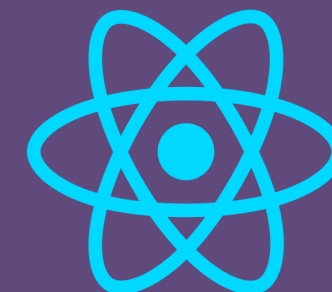


FRONTEND ROADMAP

BEGINNING DILEMMA





Bacaro
Tech
CODE AND FUN

PRESENTAZIONI



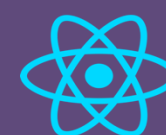
Michele Scarpa
Web developer



@michele_scarpa



michele-scarpa-90-arco



BACAROTECH



Giorgio Basile
FE developer



Michele Scarpa
Full stack developer

...VI PRESENTIAMO **BACAROTECH**

Bacaro Tech è un'iniziativa che ha il compito di ricreare quell'atmosfera gioiosa e di gruppo, tipica del bacaro veneziano, nel mondo dell'informatica, attraverso la divulgazione sui social, eventi e workshop



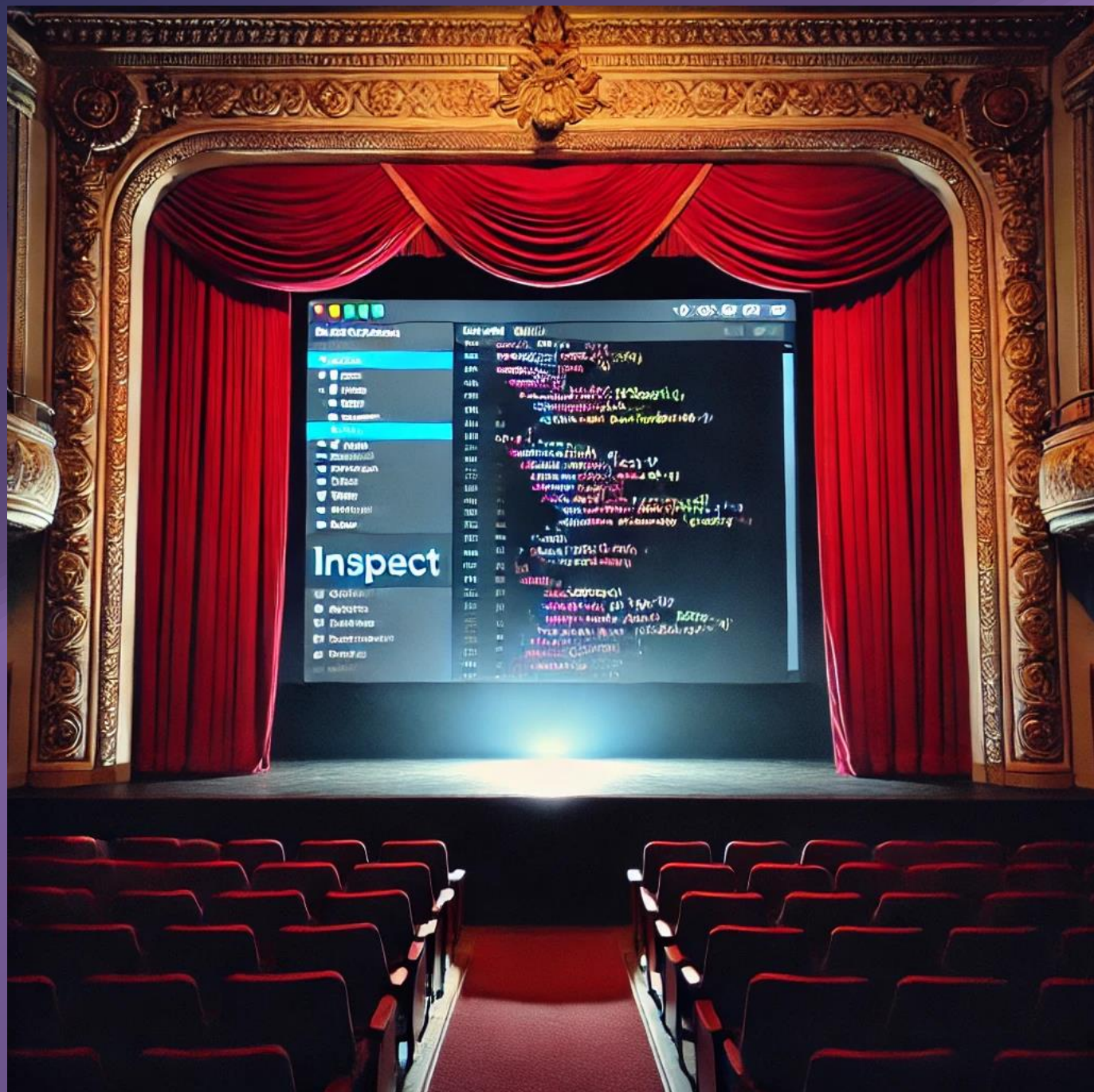
CHE COS'È BACAROTECH



@BACAROTECHOFFICIAL

**ORA INIZIAMO CON
LA ROADMAP**

CURIOSITY = INSPECTOR



Search

chrome-extension://nhdogimejiglipccpnnanhbledajbpd/build/detector-exec.js

Preserve log

Disable cache

No throttling

Invert

Hide data URLs

Hide extension URLs

Blocked requests

3rd-party requests

All

Fetch/XHR

Doc

CSS

JS

Font

Img

Media

Manifest

WS

Wasm

Other

Blocked response cookies

2000 ms

4000 ms

6000 ms

8000 ms

10000 ms

12000 ms

14000 ms

16000 ms

18000 ms

20000 ms

22000 ms

24000 ms

Name	Status	Type	Initiator	Size	Time
menu	200	xhr	10904429849:2	598 B	46 ...
detector-exec.js	200	script	detector.js:1	1.2 kB	78 ...
spFormElementPrototypeEx.js	200	script	spAutofillCore.js:3283	2.0 kB	15 ...
manifest.webmanifest	200	manifest	Other	(disk cache)	1 ms
favicon.png	200	png	Other	(disk cache)	2 ms
favicon	200	svg+xml	Other	983 B	50 ...
track	200	xhr	InstrumentHooks.js:97	201 B	20...
collect?v=2&tid=G-9RSMQ9D5LK>m=45je4al0h2v...	204	fetch	InstrumentHooks.js:97	48 B	44 ...
track	200	xhr	InstrumentHooks.js:97	120 B	40 ...

29 requests

132 kB transferred

3.2 MB resources

Finish: 21.74 s

DOMContentLoaded: 989 ms

Load: 1.69 s

Console

What's new

Network conditions

Sources

spAutofillCore.js

top

2022

2023

2024

2025

2026

2027

2028

2029

2030

2031

2032

2033

2034

2035

2036

// StopMonitoring()
StopMonitoring: function()
{
 try
 {
 this._StopLoadDelayTimer();
 this._StopDynamicallyMonitorTimer();
 if (this.MutationObserver)
 {
 if (this.Log.Info)
 spLog.logMessage('TspDOMChangesMonitor.StopMonitoring() Disconnect and c
 this.MutationObserver.disconnect();
 this.MutationObserver = null;
 this.AllowPasswordElementMonitor = false;
 }
 }
}

Watch

Breakpoints

Pause on uncaught exceptions

Pause on caught exceptions

Scope

Not paused

Call Stack

Not paused

XHR/fetch Breakpoints

DOM Breakpoints



ROADMAP.sh

ROLE- Based

Role-based Roadmaps		
Frontend	Backend	DevOps
Full Stack	AI Engineer New	Data Analyst
AI and Data Scientist	Android	iOS
PostgreSQL	Blockchain	QA
Software Architect	Cyber Security	UX Design
Game Developer	Technical Writer	MLOps
Product Manager	Developer Relations	+ Create your own Roadmap

SKILL- Based

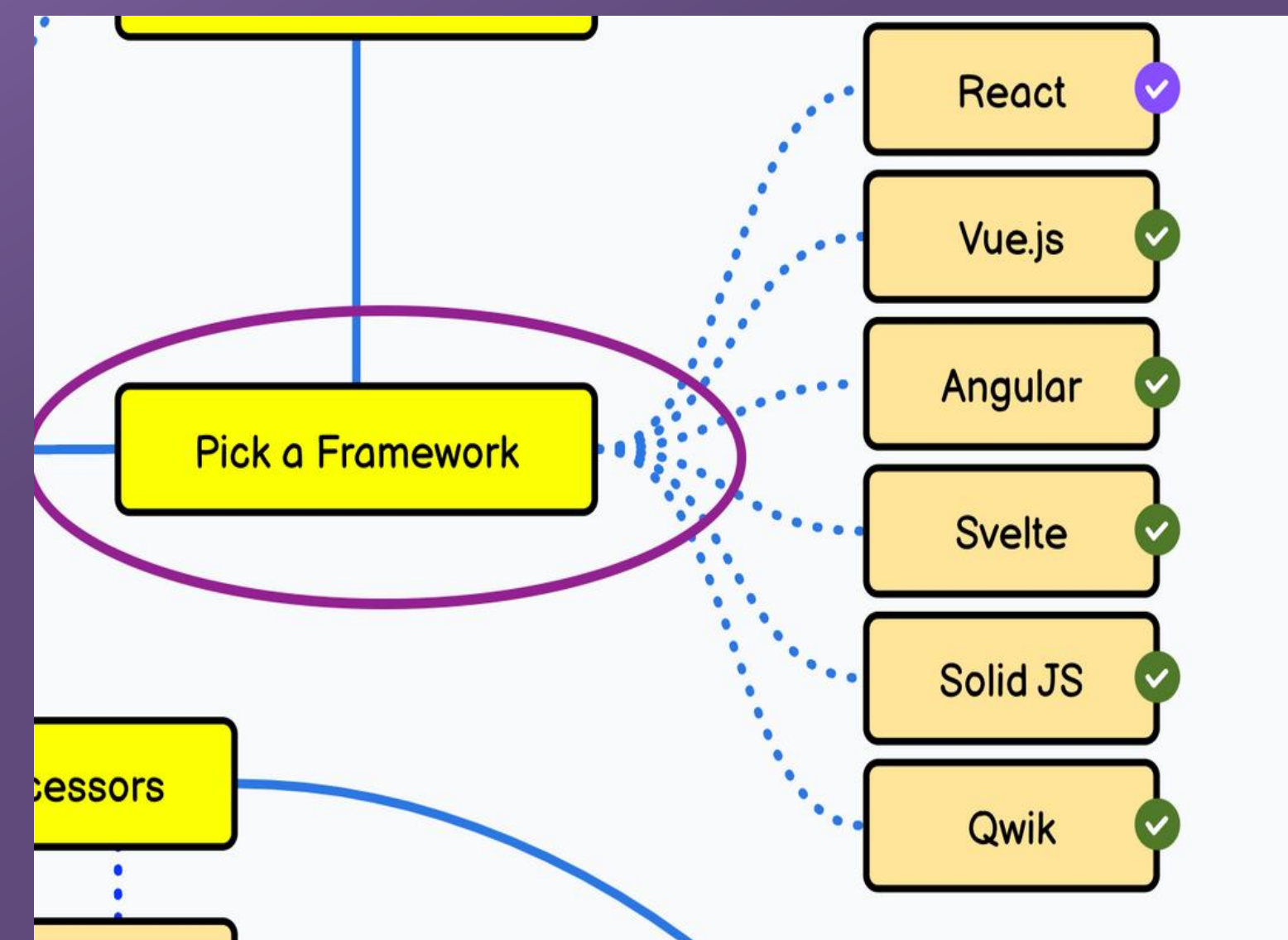
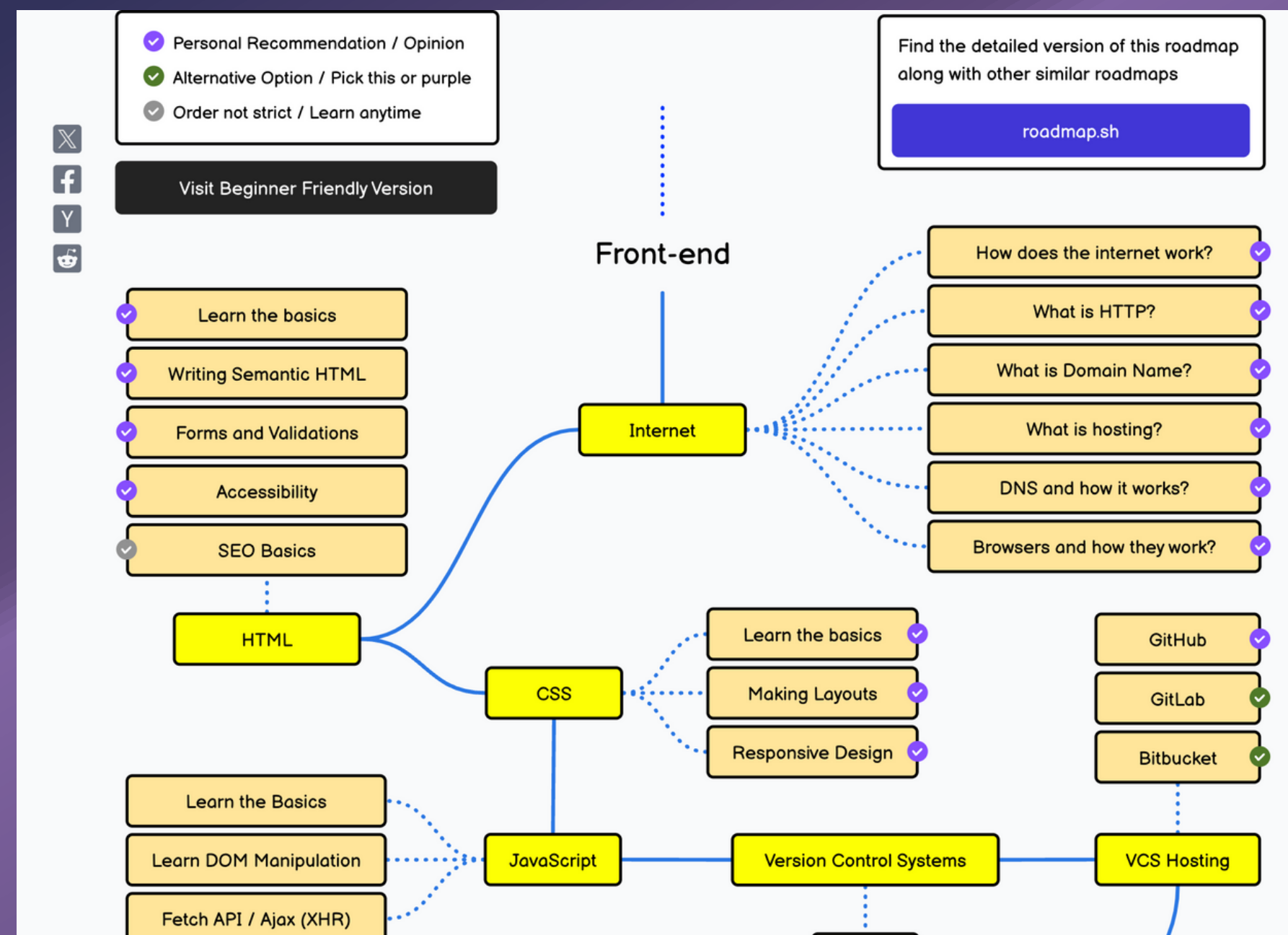
Skill-based Roadmaps		
Computer Science	React	Vue
Angular	JavaScript	Node.js
TypeScript	Python	SQL
System Design	API Design	ASP.NET Core
Java	C++	Flutter
Spring Boot	Go Roadmap	Rust
GraphQL	Design and Architecture	Design System

Developer Roadmaps





Frontend Roadmap

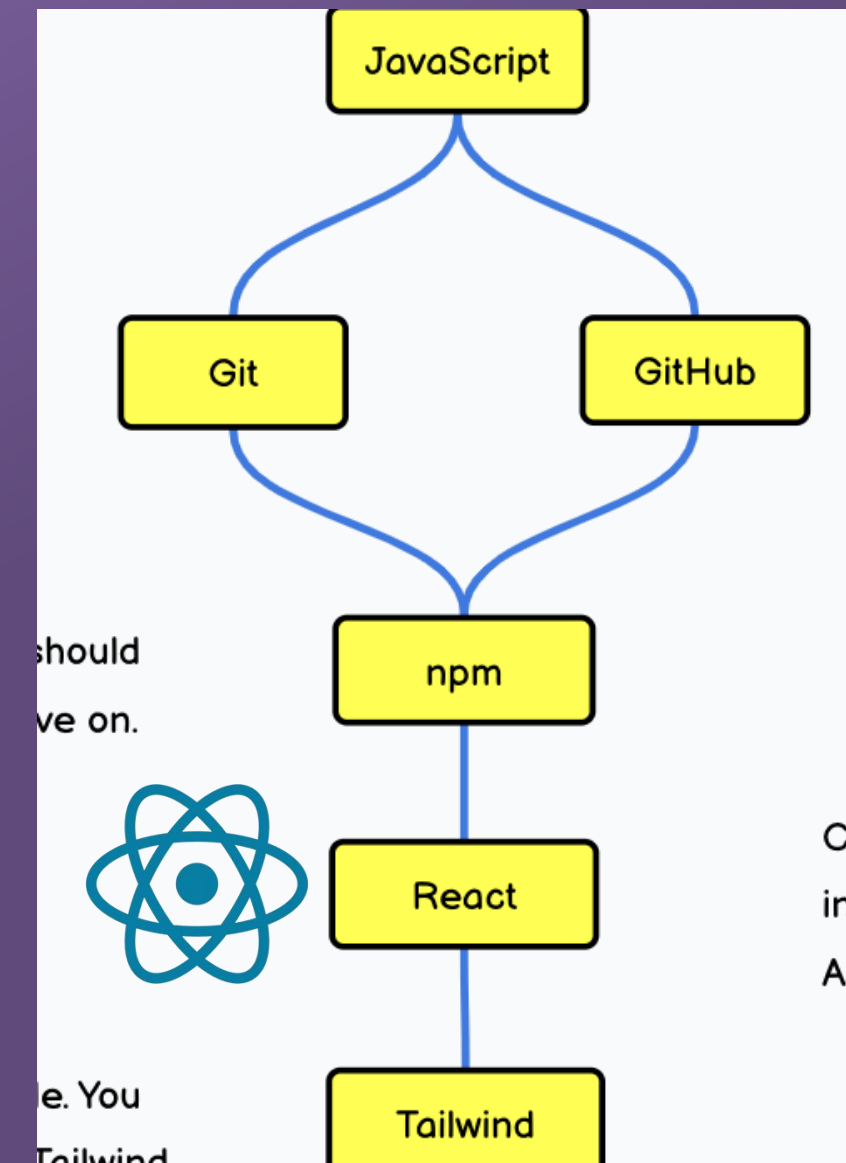




PICK A FRAMEWORK

- ✓ Personal Recommendation / Opinion
- ✓ Alternative Option / Pick this or purple
- ✓ Order not strict / Learn anytime

Visit Beginner Friendly Version



Ingredienti **per creare la nostra mappa**

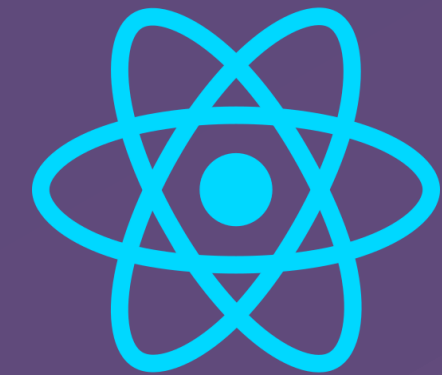
Lista di tecnologie

Confrontare

Provare



Quali alternative ci sono?



...Perché ci sono così tante alternative



a long time ago in a WEB far far away...

 1995 JavaScript

2006 JQuery

2007 Mobile

 2010 Angular

2012 Typescript

 2013 React

 2014 Vue

2015 VSCode

 2016 Svelte

Ogni tecnologia

nasce per

risolvere

determinati

problemi



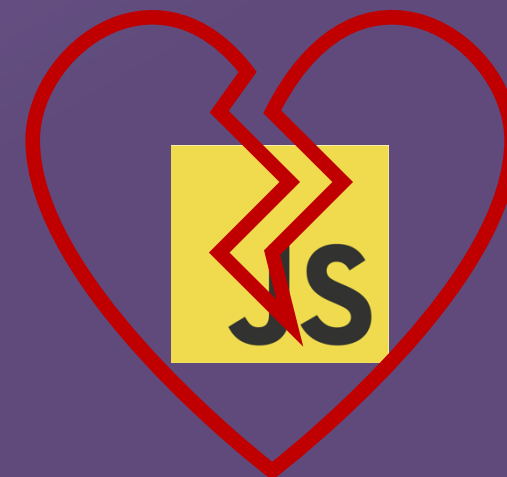
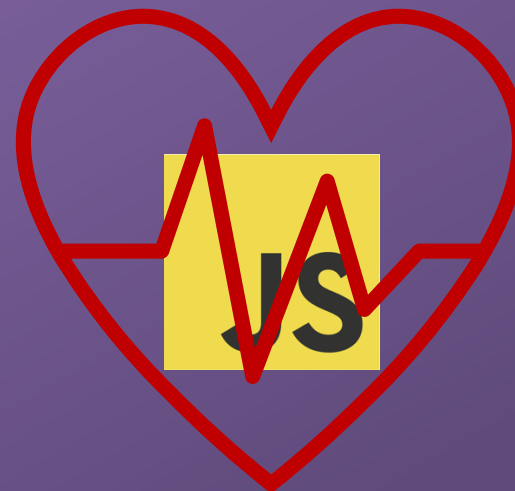
Bacaro
Tech

CODE AND FUN

Documentary



JAVASCRIPT





2006 \$



John Resig ha creato jQuery per semplificare l'interazione con il DOM

Compatibilità Cross-Browser

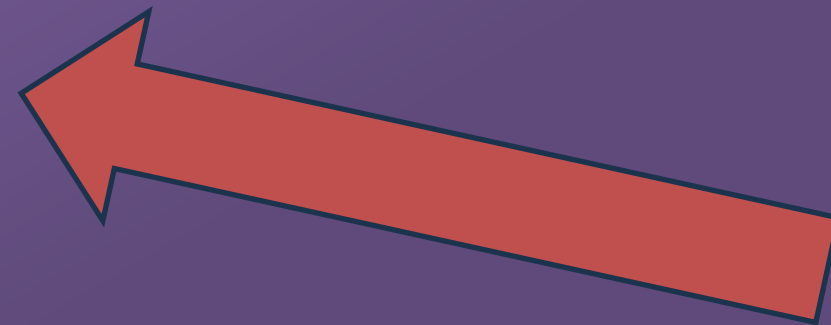
Riduzione del Codice Necessario

AJAX

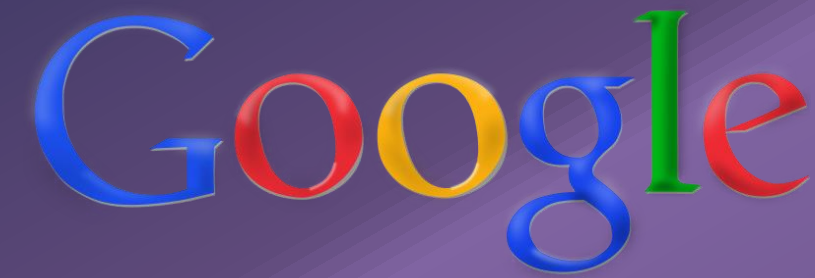
Animazioni Semplificate

Importanza storica

Progetti legacy



**Un Ponte tra il Passato e il
Presente**



ANGULAR

2010 Misko Hevery di Google Voleva un modo per creare applicazioni dinamiche senza dover aggiornare manualmente il DOM

Data Binding Bidirezionale

Struttura MVC (Model-View-Controller)

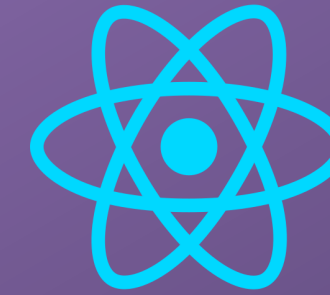
Dependency Injection

Template Dichiarativi



Bacaro
Tech

CODE AND FUN



React

2013 - Facebook rilascia React. Libreria per gestire le esigenze di interfacce utente dinamiche e ad alte prestazioni nelle sue applicazioni.

Virtual DOM

Componenti Riutilizzabili

One-Way Data Binding

JSX



2014 Evan You, crea un framework più leggero e flessibile, combinando le migliori idee di AngularJS e React.

Facilità di Apprendimento

Flessibilità

Performance Elevate

Template e Componenti Chiari





2016 - Rich Harris, dal mondo dell'editoria, crea un framework per colmare gran parte delle inefficienze dei suoi predecessori.

Compilazione a Build Time

Eliminazione del Runtime Overhead

Reattività Semplificata

Sintassi Pulita

Runtime



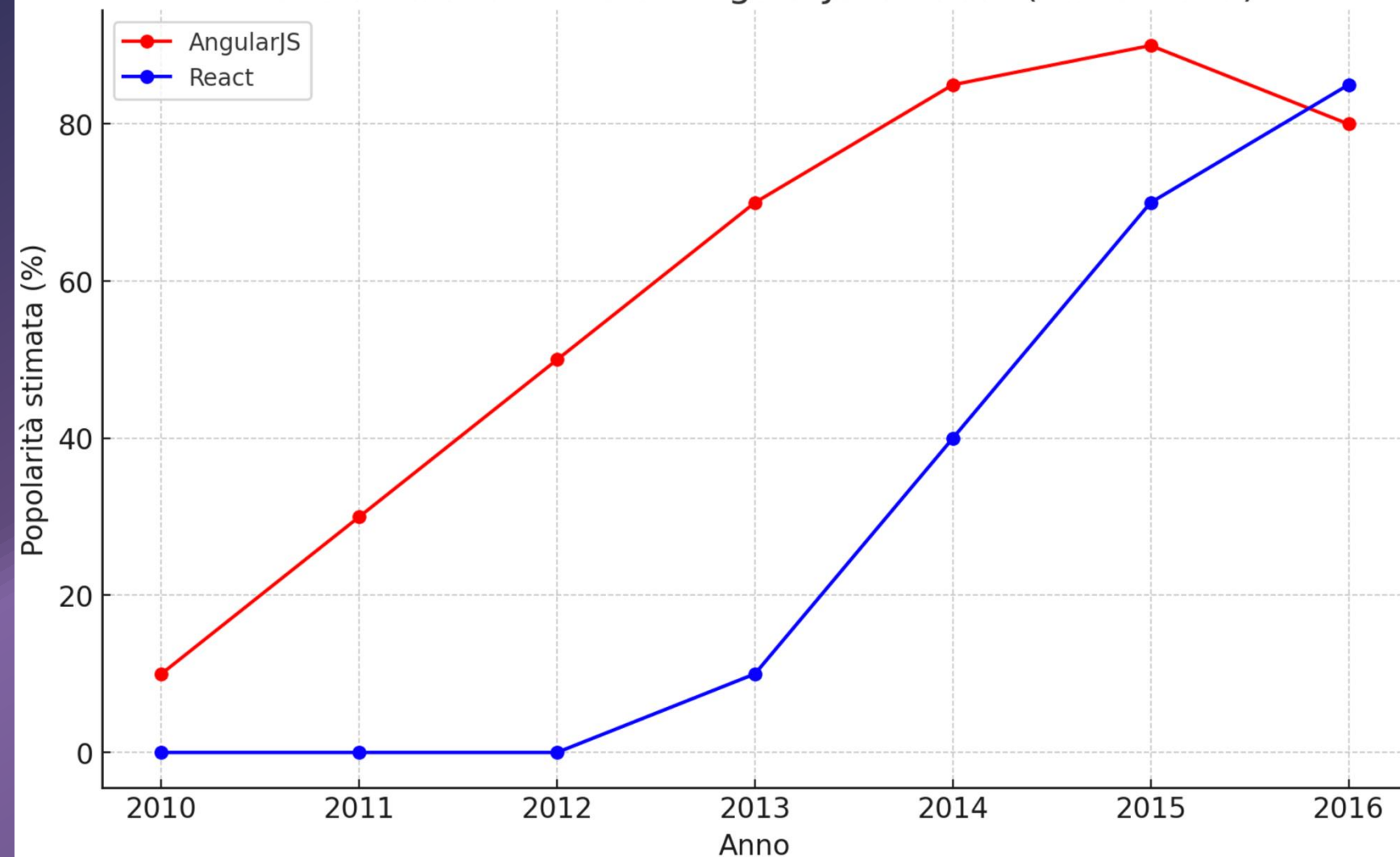
Compile-time

Quale poteva essere la
beginning version di 10 anni
fa?

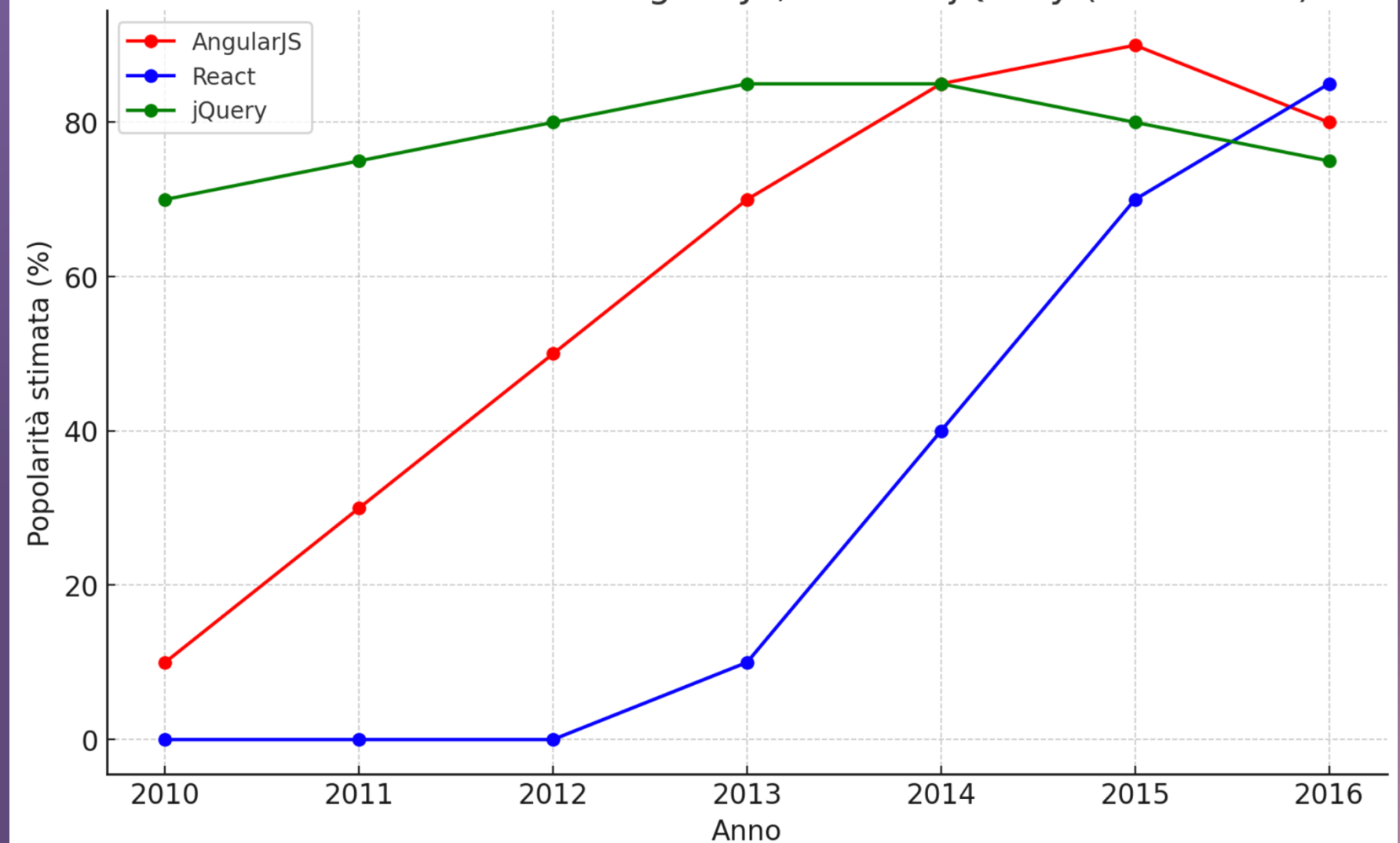


Quale poteva essere la beginning version di 10 anni fa?

Tendenze di utilizzo di AngularJS e React (2010-2016)

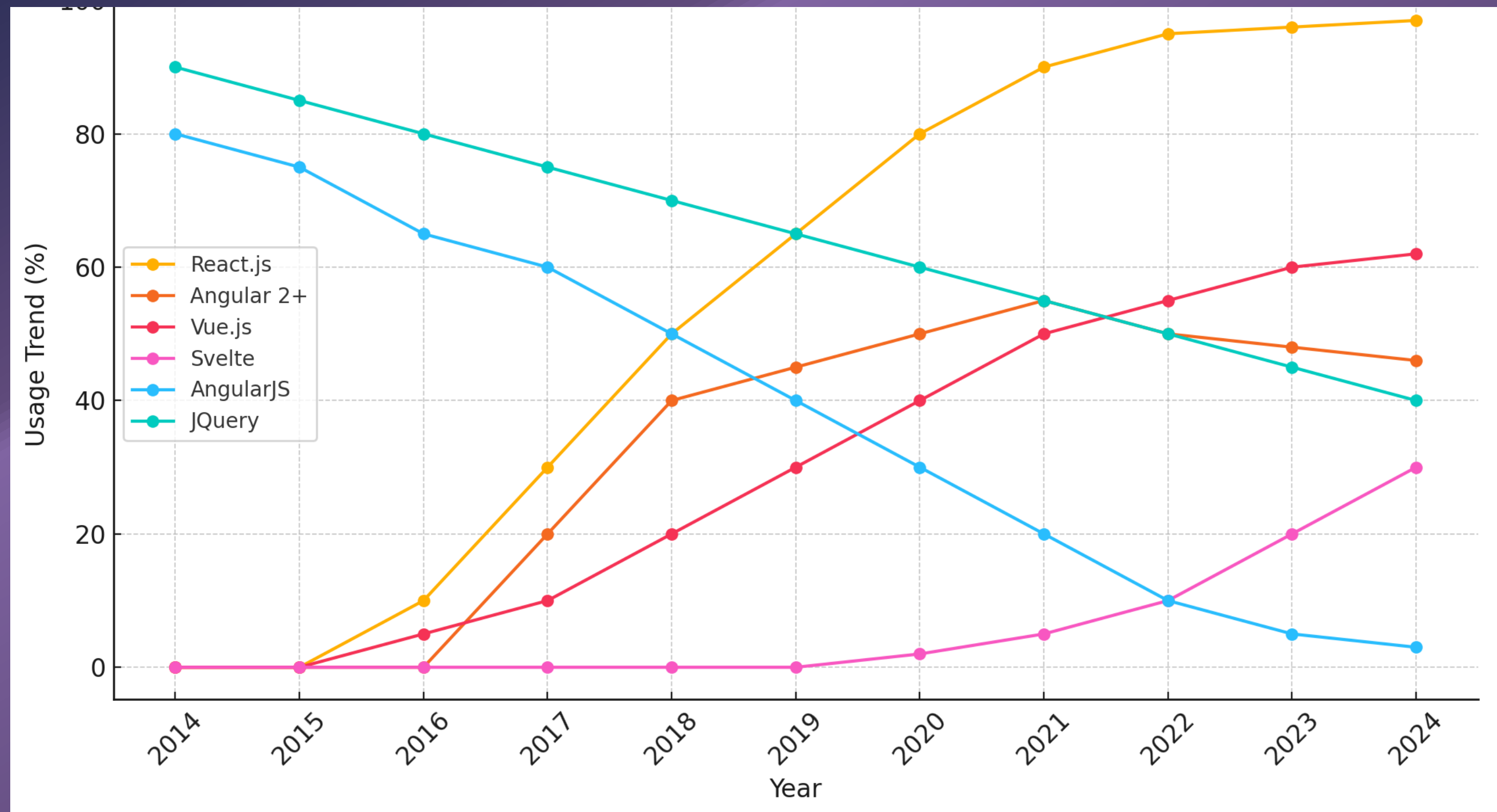


Tendenze di utilizzo di AngularJS, React e jQuery (2010-2016)





Situazione globale





2016 ANGULAR 2.0

Riscritto completamente per superare i limiti del precedente

Performance Migliorate

Component-Based Architecture

TypeScript (2012) 

Modularità Avanzata

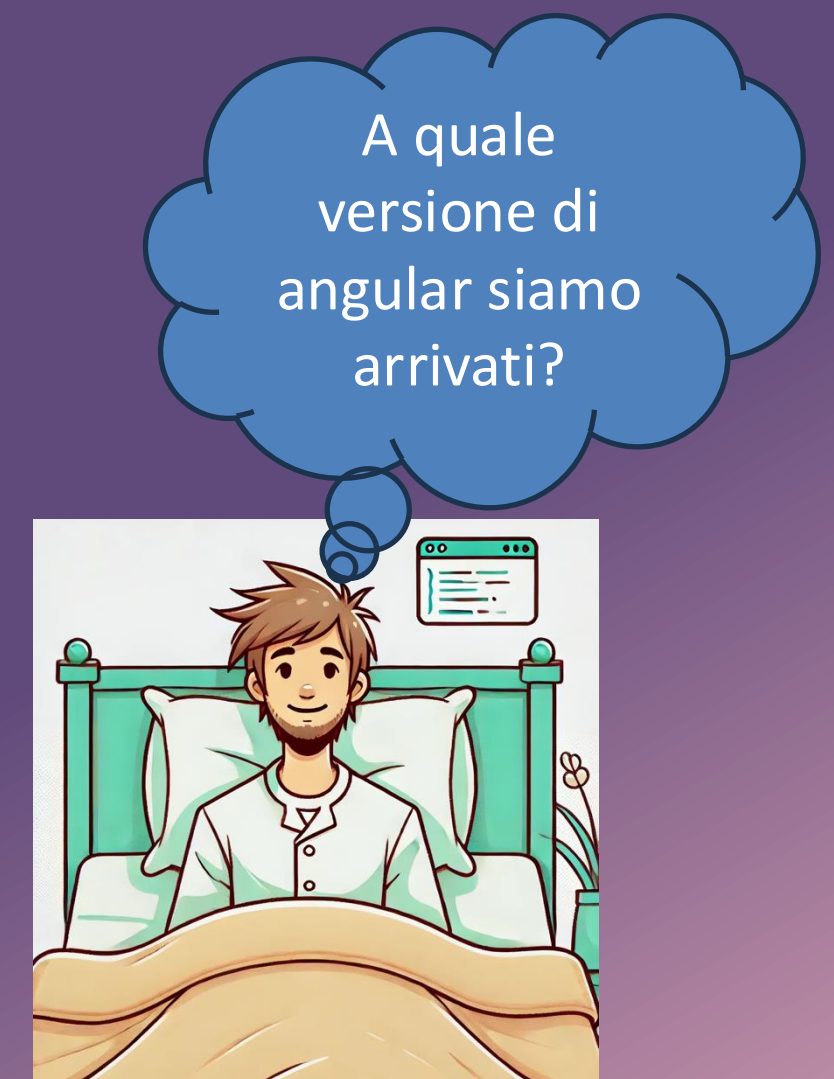
6 mesi nuova release – 18 mesi di supporto

2017 4.0 HttpClient e Angular Universal

2018 6.0 Angular Element

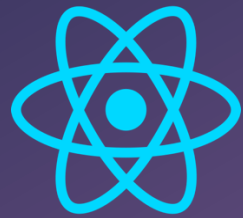
2022 14.0 Standalone Components

2023 Signals 16.0





Aggiornamenti



2015 – **React Native** (Mobile)

2017 – 16.8 **hooks** **Bye Bye Class Component**

2022 – 18.0 **SSR** Next – Remix - Expo **Bye Bye CRA**



2020 VUE 3: Composition API – Fragments
Teleport – Typescript



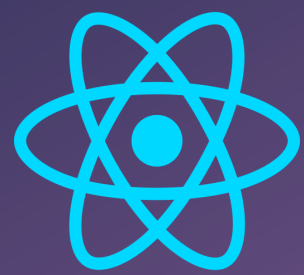
2019 Svelte 3: Svelte KIT

2024 Svelte 5: Rune



Bacaro
Tech
CODE AND FUN

Lato Server SSR - SSG



NEXTJS



nuxtjs



ANGULARUniversal



SVELTE KIT

Confrontiamo



Componenti e architettura

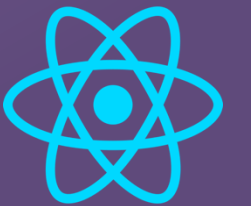


MVC

Fortemente **tipizzati** e
strutturato, forte separazione
HTML / CSS / TS
Dependency injection



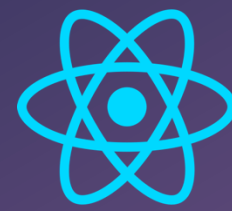
Semplici e modulari, con una
sintassi chiara e intuitiva.
Unico File



Functional Component e basati
su **hook**, che permettono di
gestire lo stato e gli effetti
collaterali.



Scritti direttamente nel
markup, con una **sintassi**
pulita e meno boilerplate.



Componente React

```
export const PokeHeader = () => {
  return (
    <Disclosure as="nav" className="bg-poke-primary-700 sticky top-0" >
      <div className="mx-auto max-w-7xl px-4 sm:px-6 lg:px-8">
        <div className="flex h-16 justify-between"> ...
      </div>
    </div>
    <DisclosurePanel className="md:hidden">
      <div className="space-y-1 px-2 pb-3 pt-2 sm:px-3">
        {navigation.map((item) => (
          <DisclosureButton
            key={item.name}
            as="a"
            href={item.href}
            aria-current={item.current ? 'page' : undefined}
            className={classNames(
              item.current ? 'bg-gray-900 text-white' : 'text-gray-300',
              'block rounded-md px-3 py-2 text-base font-medium',
            )}
            >
              {item.name}
            </DisclosureButton>
          ))}
      </div>
    </DisclosurePanel>
  )
}
```



Componente Angular

```
1  // app.component.ts
2  import { Component } from '@angular/core';
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7    styleUrls: ['./app.component.css']
8  })
9  export class AppComponent {
10    title = 'Benvenuto in Angular!';
11    counter = 0;
12
13    increment() {
14      this.counter++;
15    }
16  }
17
```

```
// app.component.html
<div class="container">
  <h1>{{ title }}</h1>
  <p>Contatore: {{ counter }}</p>
  <button (click)="increment()">Incrementa</button>
</div>

// app.component.css
.container {
  text-align: center;
  margin-top: 50px;
}
button {
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
}
```



Componente VUE

```
<template>
  <div class="container">
    <h1>{{ title }}</h1>
    <p>Contatore: {{ counter }}</p>
    <button @click="increment">Incrementa</button>
  </div>
</template>

<script>
export default {
  name: 'SimpleVueComponent',
  data() {
    return {
      title: 'Benvenuto in Vue!',
      counter: 0
    };
  },
  methods: {
    increment() {
      this.counter++;
    }
  }
};
</script>
```

```
<style scoped>
.container {
  text-align: center;
  margin-top: 50px;
}
button {
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
}
</style>
```




Componente Svelte

.svelte HTML CSS e JS
unico file

```
<script>
  import "../../app.css";
  import FakeSessionize from "../../components/FakeSessionize.svelte";
  import SourceFakeSessionize from "../../components/SourceFakeSessionize.svelte";

  function handleFlip() {
    console.log("click")
    const front = document.getElementById("front");

    const back = document.getElementById("back");
    if (front?.classList.contains('flipped')) {
      front?.classList.remove("flipped");
    } else {
      front?.classList.add("flipped");
    }

    if (back?.classList.contains('flipped')) {
      back?.classList.remove("flipped");
    } else {
      back?.classList.add("flipped");
    }
  }
</script>
```

```
<section>
  <h1>
    <span class="welcome">
      <picture>
        <source srcset={welcome} type="image/webp"/>
        <img src={welcome_fallback} alt="Welcome image" />
      </picture>
    </span>to your new<br />SvelteKit app
  </h1>
  <h2>
    try editing <strong>src/routes/+page.svelte</strong>
  </h2>
  <Counter />
</section>
```

```
<style>
  section {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    flex: 0.6;
  }
  h1 { ... }
  .welcome { ... }
  .welcome img { ... }
</style>
```



NFR non functional requirements

Mentre i requisiti funzionali definiscono **cosa** deve fare un software o sistema, i requisiti non funzionali definiscono **come** il sistema si deve comportare.

WHAT HOW

NFR Web Application

Performance

Manutenibilità

Portabilità

Interoperabilità

Supporto ed Evoluzione

SEO e Internazionalizzazione

Integrabilità

Installabilità e Distribuzione



Tipi Di Applicazioni

Single Page Application (SPA)

Multi Page Application (MPA)

Progressive Web App (PWA)

Client Side Rendering (CSR)

Server Side Rendering (SSR)

Static Site Generation (SSG)

Incremental Static Regeneration (ISR)

Applicazioni Web Responsive

Dashboard

E-Commerce

Marketplace

Messaggistica

CMS

Tipo di architettura

Tipo di rendering

Tipo di utilizzo





Tipi Di Applicazioni: Focus

Single Page Application (SPA)
Multi Page Application(MPA)

Tipo di architettura

Client Side Rendering (CSR)
Server Side Rendering (SSR)

Tipo di rendering

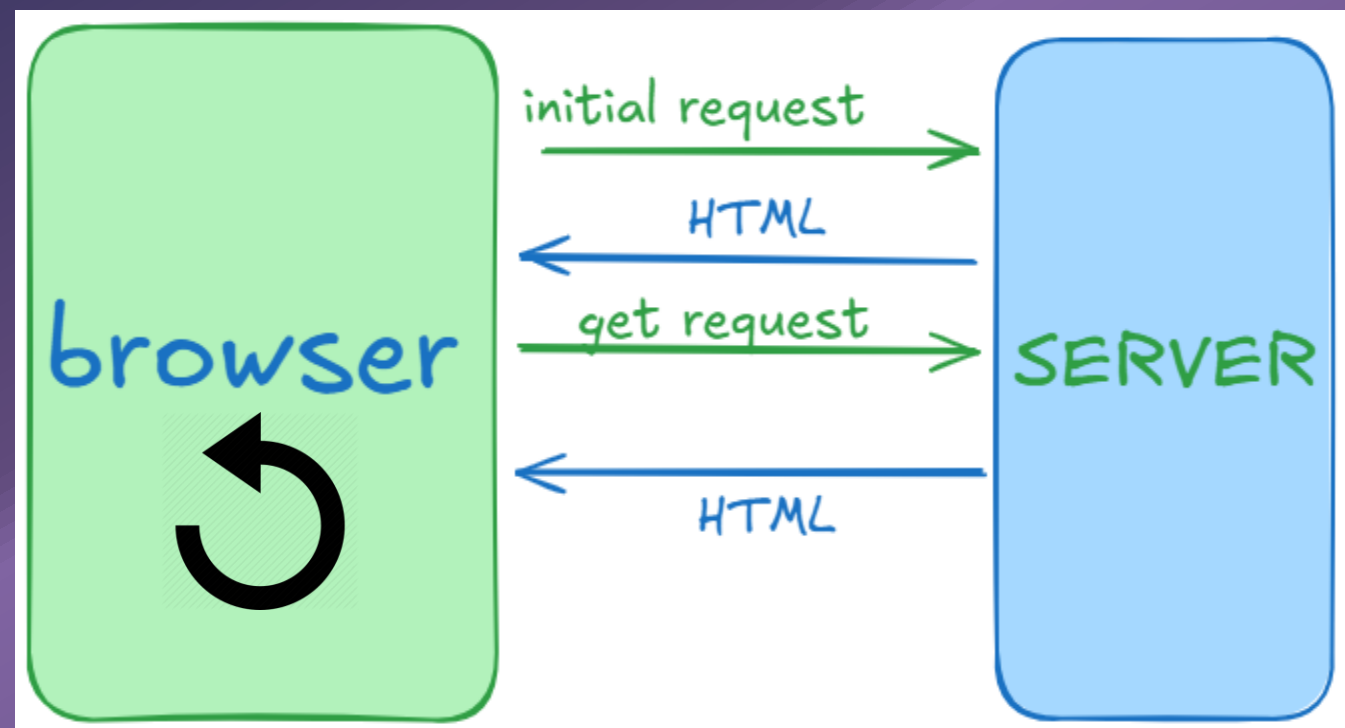
Applicazioni Web Responsive
Dashboard
Mobile
CMS

Tipo di utilizzo



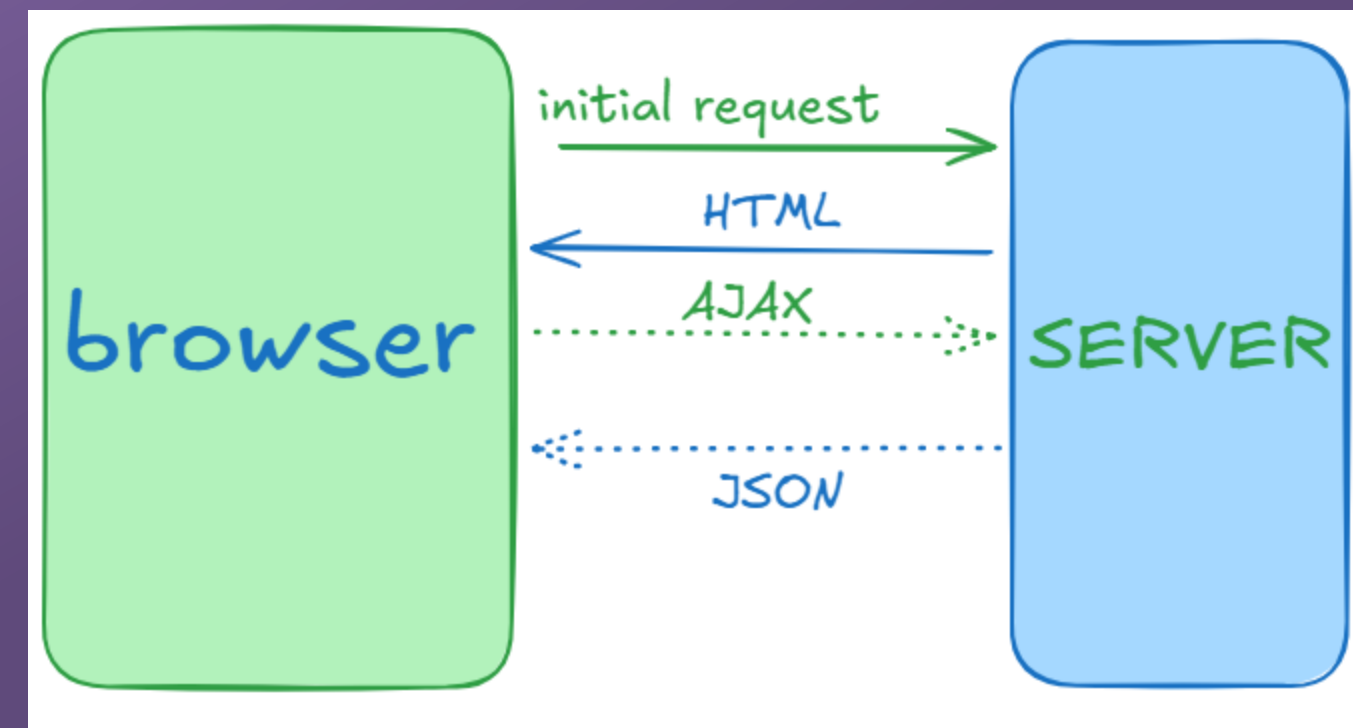
Tipi Di Architettura

MPA



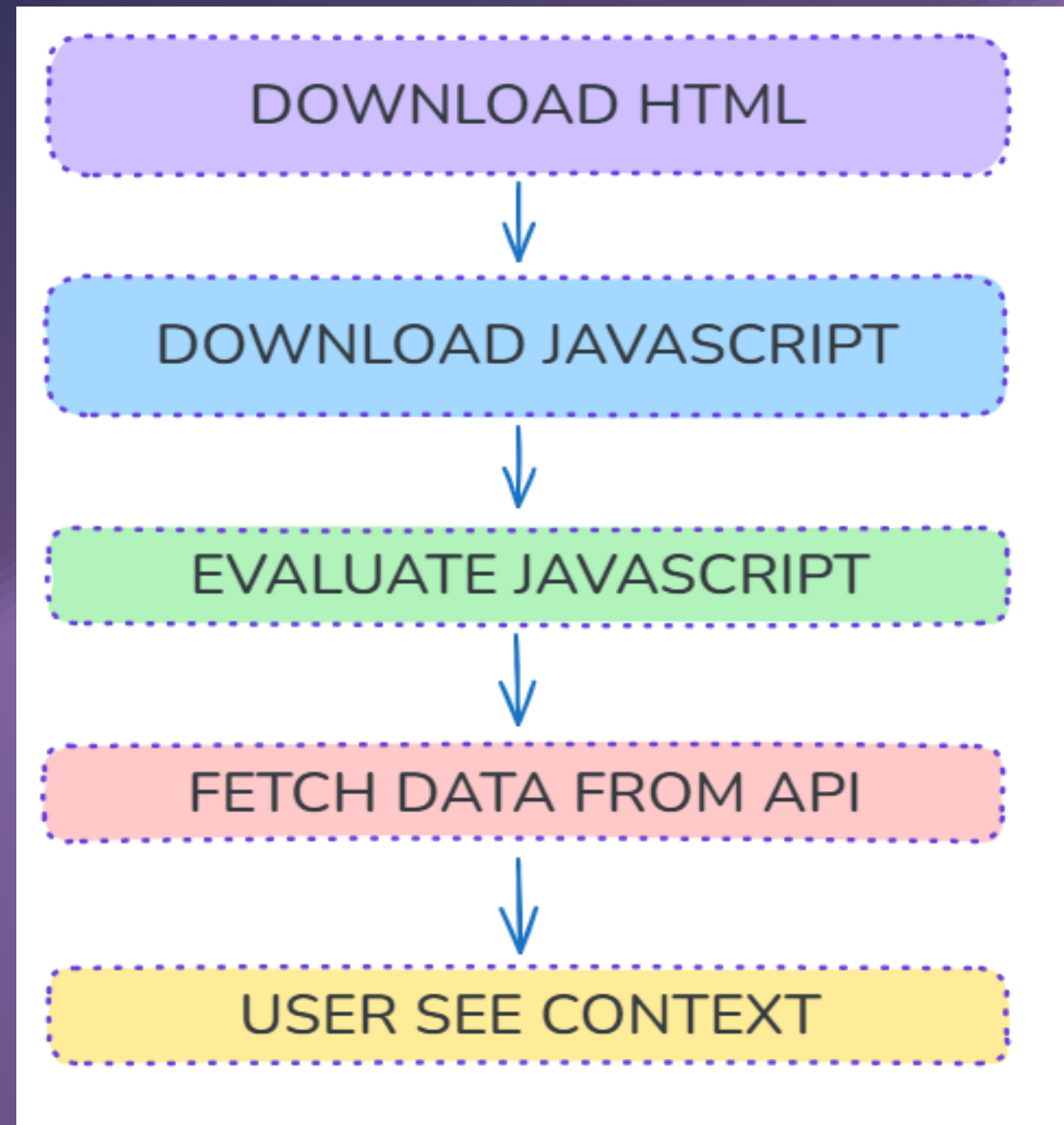
(Web site tradizionali)

SPA

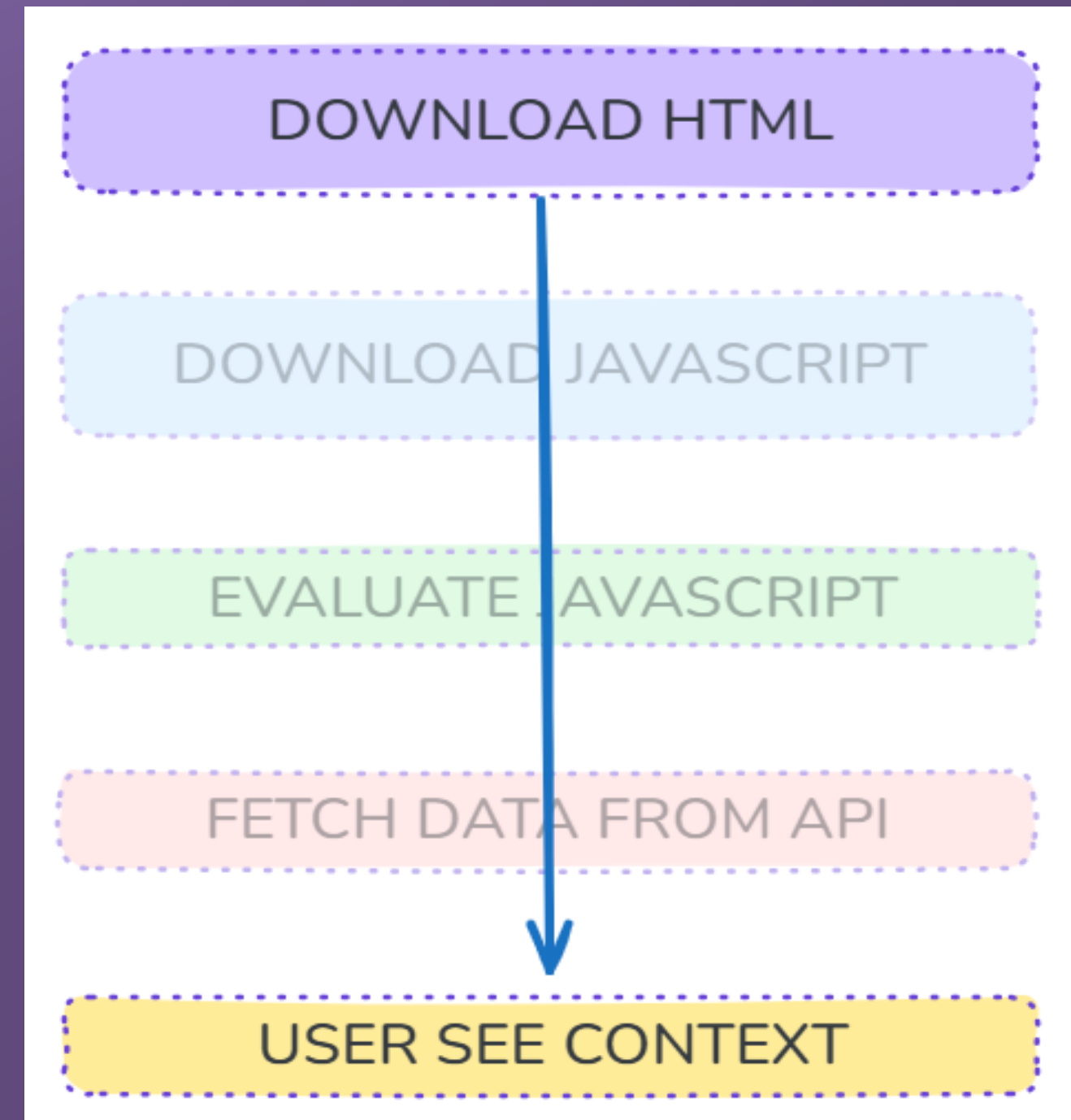








CSR



Tipi Di Rendering SSR



Match dei framework

SSR	CSR	SSG
Rendering work completed on server	Rendering done on the user's machine within browser	Rendering completed at build time before users visits site
✓ Personalization	✓ Personalization	✗ Personalization requires rehydration
👍 SEO, FCP, TTI	👍 FCP, TTFB, Low server costs	👍 SEO, TTFB, FCP, TTI
👎 TTFB, Blank Page Syndrome	👎 SEO, dependent on JavaScript	👎 Inflexible, build times
Common Frameworks: ASP.NET, Next.js, PHP (Laravel), Node.js	Common Frameworks: React, Angular, Vue    	Common Frameworks: Next.js, Gatsby, Hugu, Nuxt

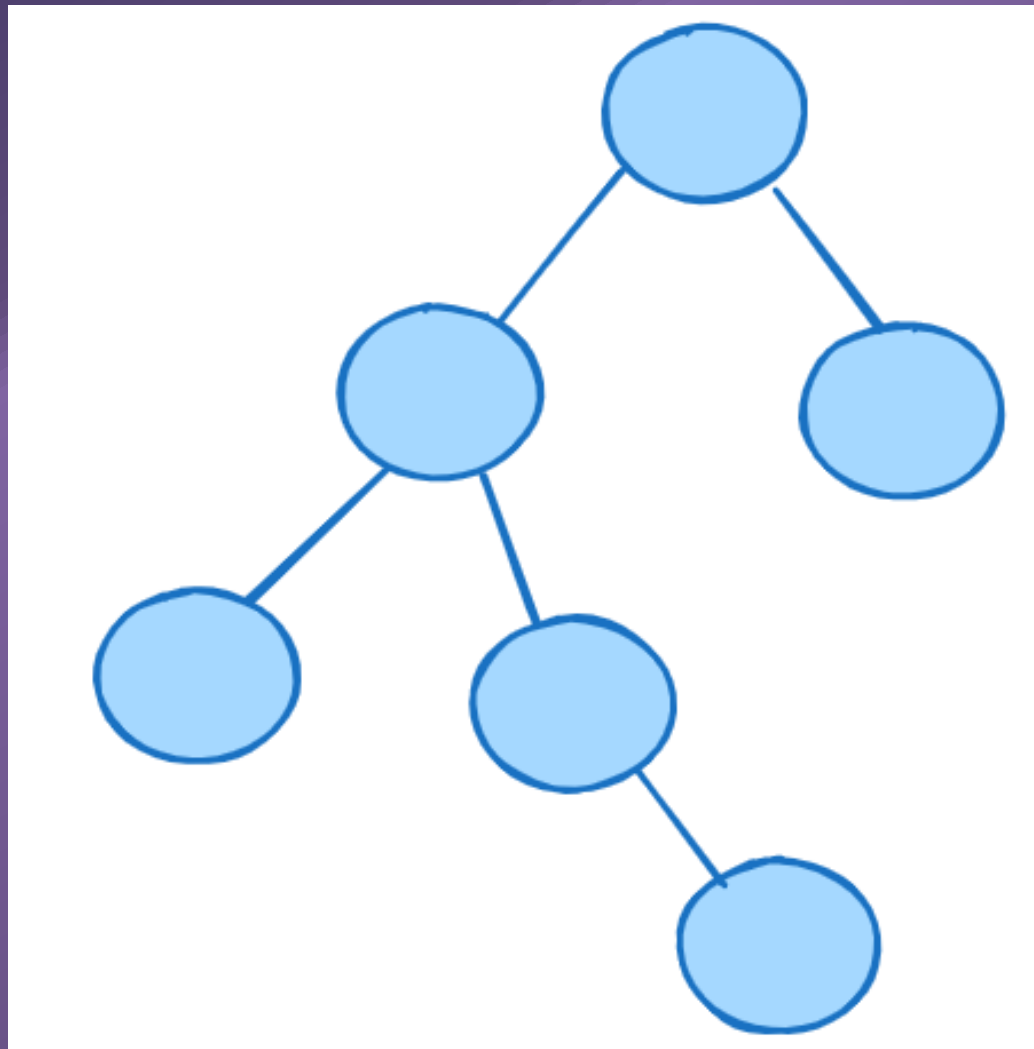


TTI = Time to interactive
FCP = First Content Paint
TTFB = Time to First Byte

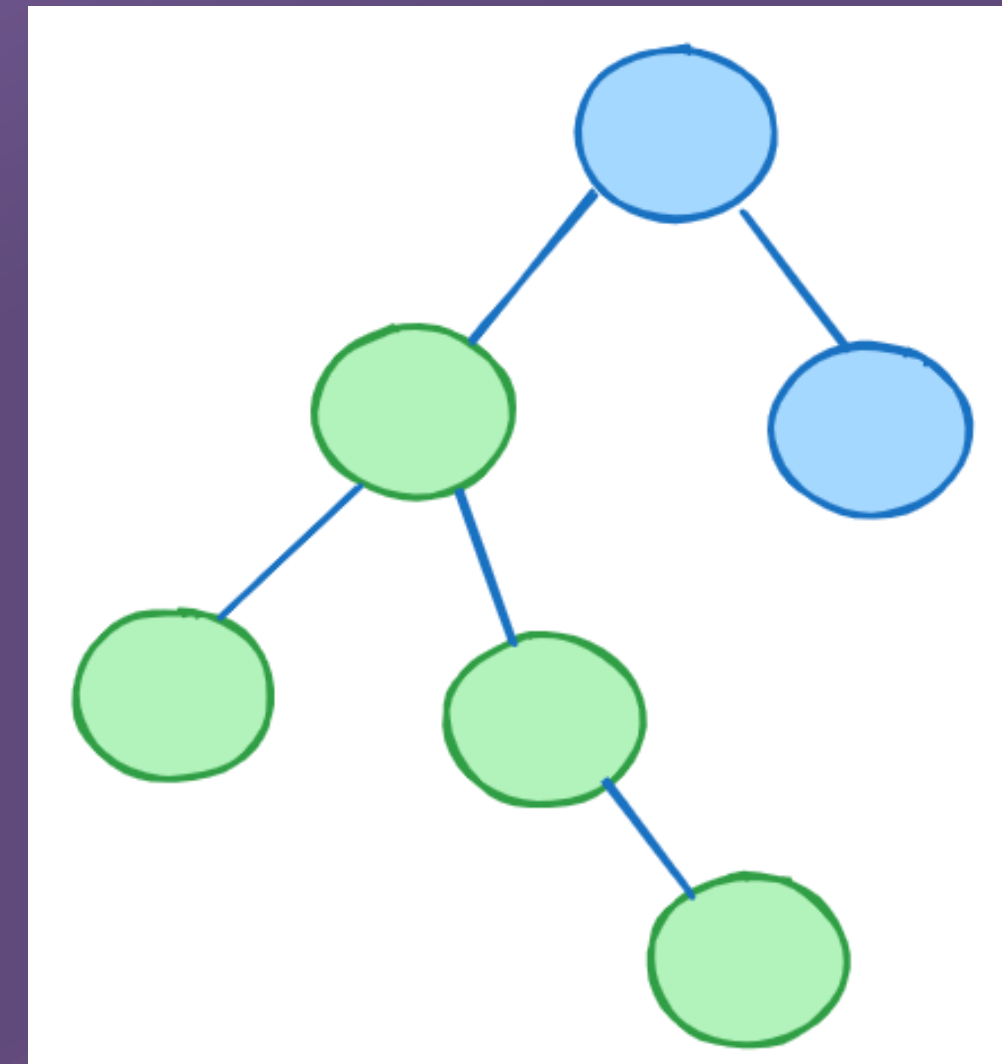
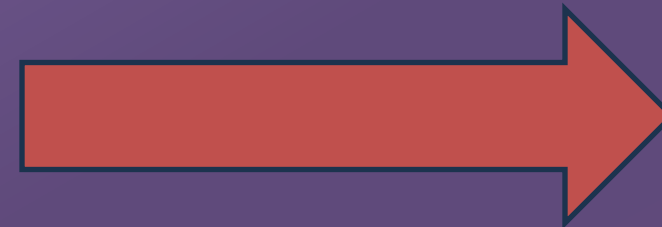


DOM e Virtual DOM

Il DOM è una rappresentazione strutturata del documento HTML
Il Virtual DOM è una copia virtuale del DOM reale, usata per rendere gli aggiornamenti più efficienti



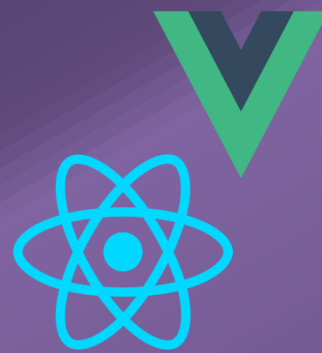
DIFFING





Runtime vs Compile-time

Runtime: Codice eseguito direttamente nel browser



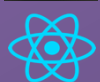
Utilizza Virtual DOM Change Detection
per ottimizzare l'aggiornamento del DOM



**Compile time: Codice ottimizzato e trasformato
prima di essere eseguito**



Compila il codice in JavaScript nativo,
eliminando la necessità di un Virtual DOM
o runtime.





Runtime vs Compile-time



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
  <title>Mini React Component</title>
  <script src="https://unpkg.com/react@17/umd/react.
  <script src="https://unpkg.com/react-dom@17/umd/re
</head>

<body>
  <div id="root"></div>
  <script>
    function MyComponent() {
      return React.createElement('h1', null, 'He
    }
    ReactDOM.render(
      React.createElement(MyComponent),
      document.getElementById('root'));
  </script>
</body>
</html>
```



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-wi
  <title>Mini Vue Component</title>
  <script src="https://cdn.jsdelivr.net/npm/vue@
</head>
<body>
  <div id="app">
    <my-component></my-component>
  </div>
  <script>
    Vue.component('my-component', {
      template: '<h1>Hello, Vue!</h1>'
    });
    new Vue({
      el: '#app'
    });
  </script>
</body>
</html>
```





Runtime vs Compile-time

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-wi
  <title>Mini Angular Component</title>
  <!-- Angular Libraries from
  <script src="https://unpkg.c
  <script src="https://unpkg.c
  <script src="https://unpkg.c
  <script src="https://unpkg.c
  </script>
</head>
```



```
<script>
  const { Component, NgModule } = ng.core;
  const { BrowserModule } = ng.platformBrowser;
  const { platformBrowserDynamic } = ng.platformBr

  @Component({
    selector: 'my-app',
    template: `<h1>Hello, Angular!</h1>`

    AppComponent { }

    module({
      eclarations: [AppComponent],
      mports: [BrowserModule],
      bootstrap: [AppComponent]
    })
    class AppModule { }

    platformBrowserDynamic().bootstrapModule(AppModu
      .catch(err => console.error(err));
</script>
```

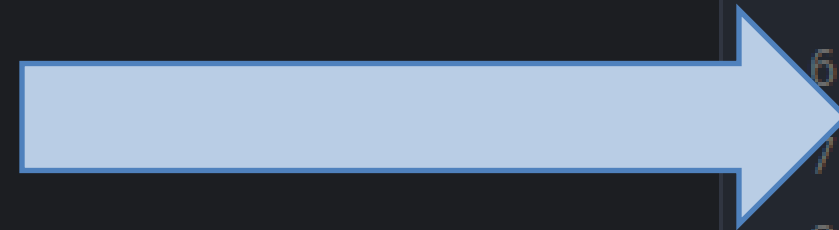




Runtime vs Compile-time



App.svelte		RUNES	MIGRATE	Result	JS output	CSS output	AST output
1	<script>			1	import "svelte/internal/disclose-version";		
2	let name = 'world';			2	import "svelte/internal/flags/legacy";		
3	</script>			3	import * as \$ from "svelte/internal/client";		
4				4			
5	<h1>Hello {name}!</h1>			5	var root = \$.template(`<h1></h1>`);		
6				6			
				7	export default function App(\$\$anchor) {		
				8	let name = 'world';		
				9	var h1 = root();		
				10			
				11	h1.textContent = `Hello \${name ?? ""}!`;		
				12	\$.append(\$\$anchor, h1);		
				13	}		





Caratteristiche Programmazione

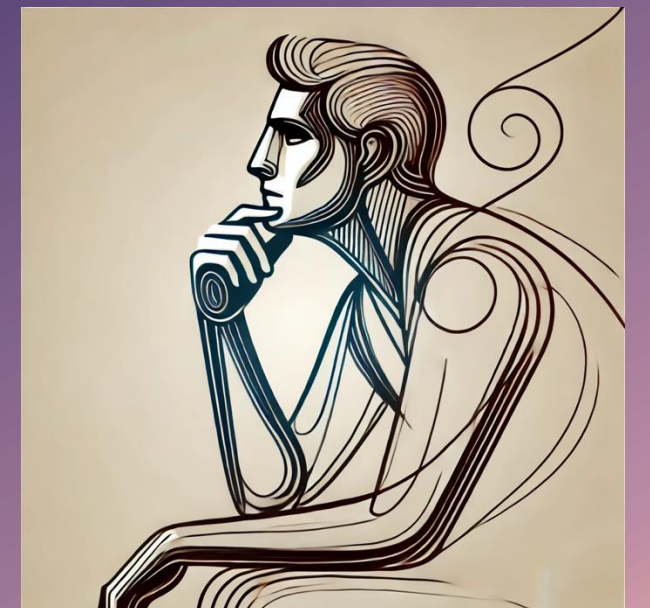
Imperativo / Dichiarativa: *come / che cosa*

Reattiva: flussi di dati

Paradigma : OOP *class* - Funzionale *function()*

Tipizzazione: Typescript - Javascript

Un buon codice consiste nel sapere *quando* usare ogni tipo di programmazione.





Match Framework

runtime	compile	imperative	declarative	reactive	Class	Function
✓			✓	✓		✓
✓	✓	✓	✓	✓	✓	
✓			✓	✓	✓	✓
	✓		✓	✓		✓



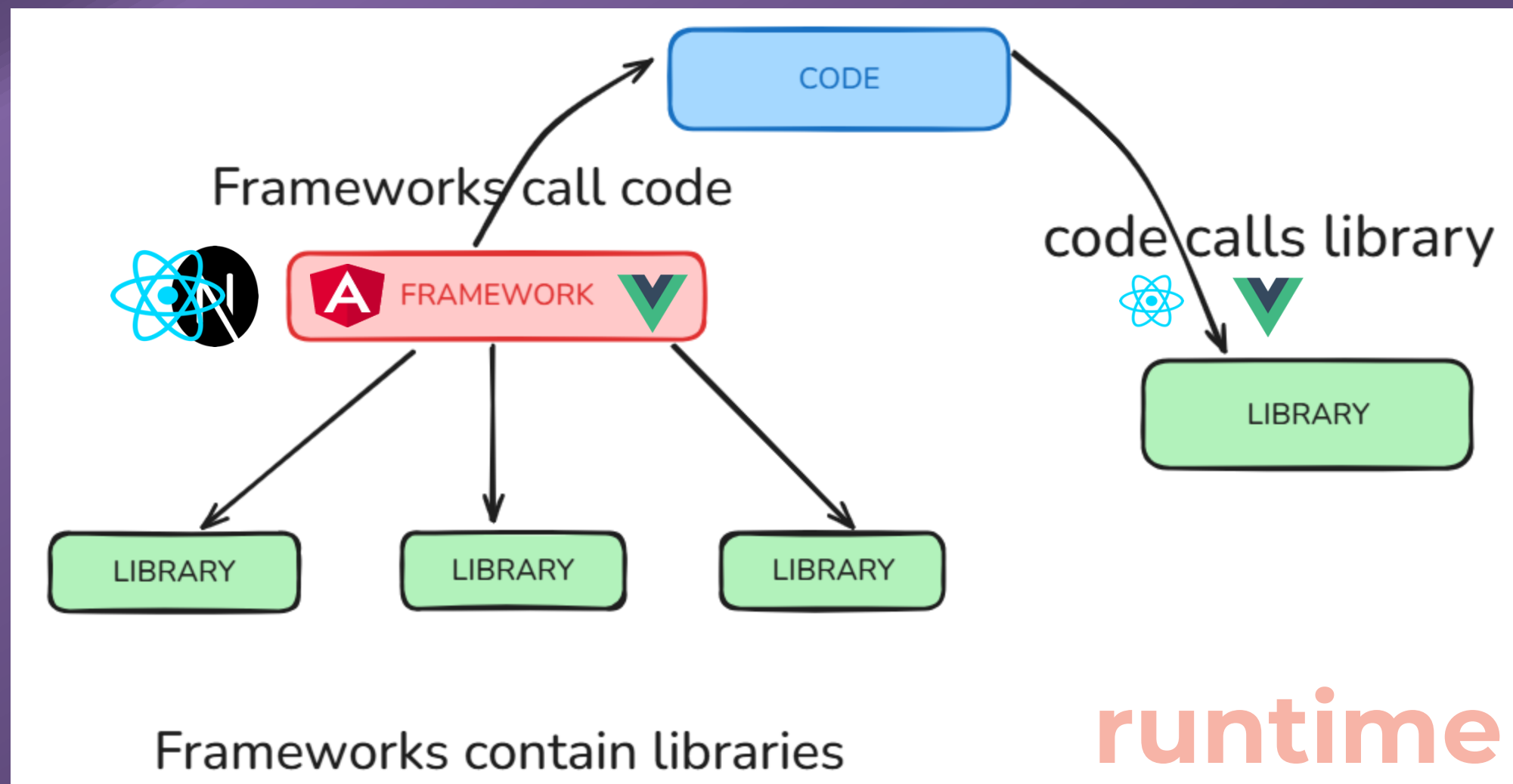
c
o
n
f
r
o
n
t
i



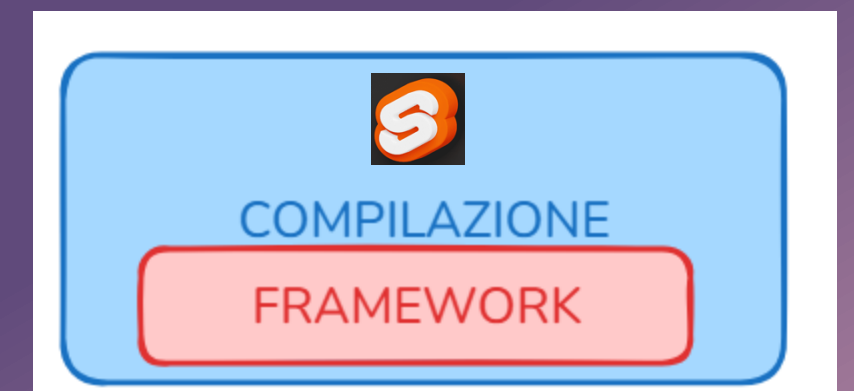


Warning Framework

1. I framework tendono a prendere decisioni al posto nostro
2. Con la loro evoluzione, ci fanno cambiare scelte diverse nel corso del tempo



Compile time





State Management

Tutte le informazioni che descrivono lo stato dell'interfaccia utente in un dato momento.

Local State: Componente

Global State: condiviso o tutto l'applicativo

Derived State: Calcolato

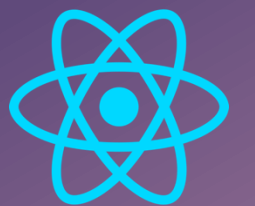


RxJS – NgRx
Services



Vuex
Composition API

Local State: useState
**Global State: Redux /
context**



Writable store
Derived Store





Data Binding

Il data binding è il meccanismo che permette di sincronizzare i dati tra il modello dell'applicazione (logica) e la vista (UI).

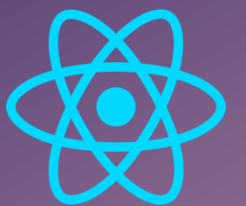
One Way: Sincronizzazione unidirezionale. I dati passano dal modello alla vista

Two Way Sincronizzazione bidirezionale. I dati vengono sincronizzati sia dal modello alla vista che dalla vista al modello.



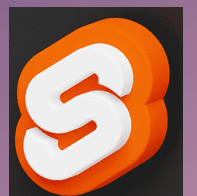
Two Way: [(ngModel)]

One Way



Two Way v-model
One Way v-bind

Two Way: direttive
One Way: Props ed
eventi

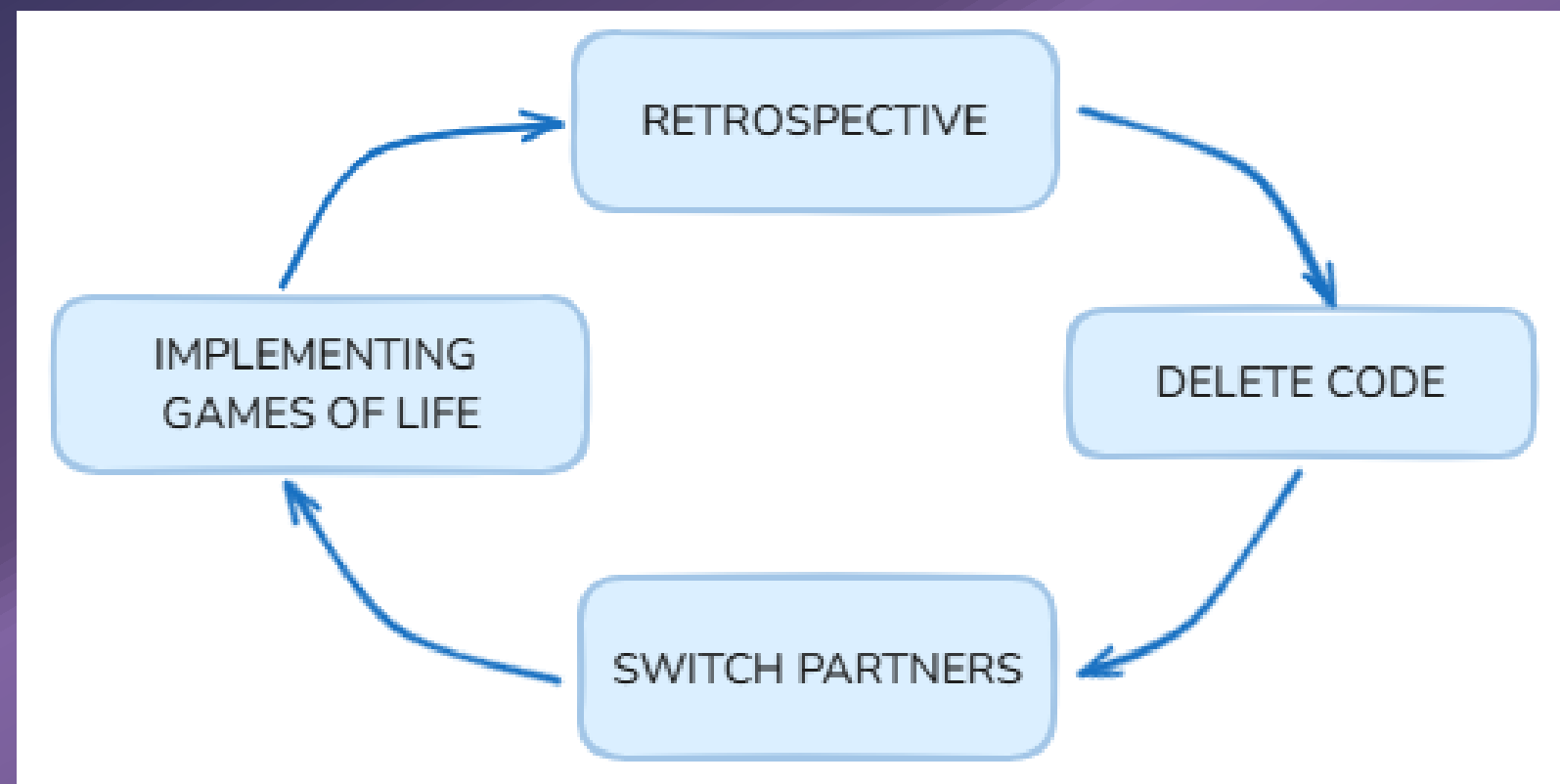


Proviamo



Code Retreat

Questo è un evento intensivo di una giornata dedicato al miglioramento delle competenze nello sviluppo software.



Effetti

- Inizia da zero
- Nuovi apprendimenti senza vincoli
- Libertà di scegliere strategie nuove
- Toglie il peso del codice legacy
- Adattamento ai vincoli di ogni sessione

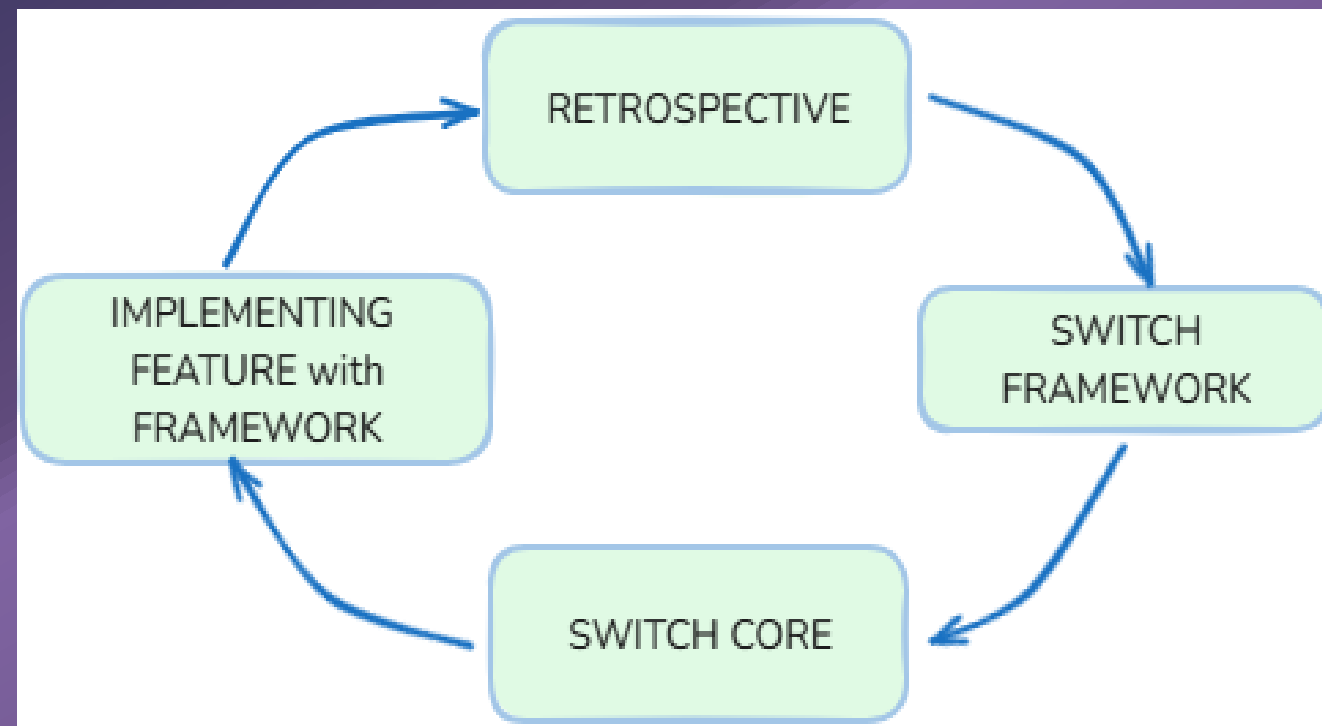
<https://www.coderetreat.org/>





Code Retreat

Possiamo applicare questo processo nel fare pratica con una nuova tecnologia frontend. Il problema, di fatto, è sempre quello: la pagina web!



- Inizia da zero – **cambiando tecnologia devi rifare quella feature usando**
- Nuovi apprendimenti – **vincoli del framework**
- Libertà di scegliere strategie nuove – **Imposizione delle strategie del framework**
- Adattamento ai vincoli di ogni sessione – **del nuovo framework**

Provare con progetti pratici!

Includete chiamate ad api, autenticazioni e casi limite



Esperimento Pokedex

PokeAPI

Home About API v2 GraphQL v1beta

PokeAPI turns 10 on December 4th! Celebrate and discover the entire source code over at [PokeAPI/pokeapi](https://github.com/PokeAPI/pokeapi)!

PokeAPI

The RESTful Pokémon API

Serving over 2.5 billion API calls each month!

All the Pokémon data you'll ever need in one place,
easily accessible through a modern free open-source RESTful API.

[Check out the docs!](#)

Try it now!

Need a hint? Try [pokemon/ditto](#), [pokemon-species/aegislash](#), [type/3](#), [ability/battle-armor](#), or [pokemon?limit=100000&offset=0](#).

Direct link to results: <https://pokeapi.co/api/v2/pokemon/ditto>

PokeAPI / pokeapi

Code Issues 134 Pull requests 6 Discussions Actions Projects

pokeapi Public

master 12 Branches 8 Tags

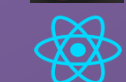
Go to file

Naramsim Merge pull request #1165 from Deleca7755/master

.circleci	cicd: generate openapi schema
.github	refactor: remove debug
Resources	fix: add -y
config	docs: update tag descriptions


pokeapi

- > ▶ nginx:1.25.3-alpine pokeapi-web-1 - Up 18 minutes
- > ▶ pokeapi/pokeapi:master pokeapi-app-1 - Up 19 minutes
- > ▶ hasura/graphql-engine:v2.36.4 pokeapi-graphql-engine-1 -
- > ▶ postgres:16.1 pokeapi-db-1 - Up 19 minutes
- > ▶ redis:7.2.3-alpine pokeapi-cache-1 - Up 19 minutes





Esperimento Pokedex

 **pokedex-frontend** Public

develop

6 Branches

0 Tags

Go to file

This branch is 21 commits ahead of main.

jollymick90

cdn test

vscode	test next
assets	create flip sessionize - svelte version
docs	build mock
frontend-iniections	cdn test
mock	create flip sessionize - svelte version
pokedex-angular	create angular app
pokedex-next	test next
pokedex-react	Merge branch 'temp' into feature/cre
pokedex-svelte	configuration pokedex base app svelt
pokedex-vue	create vue app
sessionize-flip-svelte	create next pokedex version
.gitignore	create Footer Header and PokemonLi
README.md	create flip sessionize - svelte version





Considerazioni

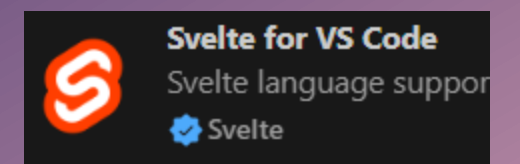
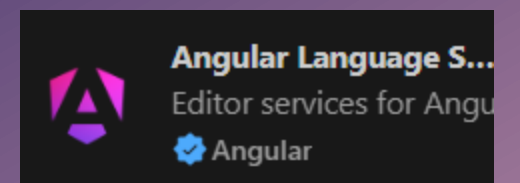
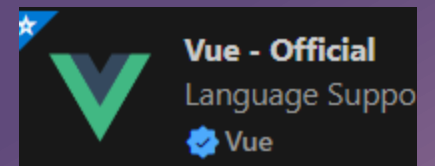
Un'unica tecnologia di CSS -> Tailwind

Sempre lo stesso Layout / Mock

Sempre la stessa struttura di cartelle (quando possibile)

Estensioni VSCode e formattazioni

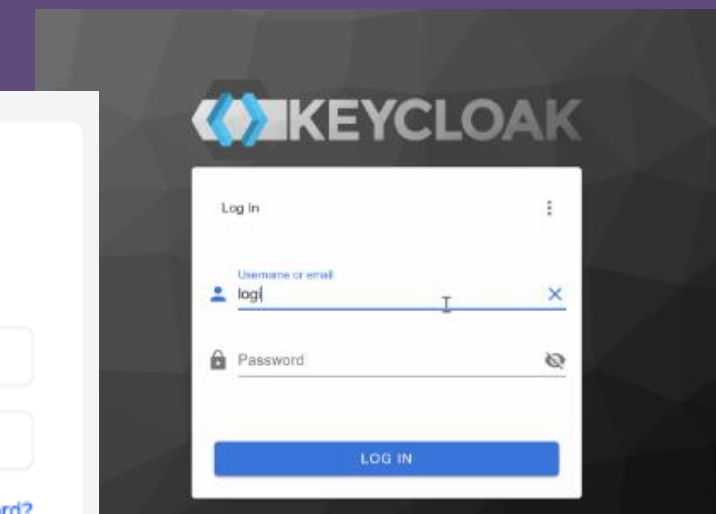
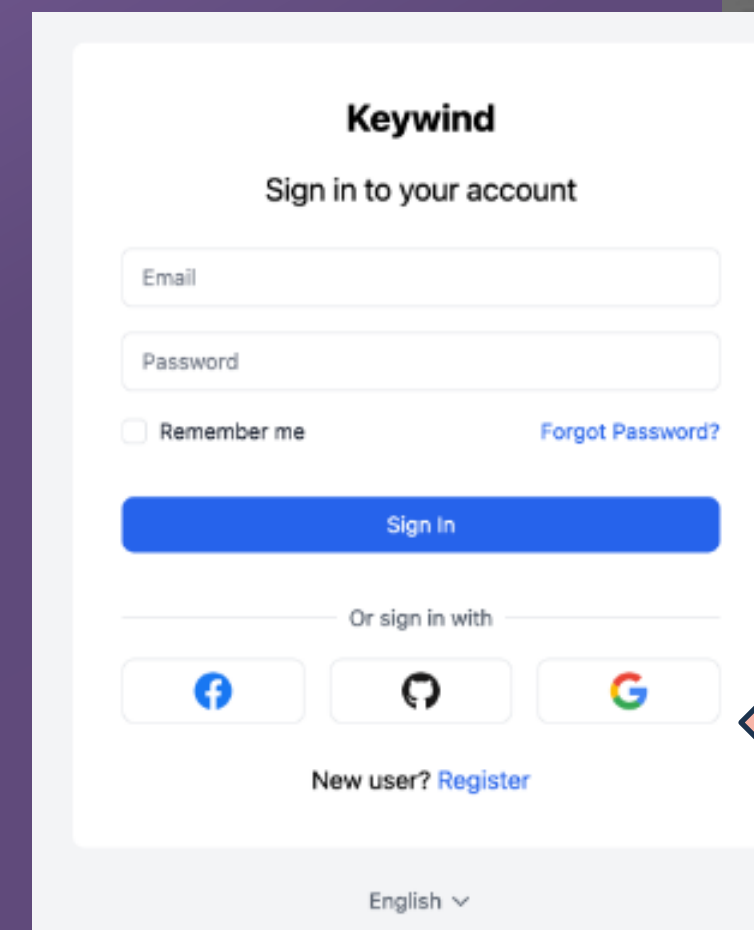
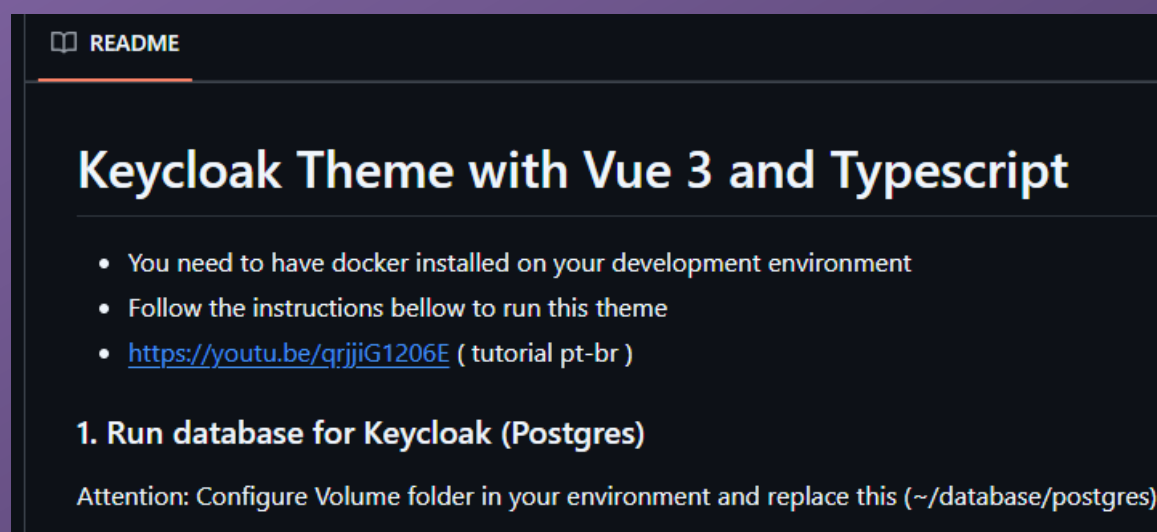
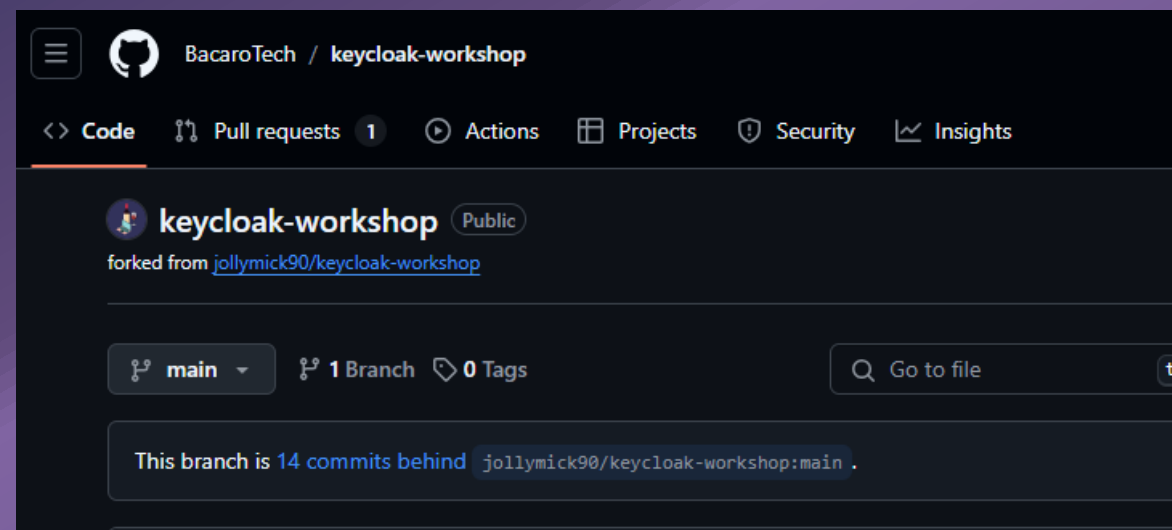
Utile anche con gli aggiornamenti dei framework familiari





Pratica: Customize Keycloak

Applicare librerie o framework per rendere più dinamiche, scalabili, strutturate, possibili. Keycloak è Identity Provider Java che renderizza le pagine di gestione dell'autenticazione usa pagine FTL e usa un





Pratica No Framework

Provare a scrivere qualcosa da zero senza librerie o framework. Puoi riscrivere il tuo modo di fare data-binding o di gestire lo stato.

Lo scopo non è, solo, di riscrivere il virtual dom o un nuovo modo di gestire gli stati ma di capire il funzionamento di esse e provare a dare il proprio contributo

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





Bacaro
Tech

CODE AND FUN

Pratica **Porsi dei limiti**

Limitare la RAM

Limitare la velocità rete

Limitare aggiornamenti

Embedded / WebView





Risultati Consigli

Questi sono i macro punti che penso siano fondamentali da affrontare per ogni framework

- **Ecosistema e community**
- **Core: Rendering – architettura – state - forms**
- **Tooling**
- **Typescript**
- **Curva di apprendimento**
- **Use Case and Best Fit**
- **Capire quanto sia facile integrare librerie.
Soprattutto quelle legacy**





... tornando alla roadmap

possiamo variare framework per aumentare la superficie esplorata

Possiamo specializzarci in una nicchia per andare in profondità dove pochi possono

Partiamo dai framework per avere più consapevolezza della programmazione. (clean-code, tipi di programmazione, librerie, architettura)



**GRAZIE DELLA VOSTRA
ATTENZIONE!**

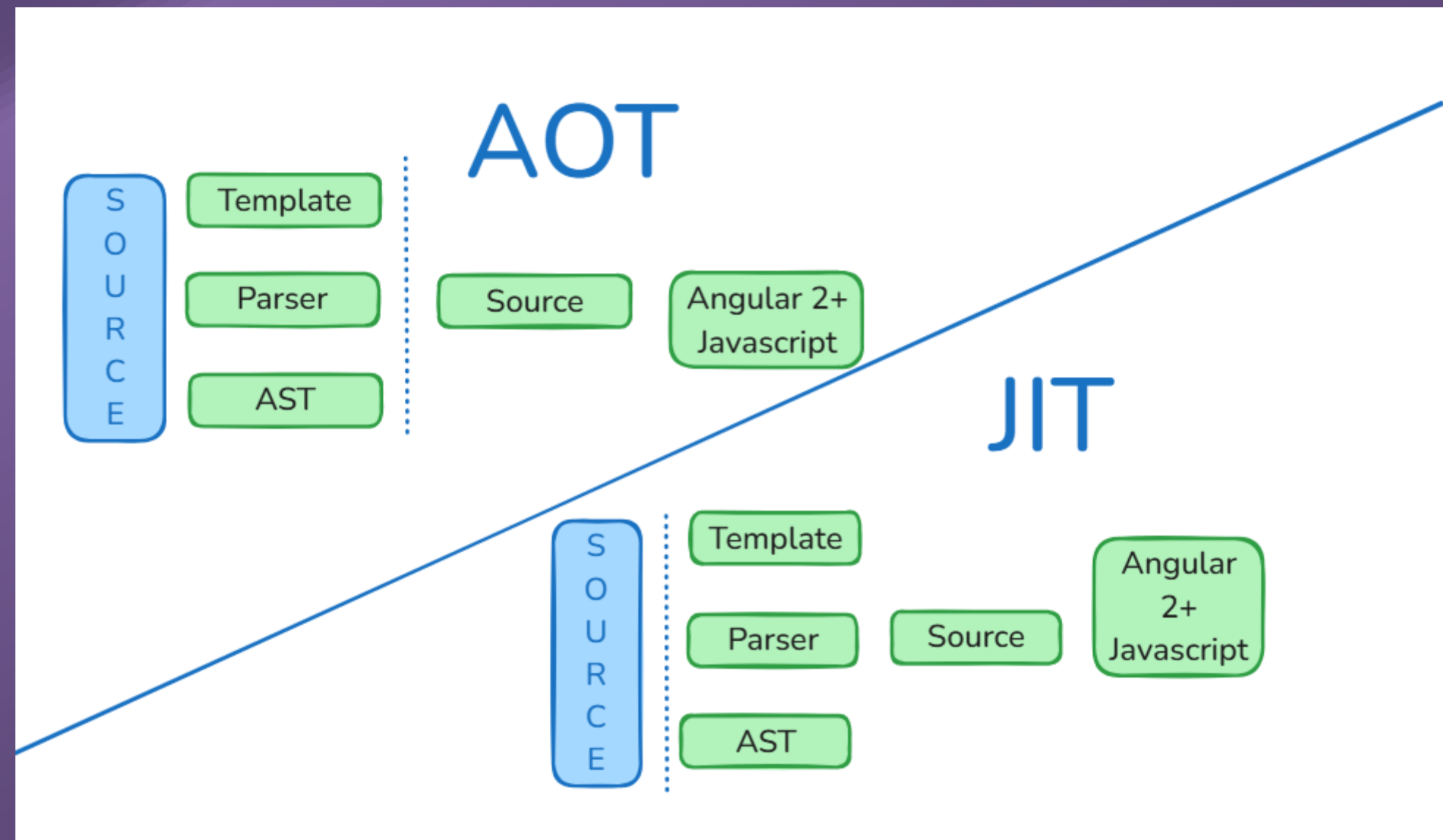


JIT e AOT

sono due modalità di compilazione del codice

Just-in-Time (JIT) Quando avviene la compilazione: La compilazione avviene al momento dell'esecuzione, ovvero direttamente nel browser dell'utente.

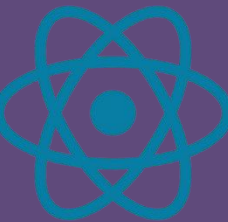
Ahead-of-Time (AOT) Ahead-of-Time (AOT) Compilation



Performance and Optimization



Runtime Performance
Ottimizzato per applicazioni
larga scala
Supporta SSG & SSR



Virtual DOM
Reconciliation
Hooks e Memoization
Supporta SSG & SSR



Runtime Performance
Ottimizzato per applicazioni
larga scala
Supporta SSG & SSR



Veloce nel caricamento
iniziale
Ridotta Dimensione Bundle
Supporta SSG & SSR



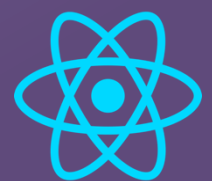
Change Detection



Virtual DOM

ottimizzato per piccolo cambiamenti

Rendering



Virtual Dom
Reconciliation Algorithm



Compila Vanilla

Javascript al build time



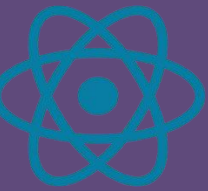
Code Simplicity and Readability



Dependency Injection e typescript migliorano la struttura e la leggibilità del codice.



Adatto ai principianti, con una struttura semplice e una sintassi chiara. La reattività integrata riduce il boilerplate e migliora la leggibilità.



JSX offre unicità e potenza, mentre gli hooks semplificano la gestione dello stato in applicazioni complesse.



Sintassi diretta e modello di reattività in cui le variabili aggiornano automaticamente l'interfaccia, riducendo la complessità del codice.



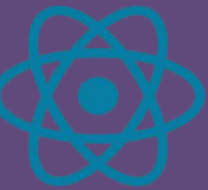
Ecosystem e Community



Google
Ecosistema Stabile



Ecosistema forte
Buona community
Molti sviluppatori poche
aziende leader



Meta
Ecosistema Vasto
Supportato da Aziende e
Sviluppatori

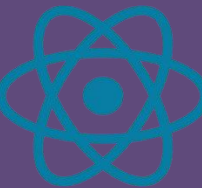


Ecosistema più piccolo
Community dedicata

Tooling



Common Line Interface
Crei quasi tutto
NX - Workspace



CRA
NextJS
React Native
Webpack - Vite



CLI per creare progetto
Vite



SvelteKIT
Semplifica SSR SSG

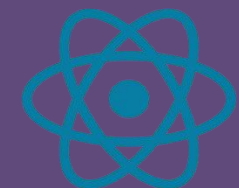


Nativo da sempre



Opzionale
Ben supportato da V3

Typescript

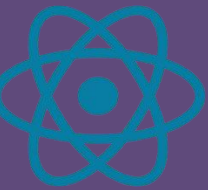


Opzionale
Largamente diffuso



Opzionale

Curva di apprendimento



Ripida

Moderata



Adatto ai principianti

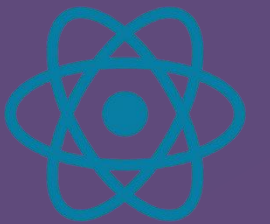
Accessibile e semplice

Use Cases and Best Fit



Enterprise

Dynamic
High Traffic
Scalability
Flexibility



produttività e facilità d'uso,
ampiamente usato in
eCommerce e dashboard di
amministrazione.

piccole e medie
performance critical
bundle size





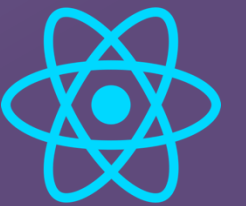
Todo Approcci di Reattività



Change Detection
Two-Way Data Binding -
ngModel
RxJS e Observables
Lifecycle Hooks



Proxy
data binding bidirezionale v-
model
computed properties
Watchers
reattività annidata



VirtualDOM
Stato e Props
Hooks
Immutabilità



Compilazione Reattiva – NO
Vdom o change detection
Basata su assegnazione per
aggiornare
Dichiarazioni reattive \$