufficiale: [https://docs.angularjs.org/tutorial/step\\_00](https://docs.angularjs.org/tutorial/step_00)








- \* Panoramica
- \* Novità JS, Typescript
- \* MVC, Single Page Applications, API design, Frameworks
- \* **Angular vs Vue vs Ember vs React**

# Un confronto

Attributes				
	ReactJS	AngularJS	EmberJS	VueJS
Type	Library to build UI	A framework	A framework	A framework
Why Choose?	If you like all things JavaScript	If you love TypeScript	If you like all things JavaScript	Easy JavaScript & HTML
Backed By	Facebook	Google	Community	Community
Initial Release	March 2013	September 2016	December 2011	February 2014
Performance	Excellent	Fine	Slow in Rendering	Excellent
Data Binding	One way	Two way	Two way	Two way

Attributes				
	ReactJS	AngularJS	EmberJS	VueJS
Widely used for	Modern web & native app development (iOS and Android)	Large-scale and feature-rich app development	Web app development	Web and SPAs development
Learning Curve	Easier than Angular	A steep learning curve	A steep learning curve	Small learning curve
Model	Based on virtual DOM (flux)	Based on MVC	Based on MVVM	Based on virtual DOM (flux)
Community Support	Facebook developers community	A large community of developers & supporters	A small community of developers	Open-source project sponsored through crowd-sourcing
Development Speed	Normal	Normal	Fast	Normal
Testing	Your Choice	Jasmine & karma	Q-unite	Karma & Mocha, Jest

Attributes	 ReactJS	 AngularJS	 EmberJS	 VueJS
	ReactJS	AngularJS	EmberJS	VueJS
Documentation	Good	Good	Good	Good
Popularity	Widely popular among developers	Widely popular among developers	More than 20,000 stars added on Github	More than 40,000 stars added on Github during the year
Best attraction	Virtual DOM	Oldest Framework	Principals First	Combination of React & Angular
Companies Using	Used by Facebook, Uber, Netflix, Twitter, Reddit, Paypal, Walmart, & others	Used by Google, Forbes, Wix, & weather.com	Netflix, LinkedIn, Sony & many more.	Used by Alibaba, Baidu, GitLab, & others

# Altri puntatori

- \* <https://developers.google.com/>
- \* <https://www.youtube.com/channel/UCnUYZLuoy1rq1aVMwx4aTzw>
- \* <https://webpack.js.org/>
- \* <https://mochajs.org/>
- \* ...

# Il futuro?

