

SUPSI

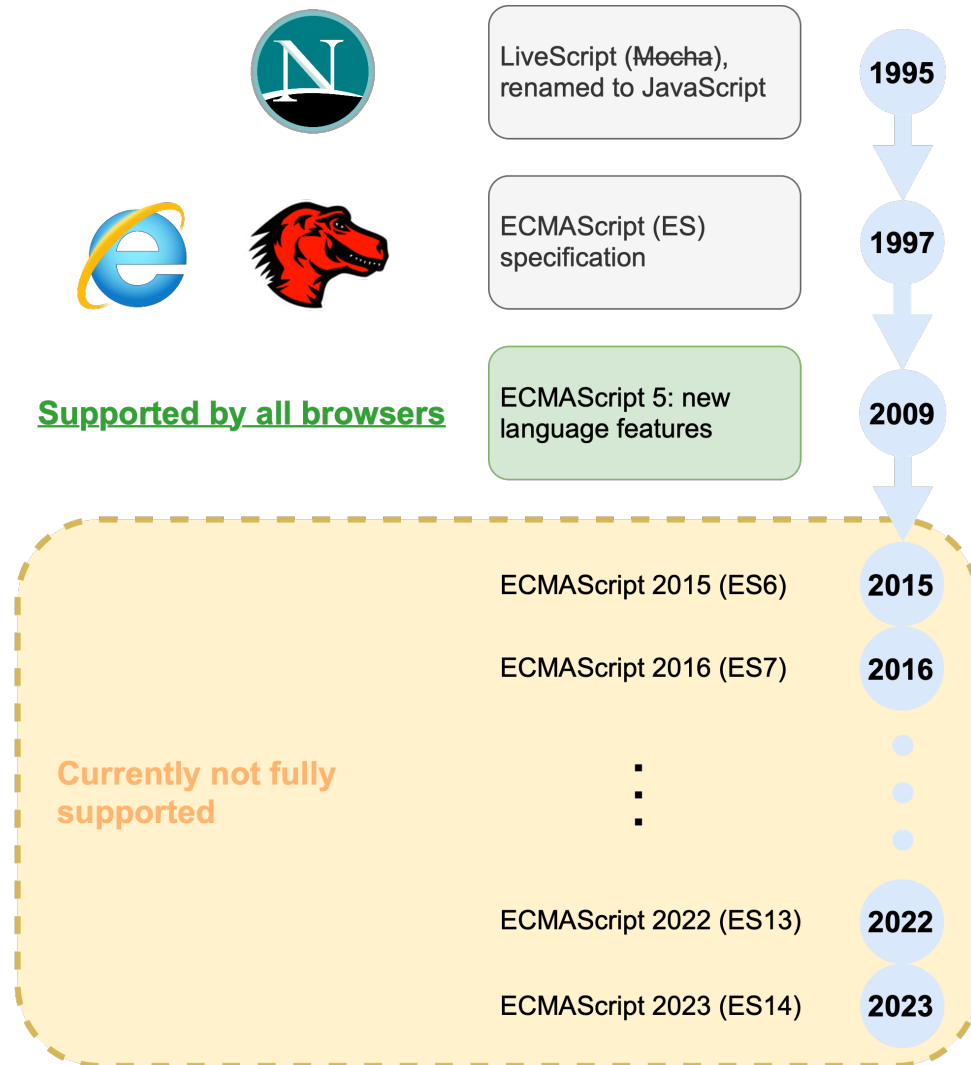
Introduzione a JavaScript moderno

Roberto Guidi

JavaScript - Riassunto caratteristiche principali

- Linguaggio di alto livello, human readable
- Interpretato
 - Viene tradotto direttamente a run-time dagli engine js (V8, SpiderMonkey, Nitro, Chakra...)
 - Run-time errors: non avendo un passaggio di compilazione è facile incorrere in errori a run-time.
- Supporto a tipi, operatori, oggetti e metodi
- Sviluppato inizialmente per aggiungere interattività alle pagine web
- Progettato per lavorare con HTML e CSS attraverso il DOM (Document Object Model)

JavaScript - Storia



JavaScript - Transpilers

- Il codice sorgente in ES6+ può essere convertito tramite un transpiler come Babel in codice ECMAScript 5, compatibile con tutti i browsers.
- In questo modo è possibile usare feature moderne del linguaggio ed avere il supporto anche per vecchi browser
- Quando tutti i browser saranno in linea con le nuove specifiche del linguaggio questo passaggio di “compilazione” potrà essere rimosso

JavaScript - Transpilers

```
//ES6+  
let courseName = 'Web Applications 2';
```

BABEL

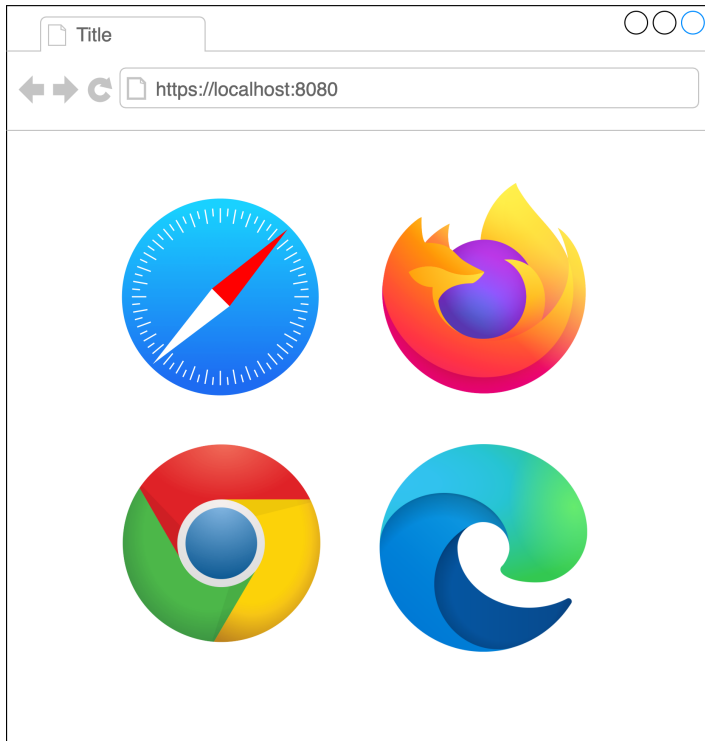
```
//ECMAScript 5 compatible  
var courseName = 'Web Applications 2';
```

JavaScript in browsers e backends

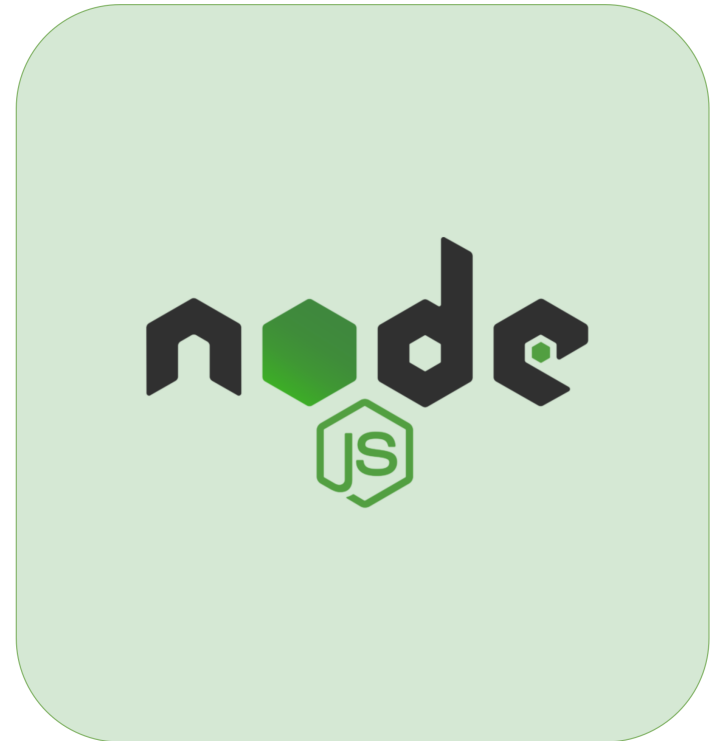
- Nonostante JavaScript sia nato come linguaggio che esegue dentro il browser, è stato portato in altri ambienti grazie a progetti come Node.js
- Questo ci permette di utilizzare JavaScript anche lato server
- Node.js è cross-platform ed è disponibile sulle maggiori piattaforme (Linux, OS X, Windows)
- Da Node.js non è possibile accedere al DOM (Document Object Model).

JavaScript in browsers e backends

Client



Server



JavaScript - Tools per lo sviluppo

- Package manager
 - come in altri linguaggi anche in js è utile utilizzare un gestore delle dipendenze simile a `maven` nel mondo java.
 - Un tool di questo tipo ci permette di gestire in maniera semplice le dipendenze dei nostri progetti (librerie, componenti aggiuntivi,...)
 - Spesso non è necessario reinventare la ruota ma possono essere sfruttati componenti messi a disposizione dalla community open source.
 - Tra i più famosi nel mondo js: `npm` (installato di default assieme a `node`) e `yarn`
- Server HTTP locale
 - permette di servire al browser i file necessari per eseguire il nostro progetto
 - live reload

JavaScript - Tools per lo sviluppo

- Transpiler
 - Tool per la conversione automatica da ES2015+ a ES5 eseguibile da tutti i browser
 - È possibile configurare un transpiler in modo che converta i sorgenti ad ogni salvataggio di file.
- Module bundler
 - Permette di combinare in un singolo file, adatto per il deployment, i sorgenti e le eventuali dipendenze esterne.
- Linter
 - Strumento che analizza il codice, direttamente all'interno dell'IDE e che aiuta a rilevare e mettere in evidenza eventuali errori.
- Test framework
 - Tool per la scrittura e l'esecuzione di test di unità o end-to-end.

JavaScript - Tools per lo sviluppo, alcuni esempi

Package management tools



Test frameworks



Module bundlers / dev servers



Linters



Transpilers



JavaScript - gestione dipendenze con nodejs e npm

- Grazie a nodejs è possibile eseguire codice javascript all'esterno del browser
- Quando installiamo nodejs, viene installato di default anche il suo package manager npm
- Utilizzando nodejs ed npm è possibile specificare i pacchetti necessari allo sviluppo delle nostre applicazioni
- La lista delle dipendenze con le relative versioni viene gestita all'interno di un file chiamato `package.json`. (questo file ha la stessa funzione del file `pom.xml` che viene usato con maven).

JavaScript - Gestione dipendenze con nodejs e npm

- È possibile creare un file `package.json` vuoto per un nuovo progetto usando il comando da terminale:

```
npm init
```

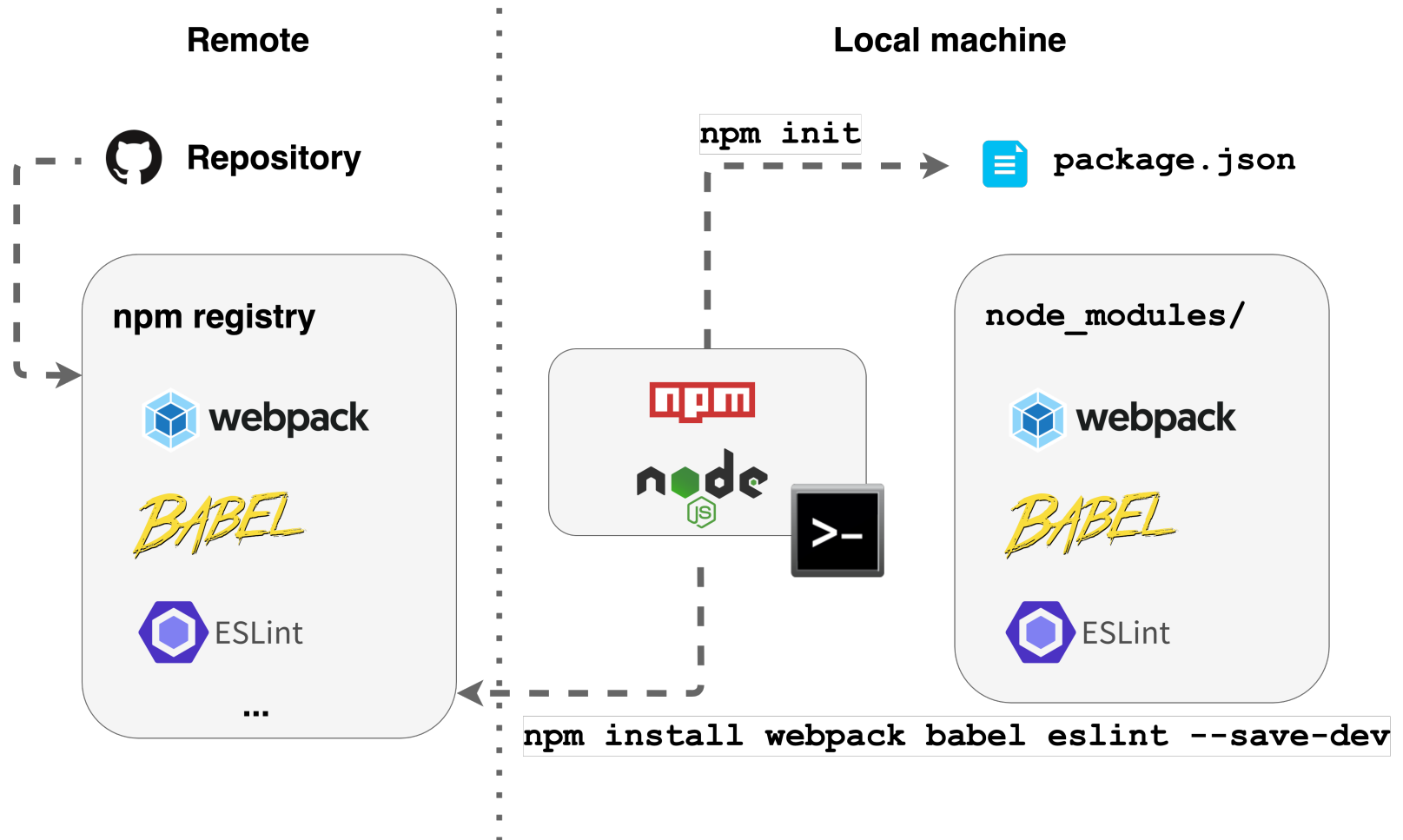
- Utilizzando in seguito il comando:

```
npm install <nome_pacchetto>
```

è possibile aggiungere una dipendenza al progetto. Essa verrà scaricata da un repository pubblico (es. github) ed installata sulla nostra macchina. Questo comando aggiunge anche al file `package.json` la dipendenza. Per le dipendenze puramente di sviluppo è possibile specificare l'opzione `--save-dev`

- Tutti i pacchetti scaricati vengono salvati all'interno della cartella `node_modules/` presente allo stesso livello del file `package.json`

JavaScript - Gestione dipendenze con nodejs e npm



Setup nodejs

- Nelle prossime lezioni utilizzeremo `nodejs` per gestire le dipendenze ed eseguire i nostri progetti JavaScript.
- Gli installer per la versione corrente di `nodejs` possono essere scaricati al seguente link:

<https://nodejs.org/en/download/>

- Installate la versione LTS per la vostra piattaforma.

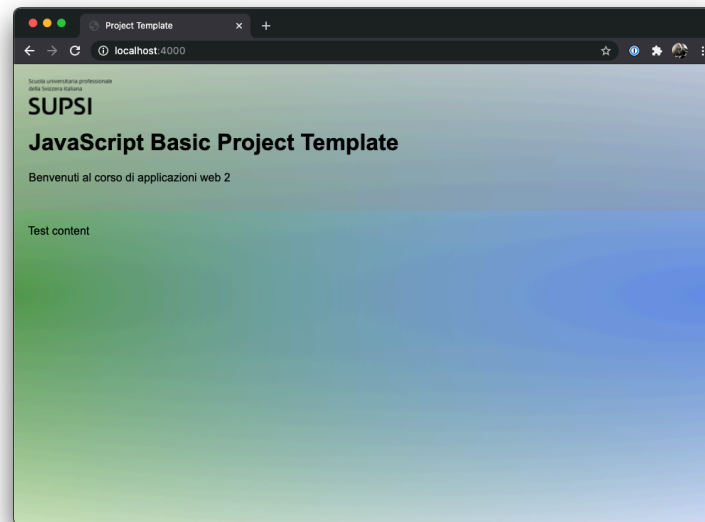
Setup progetto template

- Durante il corso utilizzeremo, al fine di sperimentare i concetti relativi alle web applications in javascript, un progetto template sul quale lavorare.
- Il progetto template è stato preparato in modo da specificare le dipendenze necessarie per `babel`, `webpack` e `eslint`, già configurate per un utilizzo di base.
- Il progetto è scaricabile al seguente link:

<https://gitlab-edu.supsi.ch/dti-isin/roberto.guidi/didattica/info/webapp2/web-app-2-basic-template.git>

Setup progetto template

- Una volta installato `nodejs` e scaricato il progetto template da Gitlab, aprite un terminale e dirigetevi alla cartella root del progetto
- Da qui eseguite il comando `npm install` per installare le dipendenze
- In seguito eseguite il comando `npm run start`
- Aprite da un browser l'indirizzo **localhost:4000**
- Qualora il progetto funzioni correttamente dovreste visualizzare la pagina seguente



Fonti e Link Utili

Fonti

- Todd Motto - UltimateCourses, JavaScript Basics: <https://ultimatecourses.com/courses/javascript>
- David Flanagan - JavaScript The Definitive Guide, Master the World's Most-Used Programming Language - O'Reilly Media (2020)
- Eric Elliot - Programming JavaScript Applications - O'Reilly Media

Link Utili:

- nodejs: <https://nodejs.org/en/>
- babel: <https://babeljs.io>
- eslint: <https://eslint.org>
- webpack: <https://webpack.js.org>