



#### Chi Sono

#### Software Engineer, ESRI & Azure Specialist Divisione GIS





Consulenza di management e Information Technology sui temi di Marketing, Vendite e Retailing

www.valuelab.it







#### Topic take aways

- Focus: perchè TypeScript e perchè migrare/integrare una soluzione JavaScript esistente a TypeScript
- Prerequisito: conoscenza basilare di TypeScript: sintassi, costrutti
- **Opzionale**: per comprendere i concetti fondamentali, non è necessario conoscere in dettaglio ogni strumento utilizzato nello stack tecnologico delle demo (riferimenti in fondo alla presentazione)
- Goodies: Struttura real-world della soluzione demo

### Nelle puntate precedenti

- CDays 2012: JavaScript avanzato: sfruttare al massimo il web
- CDays 2013: Usare Knockout.js
- CDays 2014: Single Page Applications: Come? Cosa? Perchè?
- CDays 2015: Code quality e test automatizzati con JavaScript

Disponibili sul sito dei Community Days: http://www.communitydays.it/events/





#### Follow me live

**DEMO** 

http://github.com/robymes/cdays16

**SLIDE** 

http://www.slideshare.net/RobertoMessora/da-javascript-a-typescript

### Agenda

- Chi siamo, da dove veniamo, dove andiamo: JavaScript
- TypeScript vive e lotta con noi
- Pronti, partenza, JavaScript
- Mamma, devo integrare TypeScript

## Chi siamo, da dove veniamo: JavaScript

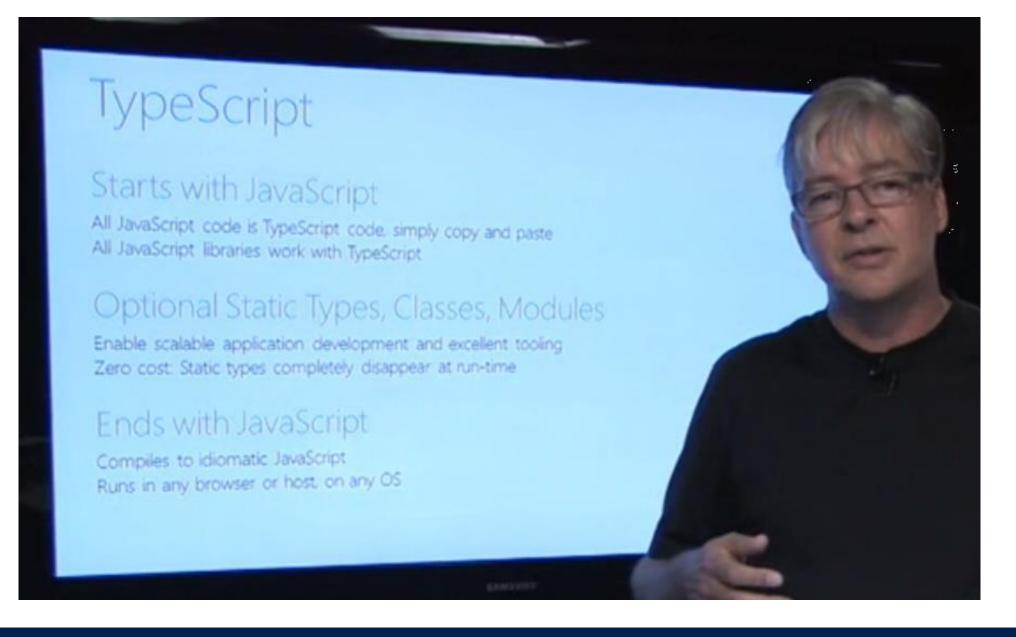
"JavaScript (in 1996) had to 'look like Java' only less so, be Java's dumb kid brother or boy-hostage sidekick. Plus, I had to be done in ten days or something worse than JS would have happened."

"JavaScript (nel 1996) doveva 'sembrare Java', solo un po' meno, essere il suo fratellino imbranato o il suo partner inesperto. In più dovevo essere pronto in dieci giorni o sarebbe accaduto qualcosa di peggio di JavaScript stesso."

Brendan Eich, ideatore di JavaScript - 2010

# TypeScript vive e lotta con noi (dal 2012)

- Linguaggio e strumenti per sviluppare applicazioni complesse che scalano in grandi team di sviluppo ("JavaScript that scales")
  - https://www.typescriptlang.org/
  - Installazione (ultima versione 2.0.3): npm install -g typescript
  - Transpiling verso JavaScript: tsc helloworld.ts → helloworld.js
  - Open source
  - Inizia e finisce con JavaScript
  - Supporta le ultime versioni di JavaScript ed anche alcune future evoluzioni (async/await, decorators)





1 Ottobre 2012

Anders Hejlsberg
Introducing TypeScript

Channel 9

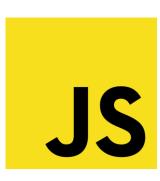
http://bit.ly/2dqJ3NN





# "Ma io ho un'applicazione JavaScript"

- Bello TypeScript, ma la mia azienda ha investito tempo, formazione, personale e denaro per sviluppare un'applicazione in JavaScript
- Dobbiamo migrare l'intera codebase? **No.**
- Quanto costa? Il tempo di imparare il linguaggio.
- Ci vogliono nuove competenze? No (anzi...).



# Stack tecnologico sviluppo JavaScript



- npm: Node.JS Package Manager
  - package.json



- Bower: Repository di librerie di terze parti
  - bower.json



- Gulp: Task manager
  - gulpfile.js

# Stack tecnologico sviluppo JavaScript



ARMA • Karma: Test runner

karma.js



• Jasmine: Unit testing



- Eslint: Linter, code quality, code standards
  - .eslintrc.json



# Stack tecnologico sviluppo TypeScript



npm, Bower, Gulp, Karma, Jasmine, Eslint



- **tsc:** Compilatore TypeScript
  - tsconfig.json



- Tslint: Linter, code quality, code standards
  - tslint.json



### TypeScript Declaration File

 Scrittura manuale: generazione di un declaration file per l'uso del JavaScript già sviluppato nel codice TypeScript



- **tsd:** DefinetelyTyped (http://definitelytyped.org/), repository **github** che raccoglie i **declaration file** delle librerie JavaScript più usate
  - tsd.json



#### Riferimenti

- Node.JS https://nodejs.org/
- Bower https://bower.io/
- Gulp.JS http://gulpjs.com/
- Karma http://karma-runner.github.io/1.0/index.html
- Jasmine http://jasmine.github.io/
- Eslint http://eslint.org/
- TypeScript https://www.typescriptlang.org/
- Tslint https://palantir.github.io/tslint/
- DefinetelyTyped http://definitelytyped.org/



