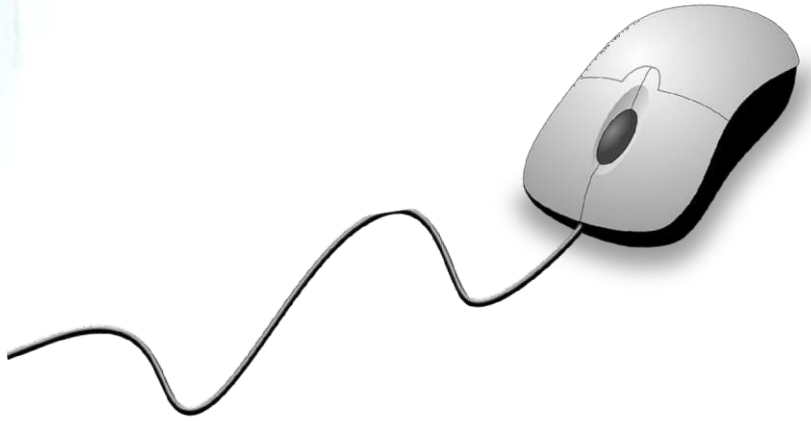
공개SW솔루션 설치 & 활용 가이드

응용SW > 콘텐츠 배포



ANGULAR 제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide

OPENUP

오픈소스 소프트웨어 통합지원센터

Open Source Software Support Center

CONTENTS

1. 개요

소개	• 사용자 인터페이스를 만들기 위한 **진보적인 자바스크립트 프론트엔드 프레임워크** • 점진적으로 채택할 수 있게 설계된 프레임워크 • 프론트 개발에 필요한 모든 요소를 갖추고 있는 프레임워크		
주요기능	• HTML 템플릿 문법 사용 • Module 단위로 어플리케이션 코드 인식 • Angular는 컴포넌트 기반 라이브러리로 컴포넌트 기반 코드를 캡슐화 하는 것에 도움이 됨		
대분류	• 응용SW	**소분류**	• 콘텐츠 배포
라이선스형태	The MIT License (MIT)	**사전설치 솔루션**	Node.js (NPM을 사용할 경우)
		버전	v15.1.0 (2022년 12월 기준)
특징	• 완전한 프레임 워크로, 프로젝트의 생성, 테스트, 빌드, 배포를 위한 모든 기능 제공 • 모듈과 컴포넌트 기반으로 동작 • 라우팅, HTTP, Form 등의 기능 라이브러리는 모두 포함시켜 자체적으로 제공 • HTML과 CSS코드를 재사용 가능하게 함 • 공식 문서의 품질이 좋음		
개발회사/커뮤니티	Google / https://forum.vuejs.org/		
공식 홈페이지	https://vuejs.org/		

2. 기능요약

• Angular 주요 기능

데코레이터 및 메타데이터	Angular는 프레임워크이므로 특정 클래스가 일반 클래스인지, 프레임워크의 어느 부분을 담당하는 클래스인지 표시할 필요가 있음 Decorator는 관련 역할을 이르는 용어로, Decorator에 데이터를 추가하여 세부적으로 설정할 경우 해당 데이터를 Metadata라고 함
컴포넌트	Angular 인스턴스를 컴포넌트로 등록하여 재사용할 수 있는 조합 가능한 컴포넌트로 만들 수 있으며 컴포넌트 등록은 전역 또는 지역으로 할 수 있음
모듈	Angular의 경우, Module 단위로 어플리케이션 코드를 인식하므로 모든 Angular 어플리케이션은 반드시 하나 이상의 Module을 가지게 됨

3. 실행환경

- Angular CLI를 사용하는 개발 환경 - Node.js 또는 yarn이 설치되어야 함
  - Windows
  - Linux
  - macOS 등
- Angular 지원 브라우저
  - Safari 6 이상
  - Chrome 23 이상
  - Firefox 21 이상
  - Edge 12 이상
  - IE 9 이상

4. 설치 및 실행

세부 목차

1. Angular CLI로 설치 및 실행 (Node.js와 npm)
2. CDN을 직접 `<script>`에 추가

- Angular 애플리케이션을 만들고 싶다면 4.1. 방법을,

  앱을 바로 실행시켜보고 싶다면 4.3. 방법대로 설치 및 실행하는것을 권장

  본 가이드에서는 코드 에디터로 VS Code를 사용하지만, 다른 텍스트 에디터를 사용해도 편집 가능

  –VS Code 다운로드: https://code.visualstudio.com/download

4. 설치 및 실행

1. Vue CLI로 설치 및 실행 (Node.js와 npm) (1/4)

- OS version: Windows 10
- Node.js version: v10.15.0
- Vue CLI를 이용하기 위해서 Node.js와 yarn 우선 설치 진행
- Windows의 경우 https://nodejs.org/ko/ 에서 Node.js 설치
- Ubuntu의 경우 아래 순서대로 bash에서 진행

```
$ sudo apt-get install curl
$ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
$ sudo apt-get install nodejs
```

- Node.js와 NPM 버전 확인 – Windows의 경우 cmd, Linux의 경우 bash에서

```
$ node -v
$ npm -v
```

– 버전 숫자가 나오면 통과

```
PS C:\> node --version
v10.15.0
PS C:\> npm --version
5.3.0
```

4. 설치 및 실행

2. Angular CLI로 설치 및 실행 (Node.js와 npm) (2/4)

1. Angular CLI 설치

- mac
```
$ sudo npm install -g @angular/cli
```
- window
```
$ npm install -g @angular/cli
```

2. 버전 확인

-
```
$ ng version
```
-
```
$ ng v
```

설치가 완료 시 해당 표시

3. 프로젝트 생성

- `ng new 프로젝트명`
- 화살표키를 이용해 아래 설정대로 생성

4. 해당 디렉토리로 이동

- `cd 프로젝트명`

```
soyun@soyun-Latitude-3520:~$ ng new hello-angular
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE hello-angular/README.md (1066 bytes)
CREATE hello-angular/.editorconfig (274 bytes)
CREATE hello-angular/.gitignore (548 bytes)
CREATE hello-angular/angular.json (2735 bytes)
CREATE hello-angular/package.json (1044 bytes)
CREATE hello-angular/tsconfig.json (901 bytes)
CREATE hello-angular/tsconfig.app.json (263 bytes)
CREATE hello-angular/tsconfig.spec.json (273 bytes)
```

4. 설치 및 실행

4.1 Vue CLI로 설치 및 실행 (Node.js와 npm) (3/4)

1. package.json의 scripts 확인

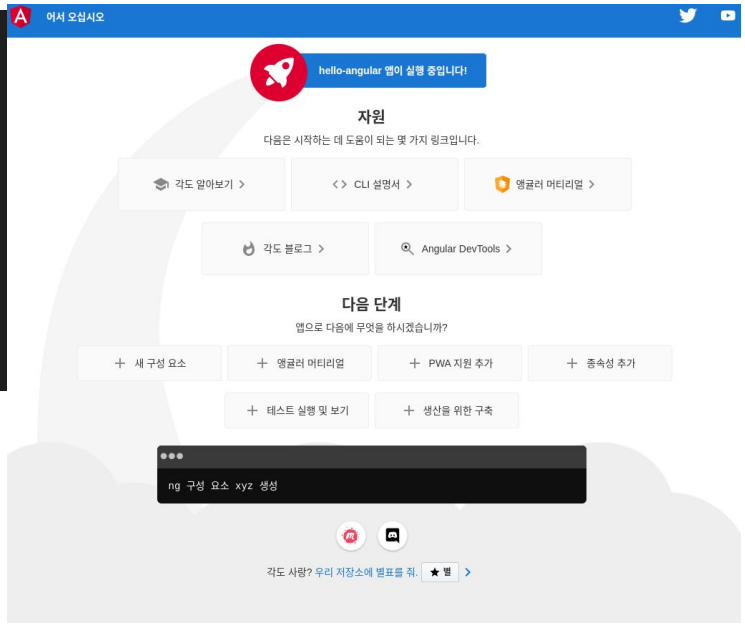
```
{
  "name": "hello-angular",
  "version": "0.0.0",
  ▷ 디버그
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
```

2. package.json에 있는 명령어를 사용해서 프로젝트를 로컬에서 실행
   - `yarn ng serve`

```
Effective status: enabled
✔ Browser application bundle generation complete.

Initial Chunk Files   | Names      |  Raw Size
vendor.js             | vendor     |   2.09 MB |
polyfills.js          | polyfills  | 314.28 kB |
styles.css, styles.js | styles     | 209.41 kB |
main.js               | main       |  48.11 kB |
runtime.js            | runtime    |   6.52 kB |
```

3. 콘솔에 나온 주소로 접속해서 앱이 실행 확인



4. 설치 및 실행

4.1 Vue CLI로 설치 및 실행 (Node.js와 npm) (4/4)

1. package.json에 있는 명령어를 사용해서 프로젝트를 빌드

   - `ng build`

```
✔ Browser application bundle generation complete.
✔ Copying assets complete.
✔ Index html generation complete.

Initial Chunk Files           | Names         |  Raw Size | Estimated Transfer Size
main.6113d1fbfd7fca98.js      | main          | 196.65 kB |               53.58 kB
polyfills.4ae95a29fb4c7a5c.js | polyfills     |  33.08 kB |               10.69 kB
runtime.83f036cc0cea350c.js   | runtime       | 904 bytes |              510 bytes
styles.ef46db3751d8e999.css   | styles        |   0 bytes |                      -

                              | Initial Total | 230.61 kB |               64.77 kB
```

2. 빌드된 결과물 확인

   - `ls dist`

```
soyun@soyun-Latitude-3520:~/hello-angular$ ls dist
hello-angular
```

3. 해당 빌드는 사용자 배포용으로 optimized

4. 설치 및 실행

4.2. CDN을 직접 `<script>`에 추가

1. 오른쪽과 같이 cdn.html 파일을 작성

- 해당 홈페이지에 들어간 후 script 복사

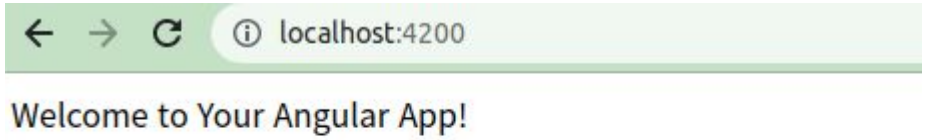
  (https://www.w3schools.com/angular/angular_intro.asp)

- 1.6.9 ver

  `<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>`

```
src > index.html > html > body > p
1  <!DOCTYPE html>
2  <html>
3  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
4  <body>
5
6    <p ng-bind="name">Welcome to Your Angular App!</p>
7
8  </body>
9  </html>
```

2. 브라우저에서 cdn.html을 열어서 확인



5. 기능소개

세부 목차

5. 기능소개

1. HTML 템플릿문법 (1/2)

- HTML 템플릿 문법을 사용 가능
- Angular 템플릿은 HTML 파서로 구문 분석이 가능한 유효한 HTML
- 데이터 바인딩을 통해서 동적으로 DOM(document object model)의 밸류를 가져오고 바꿀 수 있음
- 반응형 앱인 Angular는 앱의 상태가 변경될 때에 최소한의 DOM 조작을 통해 페이지를 변경

```
<!doctype html>
<html lang="en">
<body>
  <div>
    <h1>Hello Angular</h1>
    <p>
      For a guid and recipes on how to configur / custom check out the
      <a href="https://angular.kr/" target="_blank"></a>
    </p>
    <h3>Fruits list</h3>
    <ul>
      <li>
        Apple
      </li>
      <li>
        Grape
      </li>
    </ul>
  </div>
</body>
</html>
```

5. 기능소개

1. HTML 템플릿문법 (2/2)

- 보간법
  - 데이터를 바인딩하는 방법은 "콧수염 문법", 이중 중괄호를 사용 가능
  - `<h1>{{ name }}</h1>`에서, 실제 페이지에서는 {{ name }}부분에 name의 값이 담기게 됨.

```
<!doctype html>
<html lang="en">
<body>
  <div>
    <h1>Hello {{name}}</h1>
    <p>
      For a guid and recipes on how to configur / custom check out the
      <a href="https://angular.kr/" target="_blank"></a>
    </p>
    <h3>Fruits list</h3>
    <ul>
      <li>
        Apple
      </li>
      <li>
        Grape
      </li>
    </ul>
  </div>
</body>
</html>
```

5. 기능소개

5.2. 컴포넌트 (1/3)

- Angular는 컴포넌트를 사용가능
  - 컴포넌트: 컴포넌트는 Angular의 구성요소 중 하나로, 화면을 구성하고 관리하는 역할
- 컴포넌트 생성 방법
  - src/app 하단에서 `ng generate component` 입력 (**ng g c** 로 축약 가능)
  - 생성하고자 하는 컴포넌트 이름을 작성

```
soyun@soyun-Latitude-3520:~/hello-angular/src/app$ ng g c
? What name would you like to use for the component? hello-component
```

생성된 컴포넌트

5. 기능소개

5.2. 컴포넌트 (2/3)

- 생성된 컴포넌트 내부에 있는 hello.component.html에 내용 작성

src > app > hello > hello.component.html >

```
1   <h1>{{title}}</h1>
2   <p>This is component example</p>
```

- app.component.ts에 있는 selector를 참고하여 index.html에 작성
  (selector : 해당 컴포넌트를 사용하기 위한 명칭)

src > app > TS app.component.ts > AppComponent >

```
1   import { Component } from '@angular/core';
2
3   @Component({
4     selector: 'app-root',
5     templateUrl: './app.component.html',
6     styleUrls: ['./app.component.css']
7   })
```

src > index.html > ...

```
1   <app-root></app-root>
```

- app.component.html에서 hello.component.ts에 있는 selector를 참고하여 작성

```
1   import { Component } from '@angular/core';
2
3   @Component({
4     selector: 'app-hello',
5     templateUrl: './hello.component.html',
6     styleUrls: ['./hello.component.css']
7   })
```

https://angular.kr/guide/component-overview

5. 기능소개

5.2. 컴포넌트 (3/3)

- 컴포넌트 관련 세부 내용
- 참조: https://angular.kr/guide/component-interaction

@ViewChild()로 자식 컴포넌트 접근하기

*템플릿 지역 변수*로 자식 컴포넌트에 접근하는 것은 문법도 간단하고 이해하기 쉽습니다. 하지만 이 방식은 부모 컴포넌트의 템플릿에서만 자식 컴포넌트에 접근할 수 있기 때문에 자유롭게 활용하기에는 제한이 있습니다. 부모 컴포넌트의 *클래스*에서는 자식 컴포넌트에 접근할 수 없기 때문입니다.

*템플릿 지역 변수*를 사용하는 방식은 부모 컴포넌트의 *클래스*에서는 사용할 수 없습니다. 컴포넌트의 부모-자식 관계는 컴포넌트가 완전히 생성된 이후에 구성되기 때문입니다. 그래서 부모 컴포넌트의 클래스에서는 자식 컴포넌트의 프로퍼티를 읽거나 메소드를 실행할 수 없습니다.

부모 컴포넌트의 *클래스*에서 자식 컴포넌트에 접근하려면 자식 컴포넌트에 *ViewChild*를 사용해서 부모 컴포넌트로 ***주입(inject)*** 해야 합니다.

이 내용은 예제를 보면서 알아봅시다. 이 예제는 위에서 살펴본 카운트다운 타이머와 거의 비슷하지만, 구현 방식은 조금 다릅니다. 먼저, 자식 컴포넌트 CountdownTimerComponent 코드는 동일합니다.

> *템플릿 지역 변수*를 사용하는 방식과 *ViewChild*를 사용하는 방식은 거의 비슷합니다. 사용하려는 목적에 따라 구현 방식을 선택하면 됩니다.

5. 기능소개

5.3. Module (1 / 2)

- Angular는 *NgModule 이라는 모듈 체계로 구성

  (*NgModule : 애플리케이션 도메인이나 작업 흐름, 기능이 연관된 Angular 구성요소들을 묶어놓은 단위)

- Angular 애플리케이션에는 최상위 모듈이 반드시 존재

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],

  providers: [],
  bootstrap: [AppComponent]
})
```

5. 기능소개

5.3. Module (2 / 2)

- Module 메타 데이터 프로퍼티 및 설명
  - declarations : NgModule에 포함될 컴포넌트나 디렉티브, 파이프를 선언
  - exports : 모듈의 구성 요소를 다른 NgModule이나 컴포넌트 템플릿으로 재사용할 수 있도록 외부로 공개
  - imports : 다른 모듈에서 공개한 클래스를 지금 정의하는 NgModule에 가져올 때 사용
  - providers : NgModule 컨텍스트 안에서 사용하는 서비스 프로바이더를 지정
  - NgModule : 내부에서 사용하는 서비스는 지정된 서비스 프로바이더를 사용해서 생성되며, 필요한 경우에는 하위 계층에 사용할 서비스 프로바이더를 따로 지정 가능
  - bootstrap : 애플리케이션의 최상위 뷰로 표시될 최상위 컴포넌트를 지정

    *bootstrap 프로퍼티는 최상위 NgModule 에만 지정할 수 있음

6. 활용예제

세부 목차

6.1. 버튼클릭시 선택한 버튼의 라벨값이 나오는 예제

6. 활용예제

1. 버튼클릭시 선택한 버튼의 라벨값이 나오는 예제

- app.component.html에 다음과 같이 추가
- 버튼과 클릭시 라벨값이 나올 위치 추가

src > app > app.component.html > button

```
<h1>{{ title }}</h1>
<button (click)="click($event)">Angular</button>
<button (click)="click($event)">jQuery</button>
<button (click)="click($event)">Vue</button>
선택한 버튼 라벨 값:
<span>{{ result }}</span>
```

- app.component.ts에 다음과 같이 추가
  - {{title}}에 나올 값을 추가
  - 버튼을 click하면 실행될 함수 추가

src > app > app.component.ts > ...

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = '이벤트 바인딩 실습예제';
  result = null;
  click(event: any): void {
    this.result = event.target.innerText;
  }
}
```

- index.html에 component를 불러옴
  - app.component.ts의 selector에 있는 값을 참고 (이는 해당 컴포넌트를 사용하기 위한 명칭)

src > index.html > app-root

```
<app-root></app-root>
```

이벤트 바인딩 실습예제

Angular | jQuery | Vue 선택한 버튼 라벨 값: Angular

결과화면

7. 용어정리

용어	설 명
컴포넌트	컴포넌트는 Angular의 가장 강력한 기능 중 하나이며 기본 HTML 엘리먼트를 확장하여 재사용 가능한 코드를 캡슐화하는 데 도움이 됨
모듈	Angular의 모듈은 Angular의 컴포넌트, 디렉티브, 파이프, 서비스 등과 같이 관련이 있는 요소를 모은 하나의 단위를 의미 모듈은 다른 모듈과 결합할 수 있으며 Angular는 여러 모듈을 조합하여 하나의 애플리케이션을 구성이 가능함 또한 모듈은 다른 모듈을 import할 수 있음 Angular에서 제공하는 라이브러리 모듈이나 서드 파티 라이브러리도 import 하여 사용 가능함
템플릿	템플릿은 HTML과 Angular 고유의 템플릿 문법을 사용하여 UI의 최소 단위인 컴포넌트의 뷰를 정의하며 동적으로 변하는 데이터는 컴포넌트 클래스가 관리함 템플릿 문법의 데이터 바인딩에 의해 정적 HTML에 포함됨

Open Source Software Installation & Application Guide

오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center