



Tecnologie e applicazioni web

Introduzione

Filippo Bergamasco (filippo.bergamasco@unive.it)

<http://www.dais.unive.it/~bergamasco/>

DAIS - Università Ca' Foscari di Venezia

Anno accademico: 2017/2018

Informazioni sul corso

Pagina web del corso:

<http://www.unive.it/data/insegnamento/212562>

Pagina web “non ufficiale”

<http://www.dsi.unive.it/~bergamasco/webtech.html>

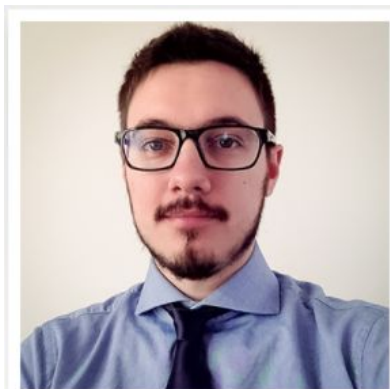
Informazioni sul corso

- 48 Ore - 24 Lezioni (frontali + lab)
- Orario:
 - Lunedì: 14:00 - 15:30
 - Martedì: 8:45 - 10:15
- Codice insegnamento: **CT0142**

Docente

- Filippo Bergamasco

<http://www.dais.unive.it/~bergamasco>



- Assegnista di ricerca
- Gruppo di ricerca KiiS
- Mi occupo di visione artificiale:
 - Ricostruzione 3D / Pattern recognition / Fotogrammetria

Esame

- L'esame prevede lo **sviluppo di un progetto** software che utilizzi tecnologie e metodi sviluppati durante il corso.
- Il progetto va consegnato congiuntamente ad una relazione (individuale) che sarà poi discussa mediante prova orale
- Consegna del progetto/relazione almeno 1 settimana prima dell'orale

Esame

- Il contenuto del progetto, e l'eventuale suddivisione in gruppi, saranno discussi prossimamente
- La consegna avverrà esclusivamente attraverso la piattaforma moodle:

<https://moodle.unive.it/course/view.php?id=1593>

Pagina moodle

Importante:

Registrarsi fin da subito utilizzando la chiave: **WAT**

La registrazione alla pagina moodle del corso è fondamentale per poter consegnare il progetto e la relazione.

Prerequisiti al corso

Non vi sono propedeuticità formali per poter sostenere questo corso

Requisiti consigliati:

- Conoscenza del linguaggio Javascript
- Conoscenza (di base) di HTML/CSS
- Reti di calcolatori

Prerequisiti al corso

Non vi sono propedeuticità formali per poter sostenere questo corso

Requisiti consigliati:

- Saper programmare
- Capacità di sintesi e progettazione
- Curiosità (capire il funzionamento delle cose)

A cosa serve questo corso?

Il corso mira a fornire una panoramica dettagliata sulle più recenti tecnologie legate al web

Lato teorico:

- Sviluppo API / Sicurezza / Protocolli / Patterns

Lato pratico:

- Tecnologie nate per il web sono oggi pervasive in ogni ambito, compresi Desktop e Mobile

A cosa serve questo corso?

In particolare, Il corso si focalizza sulle tecnologie legate all'ecosistema Javascript

- Un unico linguaggio per lato client e server
- Caratteristiche che lo rendono particolarmente “friendly” per un approccio event-driven
- Esistenza di interpreti javascript jit ottimizzati e molto veloci (Google V8)
- Base su cui sono stati sviluppati altri “dialetti” (TypeScript, Coffee-Script, etc)

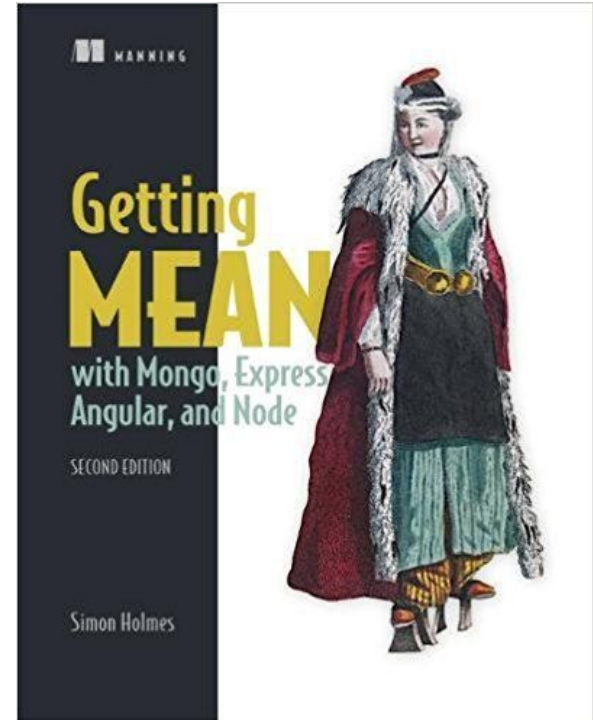
A cosa serve questo corso?

“I came to appreciate that there is a real art to gluing together applications made from different technologies. It is a skill in itself; just knowing the technologies and what they can do is only part of the challenge.”

Simon Holmes - Getting MEAN with Mongo,
Express, Angular, and Node

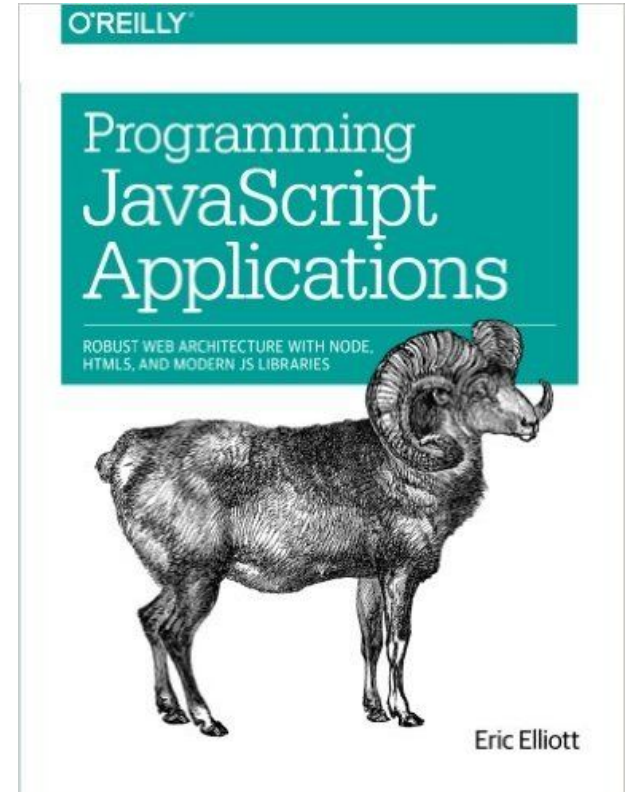
Testi di riferimento

Simon Holmes,
"Getting MEAN with Mongo,
Express, Angular, and Node",
Manning Publications, 2015.
ISBN-10: 1617292036



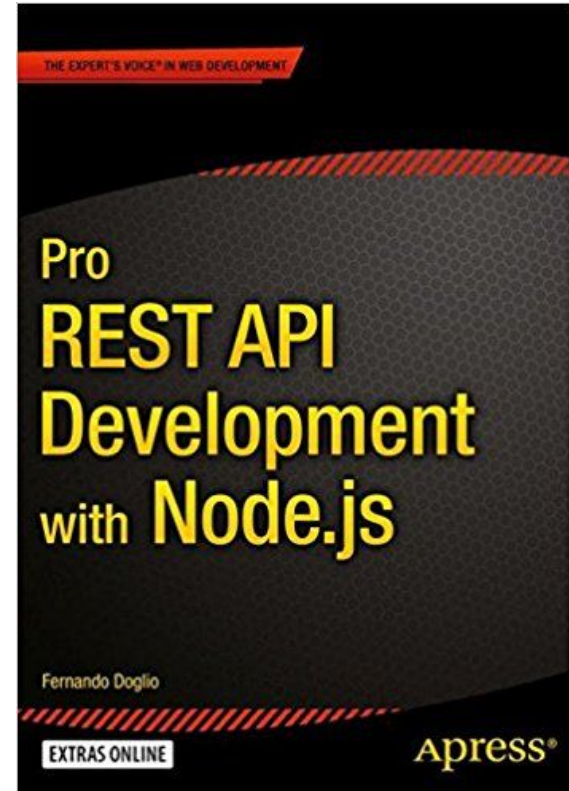
Testi di riferimento

Eric Elliott,
"Programming JavaScript
Applications: Robust Web
Architecture with Node, HTML5,
and Modern JS Libraries",
O'Reilly Media, 2014.
ISBN-10: 1491950293



Testi di riferimento

Fernando Doglio,
"Pro REST API Development
with Node.js",
Apress, 2015.
ISBN-10: 1484209184



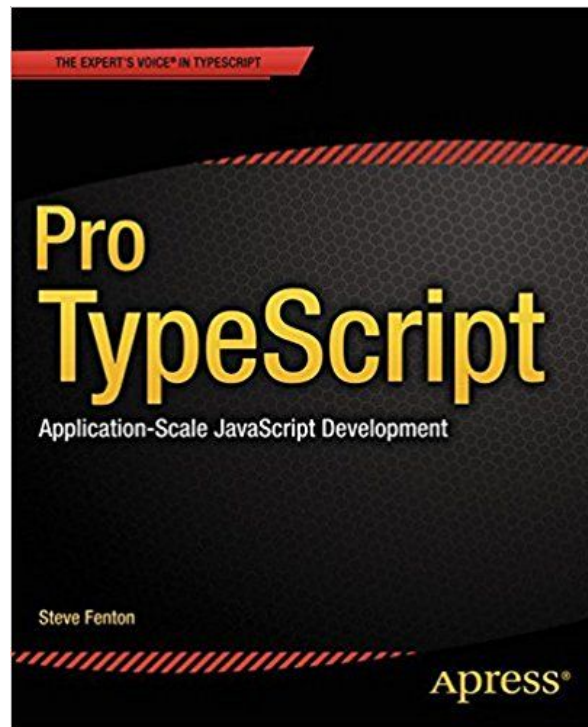
Testi di riferimento

Steve Fenton,

"Pro Typescript",

Apress, 2014.

ISBN 978-1-4302-6790-4



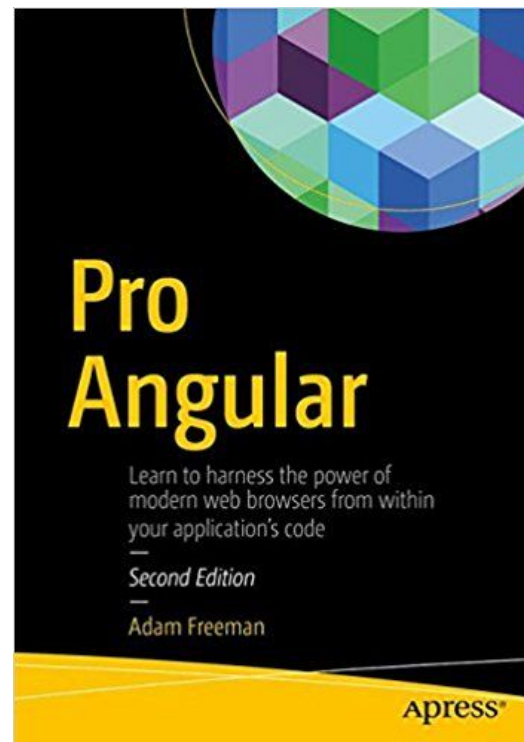
Testi di riferimento

Adam Freeman,

"Pro Angular",

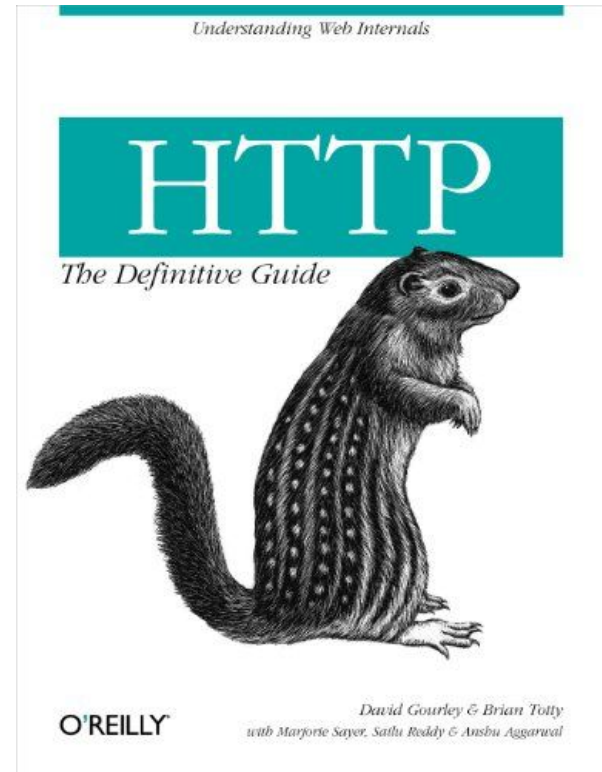
Apress 2017.

ISBN 978-1-4842-2307-9



Vivamente consigliato...

David Gourley et al.,
"HTTP: The Definitive Guide: The
Definitive Guide",
O'Reilly Media.
ISBN 1565925092



Syllabus: Part 1

- The HTTP protocol
- Cookies
- Authentication: Basic and Digest Access Authentication, JSON web token
- Websocket
- REST api
- Message-oriented middleware
- Javascript, Coffee-script and TypeScript

Syllabus: Part 2

Node.js server-side Javascript runtime

- Storia e panoramica sull'ambiente
- Il sistema di pacchetti npm
- Modello di IO Asincrono
- Event-driven programming
- API per interfacciarsi con il sistema operativo
- HTTP APIs

Syllabus: Part 3

NOSQL storage

- MongoDB
- REDIS in-memory db/cache/message broker



Syllabus: Part 4

Stack middleware

- Express js



Syllabus: Part 5

Web front-end

- Bootstrap
- Angular



Syllabus: Part 6

Mobile hybrid and progressive apps

- jQuery Mobile
- Apache Cordova
- IONIC framework



APACHE
CORDOVA™



Syllabus: Part 7

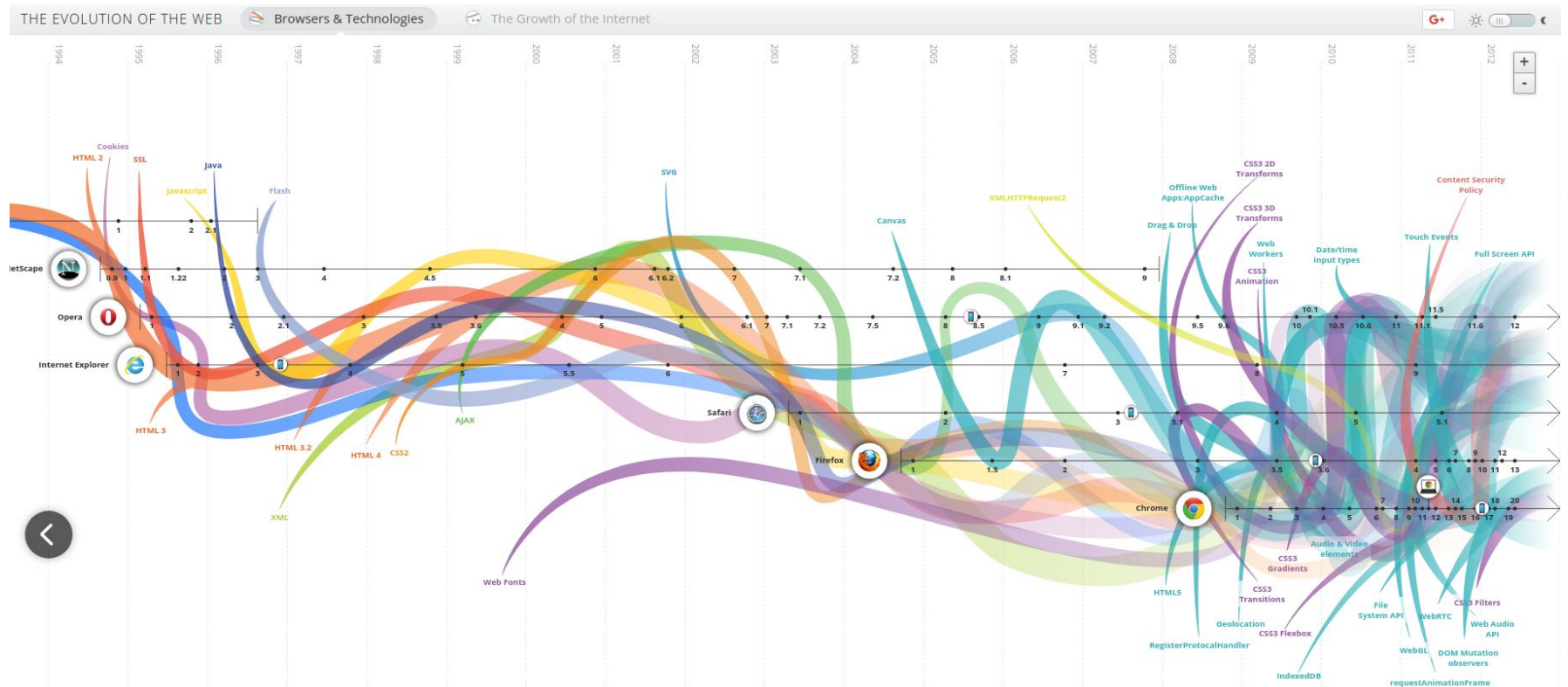
Cross-platform web-based native applications

- Electron framework



Evoluzione del web

<http://www.evolutionoftheweb.com/#/evolution/day>



Evoluzione del web

1989: Tim Berners Lee

- Ricercatore presso il CERN
- Delinea l'idea di avere una piattaforma software che possa facilitare la collaborazione tra i ricercatori in diverse parti del mondo

Evoluzione del web

1990: Viene definito l'HTTP (Tim Berners Lee et al.)

- Hyper-text transfer protocol
- Protocollo di tipo **request-response** pensato originariamente per l'interscambio di ipertesti
- Fondamentale il concetto di **risorse** identificate in modo univoco sulla rete da una stringa detta Uniform Resource Locator (**URL**)

Evoluzione del web

1990: Viene inventato l'HTML (Tim Berners Lee et al.)

- Basato su un altro linguaggio di markup: Standard Generalized Markup Language (SGML)
- Diventa uno dei mattoni fondamentali per lo sviluppo del web come lo conosciamo oggi.
- Tuttora alla base della sua infrastruttura

Evoluzione del web

1990: Viene inventato l'HTML (Tim Berners Lee et al.)

- E' un linguaggio di markup che viene interpretato da software, detti web browsers, sottoforma di documenti contenenti testo, immagini, suoni, etc.
- Linguaggio di tipo dichiarativo: permette soltanto di definire la struttura dei documenti. Non permette interattività

Evoluzione del web

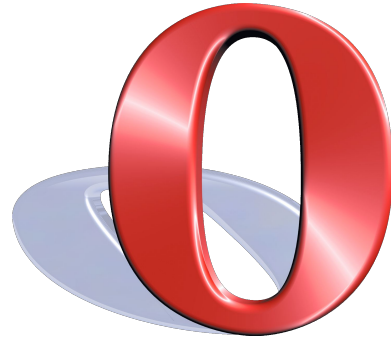
1993: Viene rilasciato MOSAIC

- Rivoluziona il concetto di web browser permettendo la visualizzazione di testo e immagini nella stessa pagina
- Ha permesso la diffusione del web come lo conosciamo oggi



Evoluzione del web

1994-1996: Nascono diversi browser



Evoluzione del web

1995: Nasce Javascript

- Incluso per la prima volta in Netscape Navigator 2
- Sintassi ispirata a “Java”
- Idea (Marc Andreessen): HTML necessitava di un linguaggio semplice da utilizzare e che permettesse agli sviluppatori web di modificare dinamicamente il contenuto delle pagine web
- Il codice Javascript veniva incluso direttamente nelle pagine (tag `<script>`)

Evoluzione del web

1996: Nasce il concetto di CSS

- Linguaggio nato per definire il modo in cui vanno visualizzati i vari elementi di un documento scritto in un linguaggio di markup.
- IDEA: Separare presentazione e contenuto
 - Più flessibilità
 - Più accessibilità delle risorse
 - Ridurre la complessità e le ripetizioni per il contenuto strutturale delle pagine web

Evoluzione del web



1996-2000: Le aziende iniziano a dare sempre più importanza alla presentazione dei loro contenuti.

- CSS e Javascript diventano tecnologie fondamentali per gli sviluppatori web per ottenere interfacce sempre più sofisticate
- **Contenuti non più statici ma generati dinamicamente**
- Nasce la figura del front-end developer

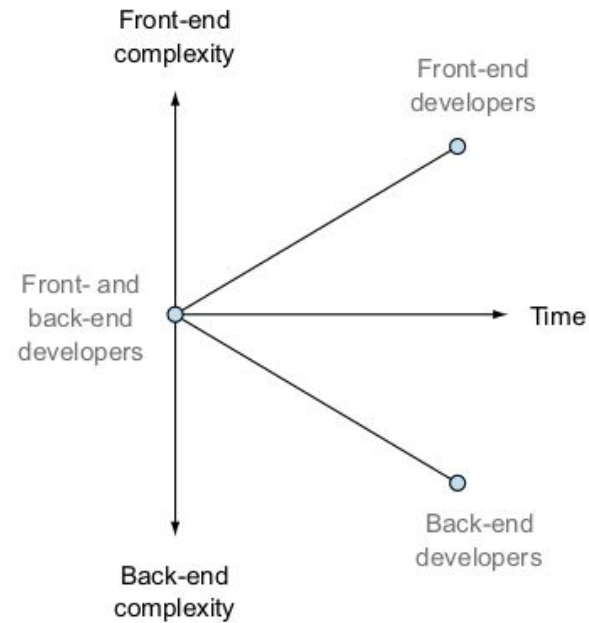
Evoluzione del web

Back-end developers:

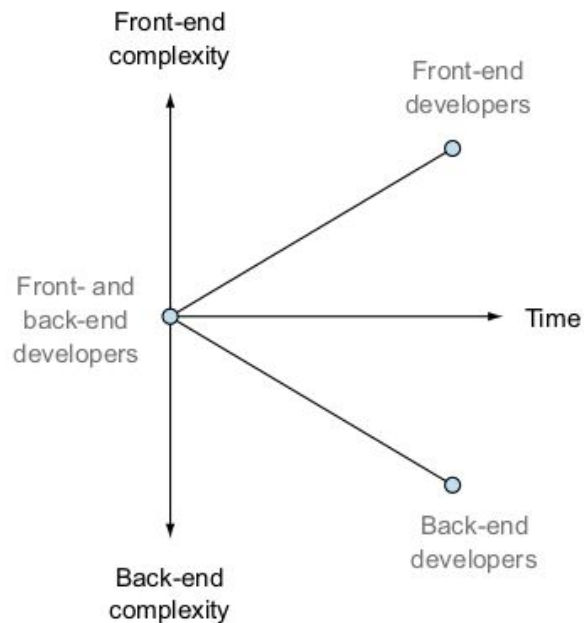
Focalizzati sulle meccaniche “dietro le quinte” di un sito web (dati, contenuti, sicurezza, struttura)

Front-end developers:

Focalizzati sulla presentazione dei contenuti e la user experience



Evoluzione del web



- In passato uno sviluppatore web doveva scegliere a-priori l'ambito in cui specializzarsi
- Front e backend comprendevano spesso tecnologie molto diverse

Evoluzione del web

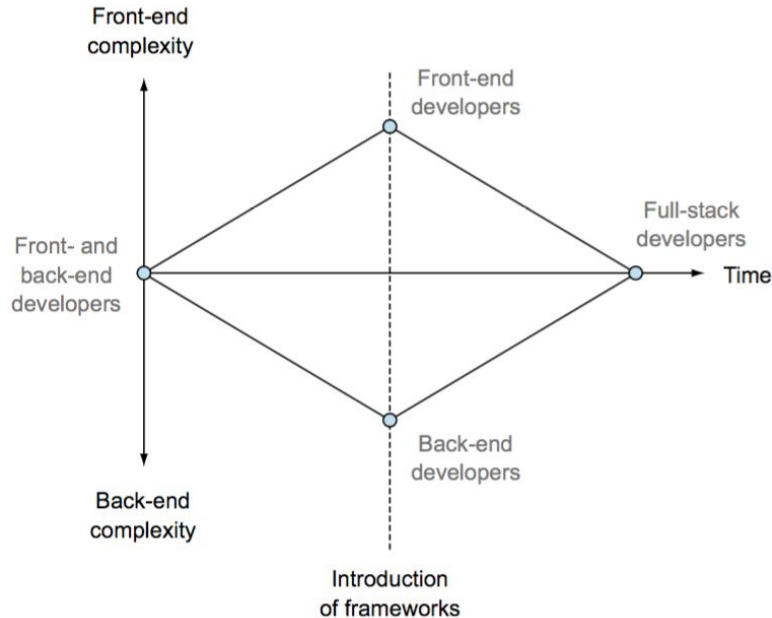
A partire dai primi anni 2000 iniziano a svilupparsi frameworks e librerie sia backend che frontend

- Backend: PHP, Ruby on Rails, Java/JSP
- Frontend: jQuery, Dojo

Un buon framework è in grado di astrarre molte delle complessità dello sviluppo di applicazioni web:

- Sviluppo più rapido e meno expertise richiesta

Evoluzione del web



- Questa tendenza all'astrazione e alla semplificazione attraverso framework ha portato via via al concetto di “sviluppatori full-stack”
- Sviluppo simultaneo (spesso con stesse tecnologie) di front e backend

Frameworks full-stack

I web-frameworks moderni forniscono oggi l'interessante possibilità di non doversi necessariamente specializzare tra back-end o front-end

- Si può seguire lo sviluppo completo di un'app web in tutte le sue parti
- Produttività, possibilità di lavoro freelance

Frameworks full-stack

Vantaggi:

- Si ha il controllo di tutte le parti di una web application. Se le stesse persone sviluppano entrambi gli aspetti (back e front end) in genere queste possono meglio interoperare
- Si utilizzano le stesse tecnologie per molteplici piattaforme: web/mobile/desktop
- Estrema velocità di implementazione di un'idea

Frameworks full-stack

Quale framework dovrei scegliere?

.... Dipende, come sempre, dal contesto.

Esiste però oggi la possibilità di usare uno stesso linguaggio per lo sviluppo di ogni aspetto di una web application: Javascript

Impareremo in questo corso a sviluppare con un insieme di tecnologie definite dall'acronimo "MEAN"

Evoluzione del web

2005 - Jesse James Garrett pubblica l'articolo "Ajax: A New Approach to Web Applications"

- Il termine Ajax viene coniato per descrivere una serie di tecnologie per lo sviluppo di applicazioni web che **comunicano in background (in modo asincrono)** con il server senza interferire con lo stato della pagina
- HTML/DOM/JSON or XML/Javascript

Evoluzione del web

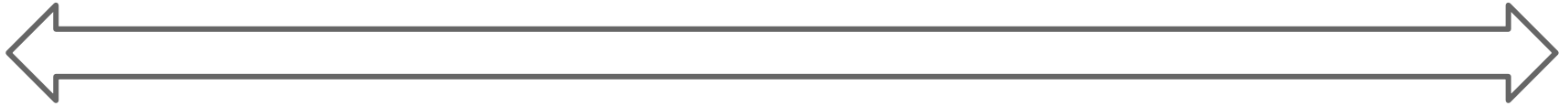
Trend: moving the application code forward in the stack

- AJAX permette di spostare la logica dell'applicazione dal server al client (browser web)

Vantaggio principale:

- Riduce fortemente il carico di lavoro sul server, e di conseguenza i costi

Server o client side?



Server side

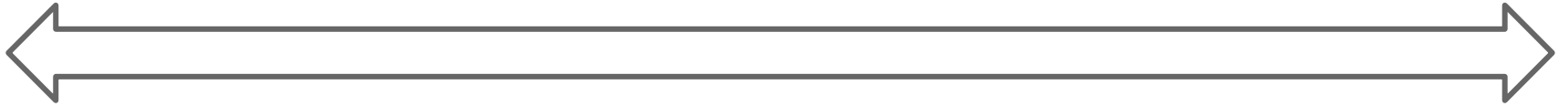
Il server gestisce la ogni aspetto dell'applicazione:
Generazione contenuti, dati,
validazione form utente, etc

Client side

(Single-page app)

Il client gestisce tutti gli aspetti sia di presentazione che di business logic. L'unica comunicazione con il server avviene per scambiare dati (JSON)

Single page application



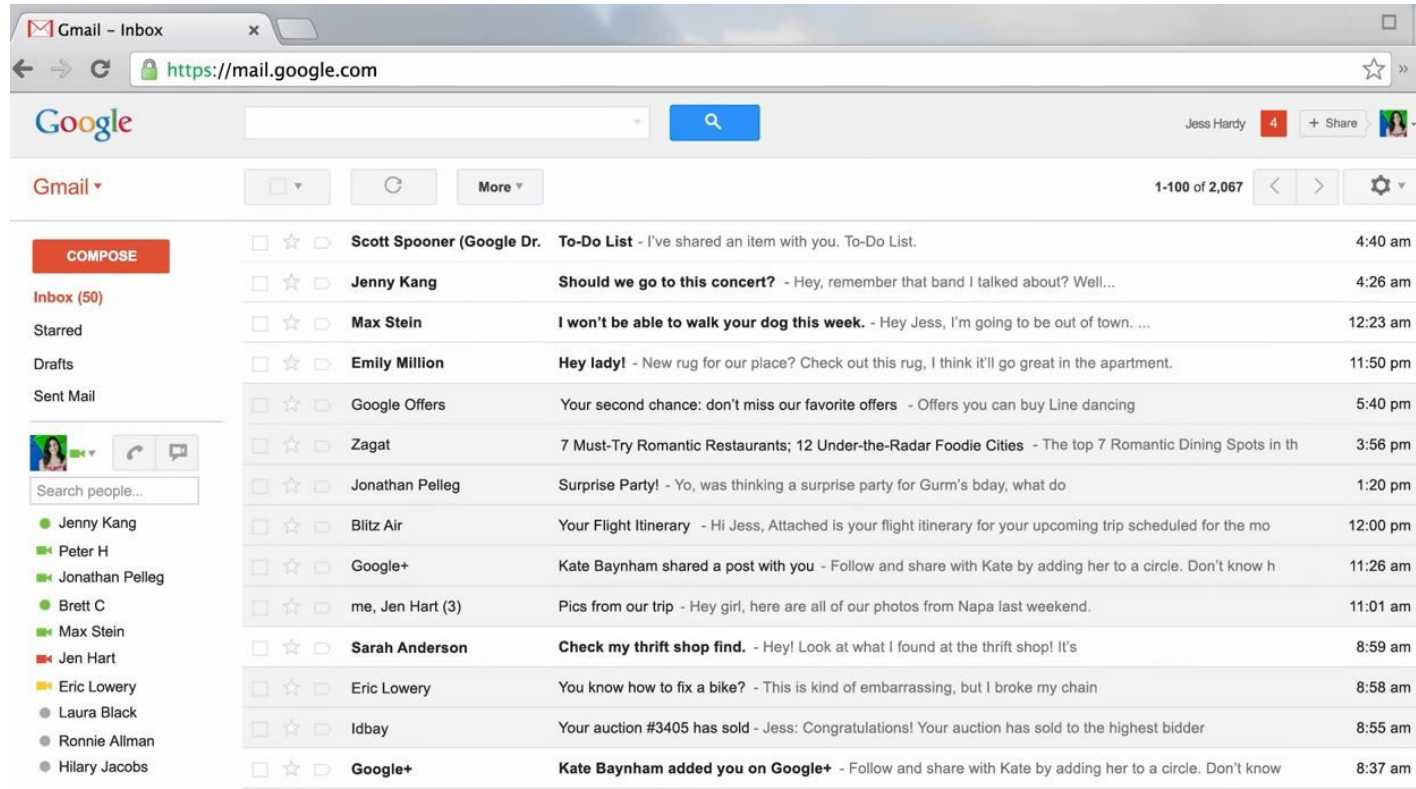
Server semplice:

Pure data API o web service

Client complesso

Necessità di interi framework per gestire la business logic dell'applicazione e l'interattività con l'utente

Single page application



Evoluzione del web

2008 - Google pubblica Chrome

- Alta aderenza agli standard (ACID tests)
- Approccio minimalista
- Sicuro, integrato con i servizi Google

Implementa il motore Javascript (open-source) V8

- Jit compiler, ottimizzato dinamicamente a runtime su euristiche in fase di esecuzione, inline caching



Web 1.0 vs 2.0

2004 - Tim O'Reilly popolarizza il concetto di web 2.0

- Non si riferisce a nessuno standard specifico ma al modo in cui le pagine sono sviluppate e fruite

Web 1.0

- Pagine principalmente statiche
- Il contenuto creato da poche persone, utenti semplici "consumatori"

Web 2.0

- Pagine dinamiche
- Il contenuto generato principalmente dagli utenti in comunità virtuali
- Social media

Web 3.0?

Web semantico

- Il contenuto è organizzato in ontologie per permettere la comprensione semantica dei contenuti sia dagli umani che dalle macchine
- Ancora oggetto di attività di ricerca più che standard de-facto...