TC-Web
Technology Consulting

Angular versioni

Angular è un **framework front-end sviluppato da Google** che permette di creare applicazioni web moderne SPA(Single Page Application), scalabili e manutenibili. Si basa su **TypeScript (** https://www.typescriptlang.org/) e offre strumenti integrati per gestione dello stato, routing, comunicazione con API e sviluppo modulare.

Angular js 1--1.7.x (nato nel 2010 “morto” gennaio 2022) https://angularjs.org/

Versioni (https://angular.dev/)
Angular 2+ 2(22–23 ottobre 2014)
,4,5,6,7,..14 ….20 (agosto 2025)

Dalla versione 14 abbandono dei moduli
Nuovo logo e sito dalla versione 18

https://v17.angular.io/guide/releases
https://angular.dev/reference/versions

Angular perchè ?

TC-Web
Technology Consulting

I limiti del JavaScript vanilla e jQuery

Quando le applicazioni web erano semplici - qualche pagina statica con un po' di interattività - JavaScript puro e jQuery erano perfetti. Ma oggi le applicazioni web sono diventate **vere e proprie applicazioni desktop** che girano nel browser, e qui emergono i problemi:

Gestione dello stato caotica Con JavaScript vanilla, lo stato dell'applicazione è sparso ovunque: variabili globali, attributi DOM, localStorage. Non hai un posto centrale dove guardare per capire "cosa sta succedendo nella mia app right now".

Man mano che l'applicazione cresce, strutturare il tuo codice in modo pulito, gestibile e, soprattutto, testabile, diventa sempre più difficile.

Usando una struttura come Angular, tutto diventa più facile.

Problema ? C'è una curva di apprendimento :-)
Moduli, Componenti, Service, Direttive, Pipe, Life Cycle Hook, etc.

Angular o React o Vue o..

https://www.browserstack.com/guide/angular-vs-react-vs-vue

Angular è migliore di React / Vue.js?

Ognuno di questi framework/librerie ha punti di forza e punti deboli

Scelte legate al tipo di progetto in base alle proprie esigenze

Angular CLI

Angular cli

Angular cli permette di creare velocemente tutti gli elementi utilizzati in Angular (moduli, componenti, servizi, direttive etc.)

Creazione di componenti

Creare un componente con template e css inline e senza file di test senza creare una cartella

ng generate component nome-componente --inline-template --inline-style --skip-tests --flat
(ng g c nome-componente -t -s --skip-tests --flat)

Per tutte le opzioni su ng generate

ng g c --help

Installare Angular cli

Opzione 1 – Usare Angular CLI globale
https://angular.dev/tools/cli/setup-local#
npm install -g @angular/cli
ng new nome-progetto

Vantaggi:

- Hai sempre il comando ng disponibile ovunque.
- Se lavori spesso con Angular, è molto comodo.

Svantaggi:

- Potresti avere più versioni di Angular CLI sul tuo PC rispetto a quella del progetto (es: il progetto usa Angular 18 ma tu hai installato CLI 17 globalmente).
- Devi ricordarti di aggiornare la CLI globale.

Installare Angular cli

Opzione 2 – Usare `npx` (senza installazione globale)

- `npx @angular/cli new nome-progetto`

Vantaggi:

- Non devi installare nulla a livello globale.
- Usa sempre la versione più aggiornata di Angular CLI dal registro npm.
- Eviti problemi di compatibilità con versioni diverse della CLI.

Svantaggi:

- Ogni volta che crei un nuovo progetto deve scaricare la CLI (qualche secondo in più).
- Se crei spesso progetti, può diventare un po' ridondante.

Installare Angular cli

TC-Web Technology Consulting

Opzione 3 – Usare pnpm

pnpm mantiene uno **store globale** contenente ogni pacchetto una sola volta e nei progetti crea **hard link**/symlink. Risultato: più progetti, ma i file delle dipendenze sono condivisi a livello di disco.

Setup rapido

```
# Abilita corepack (gestisce pnpm/yarn)
corepack enable

# Attiva l'ultima versione di pnpm
corepack prepare pnpm@latest --activate

# Crea un nuovo progetto Angular usando pnpm§
npx @angular/cli new mia-app --packageManager=pnpm

# oppure, se hai già ng globale:
ng new mia-app --packageManager=pnpm
#Prog. presenti: rmdir /s node_modules e del package-lock.json
pnpm install
```

Moduli vs standalone components

TC-Web Technology Consulting

npx @angular/cli@14 new nome-progetto

(in tsconfig.json `"compilerOptions": {  "skipLibCheck": true,)`

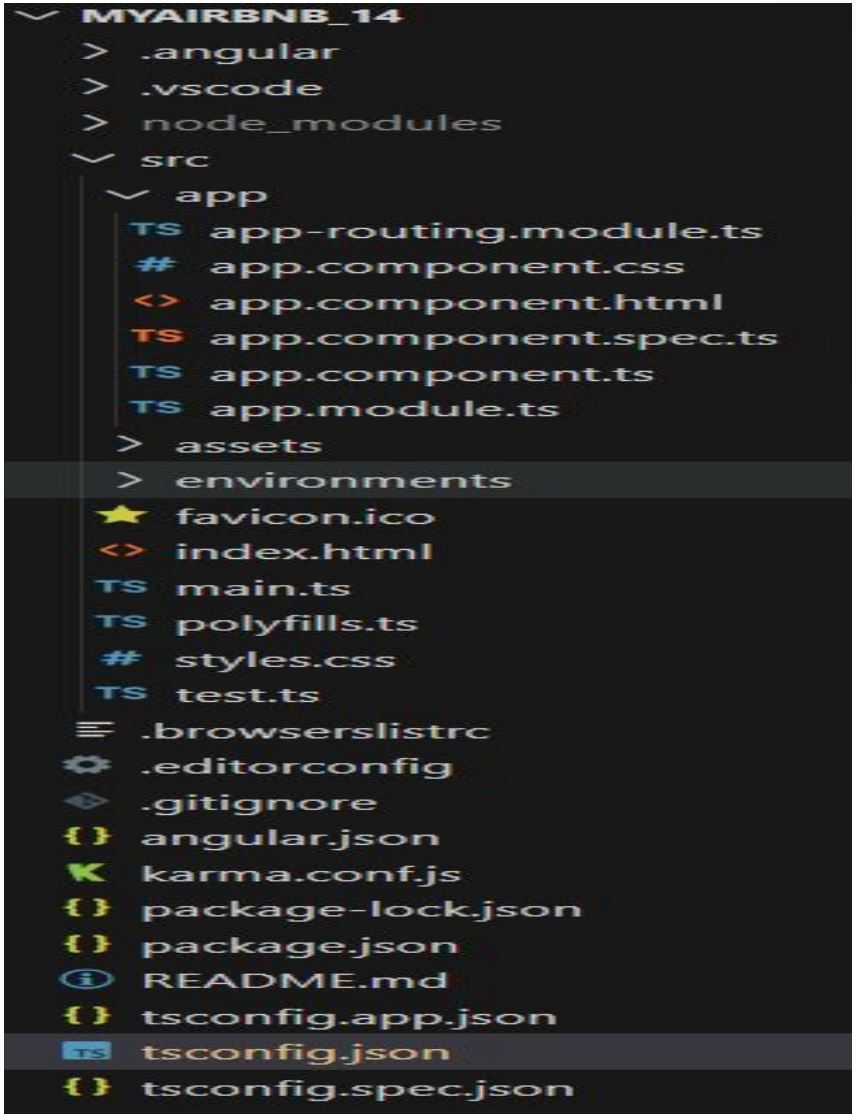
Storicamente Angular ha sempre usato i **moduli** (`NgModule`) per organizzare un'applicazione. Dalla versione **14** sono arrivati i **componenti standalone**

Ogni app aveva almeno un `AppModule`:

```
@NgModule({
  declarations: [AppComponent, HomeComponent],
  imports: [BrowserModule, FormsModule],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

- I componenti, direttive e pipe devono essere **dichiarati in un modulo**.
- Per usare un altro modulo (es. `FormsModule`), va importato in `imports`.
- Struttura più verbosa ma ordinata in progetti grandi.

Progetto Angular con moduli



npm e package.json

npm utilizza un file chiamato **package.json** per ottenere un elenco dei pacchetti software necessari per un progetto.

scripts Elenco di script che possono essere eseguiti dalla riga di comando. (es npm start → ng serve)

dependencies Elenco di pacchetti NPM su cui l'applicazione Web si basa per l'esecuzione. Ogni pacchetto è specificato con un numero di versione
devDependencies Elenco di pacchetti NPM a cui si fa affidamento per lo sviluppo ma che non sono richiesti dall'applicazione una volta che è stato distribuito.

Esempio: pacchetti che compilano i file TypeScript,

Nota: Se si modifica il package.json occorre poi fare **npm install**

Aggiungere librerie/funzionalità

Si usa npm di node (o pnpm) , ad esempio

npm install bootstrap@4.0.0 font-awesome@4.7.0

Se si tratta come in questo caso di librerie CSS occorre anche modificare il file angular.json

```
  "styles": [
    "node_modules/bootstrap/dist/css/bootstrap.min.css",
  "node_modules/font-awesome/css/font-awesome.min.css", "styles.css"
  ],
```

Moduli vs standalone components

```
ng new airbnb_20
(npm install @rollup/rollup-win32-x64-msvc --save-dev
npm install @esbuild/win32-x64 )
```

componenti standalone

GRAZIE A TUTTI…..

TC-Web
Technology Consulting

Via Schiaparelli 14 - 10148 Torino
https://www.tc-web.it
Mail: franco.grivet@tc-web.it