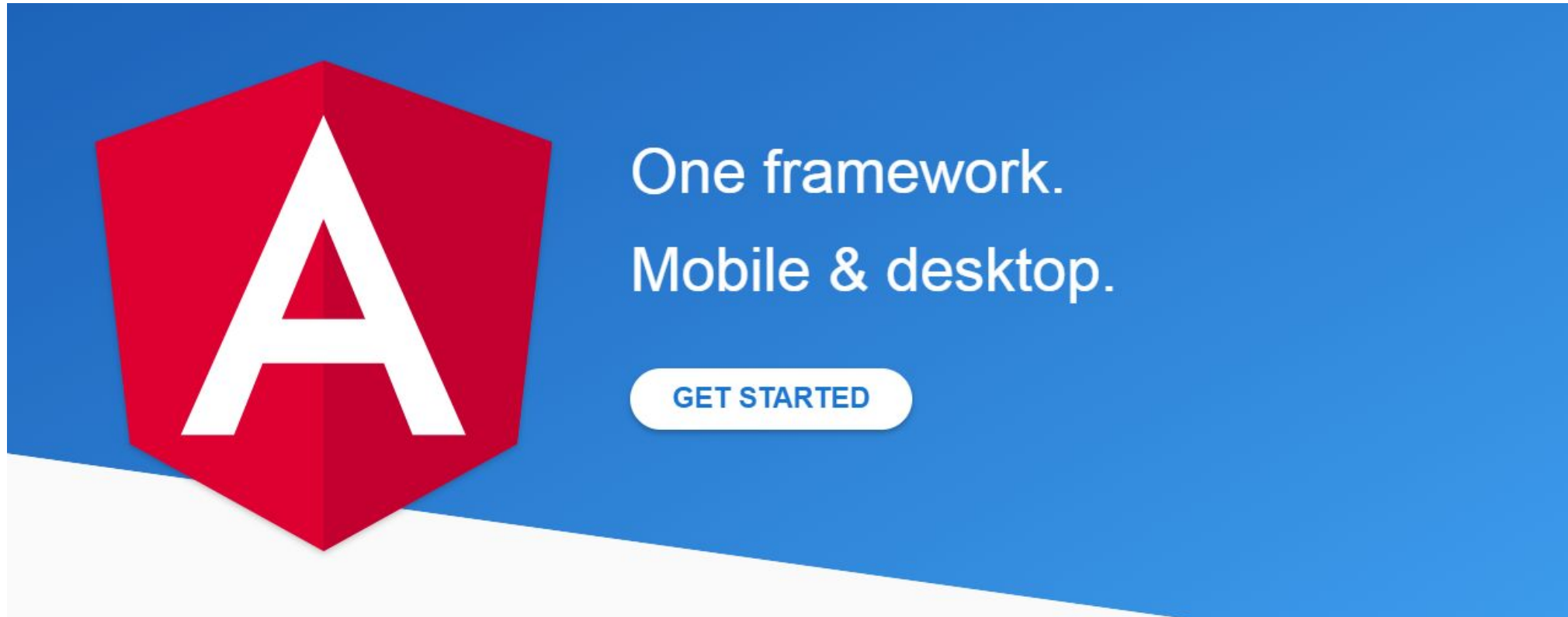


# Angular

Веб-технології

# Що таке Angular

<https://angular.io/>



# Latest version

Остання версія Angular - Angular [7.2.13](https://github.com/angular/angular) вийшла 12 квітня 2019 року. Офіційний репозиторій фреймворка на гітхабе: <https://github.com/angular/angular>. Там ви можете знайти самі вихідні файли, а також деяку додаткову інформацію.

# Головні аспекти

- Angular надає таку функціональність, як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизація і так далі.
- Однією з ключових особливостей Angular є те, що він використовує в якості мови програмування TypeScript. Тому перед початком роботи рекомендується ознайомитися з основами цієї мови, про які можна прочитати тут.
- Але ми не обмежені мовою TypeScript. При бажанні можемо писати програми на Angular за допомогою таких мов як Dart або JavaScript. Однак TypeScript все таки є основною мовою для Angular.

# Початок роботи

- Для роботи з Angular необхідно встановити сервер Node.js і пакетний менеджер npm, якщо вони відсутні на робочій машині.

[Node.js](https://nodejs.org/en/) (https://nodejs.org/en/)

Перевірте, що використовуєте, принаймні, Node 6.9.x та npm 3.x.x, запустивши `node -v` та `npm -v` у вікні терміналу / консолі. Старіші версії виробляють помилки, але новіші версії чудово.

# Використання Angular CLI

- Встановіть глобально Angular CLI

```
npm install -g @angular/cli
```

- Створіть новий проект

```
ng new my-app
```

- Запустіть сервер та додаток

```
cd my-app
```

```
ng serve --open
```

EXPLORER

- OPEN EDITORS
- MY-APP
  - e2e
  - node\_modules
  - src
    - app
      - app.component.css
      - app.component.html
      - app.component.spec.ts
      - app.component.ts
      - app.module.ts
    - assets
    - environments
    - favicon.ico
    - index.html
    - main.ts
    - polyfills.ts
    - styles.css
    - test.ts
    - tsconfig.app.json
    - tsconfig.spec.json
    - typings.d.ts
    - .angular-cli.json
    - .editorconfig
    - .gitignore
    - karma.conf.js
    - package-lock.json
    - package.json
    - protractor.conf.js



- Show All Commands    Ctrl+Shift+P
- Go to File            Ctrl+P
- Find in Files        Ctrl+Shift+F
- Start Debugging      F5
- Toggle Terminal     Ctrl+`

# Створення проекту власноруч

- Для цього визначимо на жорсткому диску папку програми. Шлях вона буде називатися helloapp. У цій папці створимо новий файл **package.json** наступного змісту

```
{
  "name": "helloapp",
  "version": "1.0.0",
  "description": "First Angular 5 Project",
  "author": "Eugene Popov <metanit.com>",
  "scripts": {
    "dev": "webpack-dev-server --hot --open",
    "build": "webpack"
  },
  "dependencies": {
    "@angular/common": "~5.0.0",
    "@angular/compiler": "~5.0.0",
    "@angular/core": "~5.0.0",
    "@angular/forms": "~5.0.0",
    "@angular/platform-browser": "~5.0.0",
    "@angular/platform-browser-dynamic": "~5.0.0",
    "@angular/router": "~5.0.0",
    "core-js": "^2.4.1",
    "rxjs": "^5.5.2",
    "zone.js": "^0.8.14"
  },
  "devDependencies": {
    "@types/node": "^8.0.47",
    "typescript": "^2.6.0",
    "webpack": "^3.6.0",
    "webpack-dev-server": "^2.9.1",
    "angular2-template-loader": "^0.6.2",
    "awesome-typescript-loader": "^3.3.0",
    "uglifyjs-webpack-plugin": "^1.0.1"
  }
}
```



- Відкриваємо командний рядок (термінал) і перейдемо в ній до папки проекту за допомогою команди `cd`:

```
C:\WINDOWS\system32>cd C:\angular2\helloapp
```

- І потім виконаємо команду `npm install`, яка встановить всі необхідні модулі

```
C:\angular2\helloapp>npm install
```

- Створимо в папці проекту підпапку, яку назвемо `src` - вона буде містити всі вихідні файли. І далі в папці `src` створимо підкаталог `app`

# Створення компонента Angular

Компоненти представляють основні будівельні блоки додатки Angular 2. Кожна програма Angular має як мінімум один компонент. Тому створимо в папці src / app новий файл, який назвемо app.component.ts і в якому визначимо наступний код компонента

```
import { Component } from '@angular/core';

@Component({
  selector: 'my-app',
  template: `<label>Введіть ім'я:</label>
             <input [(ngModel)]="name" placeholder="name">
             <h1>Добро пожаловать {{name}}!</h1>`
})
export class AppComponent {
  name= "";
}
```

# Створення модулю додатка

Додаток Angular складається з модулів. Модульна структура дозволяє легко довантажувати і задіяти тільки ті модулі, які безпосередньо необхідні. І кожен додаток має як мінімум один кореневий модуль. Тому створимо в папці `src / app` новий файл, який назвемо `app.module.ts` наступного змісту

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { AppComponent } from './app.component';
@NgModule({
  imports: [ BrowserModule, FormsModule ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

# Запуск додатка

Тепер нам треба вказати Angular, як запускати наш додаток. Для цього створимо в папці src (на рівень вище, ніж розташовані файли app.component.ts і app.module.ts) файл main.ts наступного змісту:

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';  
import { AppModule } from './app/app.module';  
const platform = platformBrowserDynamic();  
platform.bootstrapModule(AppModule);
```

- Також в папці src визначимо ще один файл, який назвемо polyfills.ts з наступним кодом

```
import 'core-js/es6';  
// для підтримки Reflect Api  
import 'core-js/es7/reflect';  
// zone використовується angular  
import 'zone.js/dist/zone';
```

# Визначення конфігурації

Оскільки для визначення коду програми застосовується мова TypeScript, тому також створимо в кореневій папці проекту новий файл `tsconfig.json`

```
{  
  "compilerOptions": {  
    "target": "es5",  
    "module": "es2015",  
    "moduleResolution": "node",  
    "sourceMap": true,  
    "emitDecoratorMetadata": true,  
    "experimentalDecorators": true,  
    "lib": [ "es2015", "dom" ],  
    "noImplicitAny": true,  
    "suppressImplicitAnyIndexErrors": true,  
    "typeRoots": [  
      "node_modules/@types/"  
    ]  
  },  
  "exclude": [  
    "node_modules"  
  ]  
}
```

Оскільки наш додаток буде розбито на кілька окремих файлів, то для їх складання будемо використовувати популярний складальник webpack. Тому також визначимо в кореневій папці проекту ще один файл webpack.config.js:

```
var path = require('path');
var webpack = require('webpack');
var UglifyJSPlugin = require('uglifyjs-webpack-plugin'); // плагин минимизации
module.exports = {
  entry: {
    'polyfills': './src/polyfills.ts',
    'app': './src/main.ts'
  },
  output: {
    path: path.resolve(__dirname, './public'), // путь к каталогу выходных файлов - папка public
    publicPath: '/public/',
    filename: "[name].js" // название создаваемого файла
  },
  resolve: {
    extensions: ['.ts', '.js']
  },
  module: {
    rules: [ //загрузчик для ts
      {
        test: /\.ts$/, // определяем тип файлов
        use: [
          {
            loader: 'awesome-typescript-loader',
            options: { configFile: path.resolve(__dirname, 'tsconfig.json') }
          },
          'angular2-template-loader'
        ]
      }
    ]
  },
  plugins: [
    new webpack.ContextReplacementPlugin(
      /angular(\\|\/)core/,
      path.resolve(__dirname, 'src'), // каталог с исходными файлами
      {} // карта маршрутов
    ),
    new webpack.optimize.CommonsChunkPlugin({
      name: ['app', 'polyfills']
    }),
    new UglifyJSPlugin()
  ]
}
```

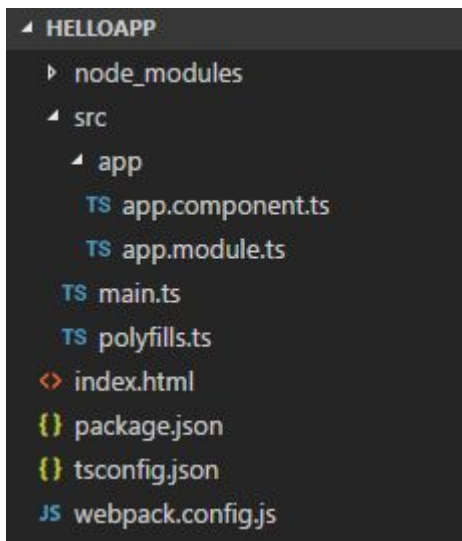
# Створення головної сторінки

Визначимо головну сторінку index.html в кореневій папці проекту

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Hello Angular 5</title>
</head>
<body>
  <my-app>Завантаження...</my-app>
  <script src="public/polyfills.js"></script>
  <script src="public/app.js"></script>
</body>
</html>
```



# Збірка та запуск проекту



```
C:\angular2\helloapp>npm run dev
```

- <https://www.sitepoint.com/angular-2-tutorial/>