https://www.tutorialspoint.com/angular7/index.htm

Angular

Egy nyílt forrású javascript keretrendszer.

Version	Released Date
Angular JS	October 2010
Angular 2.0	Sept 2016
Angular 4.0	March 2017
Angular 5.0	November 2017
Angular 6.0	May 2018
Angular 7.0	October 2018
Angular 8.0	March/April 2019
Angular 9.0	September/ October 2019

Angular update:

```
ng update @angular/cli @angular/core
```

Angular7 - Environment Setup

- Nodejs
- Npm
- Angular CLI
- IDE a forráskód írásához

Angular CLI install:

```
npm install –g @angular/cli
```

Új projekt létrehozása:

```
ng new my-dream-app // name of the project
cd my-dream-app
```

```
ng serve
```

Alapértelmezetten:

```
http://localhost:4200/
```

Megváltoztatása:

```
ng serve --host 0.0.0.0 -port 4205
```

The angular7-app/ folder has the following **folder structure**−

- **e2e/** − end to end test folder
- **node_modules/** − The npm package installed is node_modules.
- **src/** −Ahova írjuk az alkalmazás kódját. Az src/-n bekül az app/-ban vannak a projekthez szükséges fájlok.

The angular7-app/ folder has the following **file structure** −

- **angular.json** − project name, version of cli, etc.
- **.editorconfig** − config file
- **.gitignore** −
- **package.json** − The package.json: tehát hogy milyen könyvtárakat telepítettünk a node_modules mappa milyen könyvtárakat tartalmaz. Az npm install kiadásával ezek a könyvtárak feltepülnek, (a node_modeles mappába kerülnek ha nincsenek ott).

app

Ez több fájlt is tartalmaz, itt mindenképpen találunk egy html, css és ts fájlt.

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

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

@NgModule:

Declarations - A declaration-ben a komponensek referenciái tárolódnak. Az App komponens az alapértelmezett komponens, amelyet új projekt indításakor automatikusan létrehoz az angular. Új komponenseket mi magunk generálunk.

Importálás (Imports) – A modulok-at importálni kell. Jelenleg a BrowserModule az import része, amelyet @ angular / platform-browser segítségével importáltunk. Van még egy routing modul is, ez az AppRoutingModule.

Providers – Itt vannak a service-ek referenciái.

Bootstrap - Ez a létrehozandó alapértelmezett komponensre mutat, azaz az AppComponentre.

app.component.html

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1>Welcome to {{ title }}!</h1>
   <img width = "300" alt = "Angular Logo"
   src =
"
cvMjAwMC9zdmciIHZp

ZXdCb3g9IjAgMCAyNTAgMjUwIj4KICAgIDxwYXRoIGZpbGw9IiNERDAwMzEiIGQ9I
k0xMjUgMzBMMzEuOSA

2My4ybDE0LjIgMTIzLjFMMTI1IDIzMGw3OC45LTQzLjcgMTQuMi0xMjMuMXoiIC8+
CiAgICA8cGF0aCBma

WxsPSIjQzMwMDJGIiBkPSJNMTI1IDMwdjIyLjItLjFWMjMwbDc4LjktNDMuNyAxNC
4yLTEyMy4xTDEyNSA

zMHoiIC8+CiAgICA8cGF0aCZmlsbD0iI0ZGRkZGRiIgZD0iTTEyNS1Mi4xTDY2
LjggMTgyLjZoMTgyLjN2

wxMS43LTI5LjJoNDkuNGwxMS43IDI5LjJIMTgzLTEyNSA1Mi4xem0xNyA4My4zaC0
zNGwxNy00MC45IDE3I

DQwLjl6IiAvPgogIDwvc3ZnPg==">7>
</div>

<h2>Here are some links to help you start:</h2>
<ul>
   <li>
      <h2><a target = "_blank" rel = "noopener"
         href = "https://angular.io/tutorial">Tour of Heroes</a>
      </h2>
   </li>
   <li>
      <h2><a target = "_blank" rel = "noopener"
         href = https://angular.io/cli">CLI Documentation</>
      </h2>
   </li>
```

```
    <li>
       <h2><a target = "_blank" rel = "noopener"
          href = "https://blog.angular.io/">Angular blog</a>
       </h2>
    </li>
</ul>
<router-outlet></router-outlet>
```

app.component.spec.ts

Ide írjuk a tesztet.

app.component.ts

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7';
}
```

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

const routes: Routes = [];
@NgModule({
   imports: [RouterModule.forRoot(routes)],
   exports: [RouterModule]
})
export class AppRoutingModule { }
```

Assets

Itt tárolhatjuk a képeket, js fájlokat.

Environment

Ez a mappa a production vagy a dev environment fájljait tartalmazza. A mappa két fájlt tartalmaz:

• environment.prod.ts

• environment.ts

Mindkét fájl részletezi, hogy a végleges fájlt a production vagy a dev környezetben kell-e fordítani:

Az angular7-app / mappa kiegészítő fájlszerkezete a következőket tartalmazza:

favicon.ico

Ez egy fájl, amelyet általában egy webhely gyökérkönyvtárában talál.

index.html

A böngésző ennek a fájlnak a tartalmát jeleníti meg.

```
<html lang = "en">
   <head>
      <meta charset = "utf-8"7gt;
      <title>Angular7App</title>
      <base href = "/">
      <meta name = "viewport" content = "width=device-width,
initial-scale=1">
      <link rel = "icon" type = "image/x-icon" href =
"favicon.ico">
   </head>
   <body>
      <app-root></app-root>
   </body>
</html>
```

A body-nak <app-root> </app-root> -ja van. Ez az a selector, amelyet az app.component.ts fájl használ, és megjeleníti az app.component.html fájl-t.

main.ts

A main.ts az a fájl, ahonnan megkezdjük a projekt fejlesztését. A szükséges modul importálásával kezdődik. Most egy /core, angular/platform-browser-dynamic, app.module és environment vannak importálva.

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-
browser-dynamic';
import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
   enableProdMode();
}
platformBrowserDynamic().bootstrapModule(AppModule).catch(err =>
console.error(err));
```

A platformBrowserDynamic().bootstrapModule(AppModule) az AppModule-t adjuk meg szülő modulnak. Ezért amikor a böngészőben lefut, index.html oldalon érhető el. Az index.html belsőleg a main.ts-re utal, amely meghívja a szülő modult, azaz az AppModule-t, amikor a következő kód lefut –

```
platformBrowserDynamic().bootstrapModule(AppModule).catch(err =>
console.error(err));
```

Amikor az AppModule meghívásra kerül, akkor az app.module.ts-t hívja meg.

```
bootstrap: [AppComponent]
```

Az app.component.ts fájlban van egy selector: app-root, amelyet az index.html fájl használ. Megjeleníti az app.component.html webhelyen található tartalmat.

polyfill.ts

Ezt a régebbi verziók kompatibilitására használják.

styles.css

A stílusfájl.

test.ts

Unit tesztek.

tsconfig.app.json

Ezt a fordítás során használják, rendelkezik a konfiguráció részleteivel, amelyeket az alkalmazás futtatásához kell használni.

tsconfig.spec.json

Tesztekhez

typings.d.ts

A Typescript definícióra szolgál.

Angular7 - Components

A fejlesztés nagy részét az Angular 7-rel a komponensek hajtják végre. A komponensek alapvetően olyan osztályok, amelyek kölcsönhatásba lépnek a komponensek .html fájljával. (A .html fog megjelenni a böngészőben).

Az alkalmazás komponense az alábbi fájlokból áll:

- app.component.css
- app.component.html
- app.component.spec.ts
- app.component.ts
- app.module.ts

És ha a routing-ot is kiválasztottuk a project generálása során, akkor az alábbi fájl is a komponnshez adódik:

- app-routing.module.ts

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
   declarations: [
      AppComponent
   ],
```

```
   imports: [
      BrowserModule,
      AppRoutingModule
   ],
   providers: [],
   bootstrap: [AppComponent]
})
export class AppModule { }
```

Új komponens létrehozása:

```
ng g component new-cmp
```

Angular7 - Data Binding

Az adatkötéshez az alábbi operátort használjuk: {{}}.

Example

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

@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7';

   // declared array of months.
   months = ["January", "February", "March", "April", "May",
"June", "July",
      "August", "September", "October", "November", "December"];

   isavailable = true; //variable is set to true
}
```

app.component.html

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1> Welcome to {{title}}. </h1>
</div>

<div> Months :
   <select>
      <option *ngFor = "let i of months">{{i}}</option>
   </select>
</div>
```

```
<br/>

<div>
   <span *ngIf = "isavailable">Condition is valid.</span>
   //over here based on if condition the text condition is valid
is displayed.
   //If the value of isavailable is set to false it will not
display the text.
</div>
```

Example

if else

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7';

   // declared array of months.
   months = ["January", "Feburary", "March", "April",
"May","June", "July",
      "August", "September", "October", "November", "December"];

   isavailable = false; //variable is set to true
}
```

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1> Welcome to {{title}}. </h1>
</div>

<div> Months :
   <select>
      <option *ngFor = "let i of months">{{i}}</option>
   </select>
</div>
<br/>

<div>
   <span *ngIf = "isavailable; else condition1">Condition is
valid.</span>
   <ng-template #condition1>Condition is invalid</ng-template>
</div>
```

if then else

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7';

   // declared array of months.
   months = ["January", "February", "March", "April", "May",
"June", "July",
      "August", "September", "October", "November", "December"];

   isavailable = true; //variable is set to true
}
```

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1> Welcome to {{title}}. </h1>
</div>

<div> Months :
   <select>
      <option *ngFor="let i of months">{{i}}</option>
   </select>
</div>
<br/>

<div>
   <span *ngIf = "isavailable; then condition1 else condition2">
      Condition is valid.
   </span>
   <ng-template #condition1>Condition is valid</ng-template>
   <ng-template #condition2>Condition is invalid</ng-template>
</div>
```

Angular7 - Event Binding

app.component.html

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1>Welcome to {{title}}.</h1>
</div>

<div> Months :
   <select>
      <option *ngFor = "let i of months">{{i}}</option>
```

```
   </select>
</div>
<br/>

<div>
   <span *ngIf = "isavailable; then condition1 else condition2">
      Condition is valid.
   </span>
   <ng-template #condition1>Condition is valid</ng-template>
   <ng-template #condition2>Condition is invalid</ng-template>
</div>
<button (click) = "myClickFunction($event)">
   Click Me
</button>
```

app.component.ts

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7';

   // declared array of months.
   months = ["January", "February", "March", "April",
"May","June", "July",
      "August", "September", "October", "November", "December"];

   isavailable = true; //variable is set to true
   myClickFunction(event) {
      //just added console.log which will display the event
details in browser on click of the button.
      alert("Button is clicked");
      console.log(event);
   }
}
```

add.component.css −

```
button {
   background-color: #2B3BCF;
   border: none;
   color: white;
   padding: 10px 10px;
   text-align: center;
   text-decoration: none;
   display: inline-block;
   font-size: 20px;
```

```
}
```

onchange event to the dropdown

app.component.html

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1>Welcome to {{title}}.</h1>
</div>

<div> Months :
   <select (change) = "changemonths($event)">
      <option *ngFor = "let i of months">{{i}}</option>
   </select>
</div>
<br/>

<div>
   <span *ngIf = "isavailable; then condition1 else condition2">
      Condition is valid.
   </span>
   <ng-template #condition1>Condition is valid</ng-template>
   <ng-template #condition2>Condition is invalid</ng-template>
</div>
<br/>

<button (click) = "myClickFunction($event)">
   Click Me
</button>
```

app.component.ts file –

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7';

   // declared array of months.
   months = ["January", "Feburary", "March", "April", "May",
"June", "July",
      "August", "September", "October", "November", "December"];

   isavailable = true; //variable is set to true
   myClickFunction(event) {
      //just added console.log which will display the event
      details in browser on click of the button.
```

```
      alert("Button is clicked");
      console.log(event);
   }
   changemonths(event) {
      console.log("Changed month from the Dropdown");
      console.log(event);
   }
}
```

Angular7 - Templates

app.component.html

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1>Welcome to {{title}}.</h1>
</div>

<div> Months :
   <select (change) = "changemonths($event)" name = "month">
      <option *ngFor = "let i of months">{{i}}</option>
   </select>
</div>
<br/>

<div>
   <span *ngIf = "isavailable;then condition1 else condition2">
      Condition is valid.
   </span>
   <ng-template #condition1>Condition is valid from template</ng-
template>
   <ng-template #condition2>Condition is invalid from
template</ng-template>
</div>
<button (click) = "myClickFunction($event)">Click Me</button>
```

app.component.ts

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7';

   // declared array of months.
   months = ["January", "February", "March", "April", "May",
"June", "July",
```

```
      "August", "September", "October", "November", "December"];
   isavailable = false; // variable is set to true

   myClickFunction(event) {
      //just added console.log which will display the event
details in browser on click of the button.
      alert("Button is clicked");
      console.log(event);
   }
   changemonths(event) {
      alert("Changed month from the Dropdown");
   }
}
```

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1> Welcome to {{title}}. </h1>
</div>

<div> Months :
   <select (change) = "changemonths($event)" name = "month">
      <option *ngFor = "let i of months">{{i}}</option>
   </select>
</div>
<br/>

<div>
   <span *ngIf = "isavailable; else condition2">
      Condition is valid.
   </span>
   <ng-template #condition1>Condition is valid from template
</ng-template>
   <ng-template #condition2>Condition is invalid from
template</ng-template>
</div>
<button (click) = "myClickFunction($event)">Click Me</button>
```

Angular7 - Directives

A direktívák javascript osztályok, amelyeket a @directive segítségével deklarálunk. Az Angular-ban 3 direktíva van:

Component Directives

Meghatározza a komponensek feldolgozását, példányosítását és futásidejű használatát.

Structural Directives

A strukturális direktívák alapvetően a dom elemek manipulálásával foglalkoznak. A strukturális direktíváknak * jelük van a direktíva előtt. Például: * ngIf és * ngFor.

Attribute Directives

Az attribútum direktívák a dom elem megjelenésének és viselkedésének megváltoztatásával foglalkoznak. Saját direktívát is készíthetünk, ezt a következőkben nézzük meg:

How to Create Custom Directives?

Ezeket a direktívákat mi magunk készítjük.

app.module.ts

```
import { BrowserModule } from'@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { NewCmpComponent } from'./new-cmp/new-cmp.component';
import { ChangeTextDirective } from './change-text.directive';

@NgModule({
   declarations: [
      AppComponent,
      NewCmpComponent,
      ChangeTextDirective
   ],
   imports: [
      BrowserModule,
      AppRoutingModule
   ],
   providers: [],
   bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.html

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "text-align:center">
   <h1> Welcome to {{title}}. </h1>
</div>
<div style = "text-align:center">
   <span changeText >Welcome to {{title}}.</span>
</div>
```

change-text.directive.ts

```
import { Directive, ElementRef} from '@angular/core';
```

```
@Directive({
   selector: '[changeText]'
})
export class ChangeTextDirective {
   constructor(Element: ElementRef) {
      console.log(Element);
      Element.nativeElement.innerText = "Text is changed by
changeText Directive.";
   }
}
```

Angular7 - Pipes

Az adat átalakítására a | karaktert használjuk.

```
{{ Welcome to Angular 7 | lowercase}}
```

Egész számok, karakterláncok, tömbök és dátum során a | konvertál a kívánt formátumban.

app.component.ts

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7 Project!';
}
```

app.component.html file −

```
<b>{{title | uppercase}}</b><br/>
<b>{{title | lowercase}}</b>
```

Néhány beépített pipe:

- Lowercasepipe
- Uppercasepipe
- Datepipe
- Currencypipe
- Jsonpipe
- Percentpipe
- Decimalpipe
- Slicepipe

app.component.ts file −

```
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Angular 7 Project!';
   todaydate = new Date();
   jsonval = {name:'Rox', age:'25', address:{a1:'Mumbai',
a2:'Karnataka'}};
   months = ["Jan", "Feb", "Mar", "April", "May", "Jun", "July",
"Aug",
      "Sept", "Oct", "Nov", "Dec"];
}
```

app.component.html file as shown below −

```
<!--The content below is only a placeholder and can be replaced.-
->
<div style = "width:100%;">
   <div style = "width:40%;float:left;border:solid 1px black;">
      <h1>Uppercase Pipe</h1>
      <b>{{title | uppercase}}</b>
      <br/>

      <h1>Lowercase Pipe</h1>
      <b>{{title | lowercase}}</b>
      <h1>Currency Pipe</h1>
      <b>{{6589.23 | currency:"USD"}}</b>
      <br/>

      <b>{{6589.23 | currency:"USD":true}}</b>
      // Boolean true is used to get the sign of the currency.
      <h1>Date pipe</h1>
      <b>{{todaydate | date:'d/M/y'}}</b>
      <br/>

      <b>{{todaydate | date:'shortTime'}}</b>
      <h1>Decimal Pipe</h1>
      <b>{{ 454.78787814 | number: '3.4-4' }}</b>
      // 3 is for main integer, 4 -4 are for integers to be
displayed.
   </div>

   <div style = "width:40%;float:left;border:solid 1px black;"<
      <h1<Json Pipe</h1>
      <b>{{ jsonval | json }}</b>
      <h1>Percent Pipe</h1>
      <b>{{00.54565 | percent}}</b>
      <h1>Slice Pipe</h1>
      <b>{{months | slice:2:6}}</b>
```

```
      // here 2 and 6 refers to the start and the end index
   </div>
</div>
```

How to Create a Custom Pipe?

app.sqrt.ts

```
import {Pipe, PipeTransform} from '@angular/core';
@Pipe ({
   name : 'sqrt'
})
export class SqrtPipe implements PipeTransform {
   transform(val : number) : number {
      return Math.sqrt(val);
   }
}
```

app.module.ts. This is done as follows −

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { NewCmpComponent } from './new-cmp/new-cmp.component';
import { ChangeTextDirective } from './change-text.directive';
import { SqrtPipe } from './app.sqrt';

@NgModule({
   declarations: [
      SqrtPipe,
      AppComponent,
      NewCmpComponent,
      ChangeTextDirective
   ],
   imports: [
      BrowserModule,
      AppRoutingModule
   ],
   providers: [],
   bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.html file.

```
<h1>Custom Pipe</h1>
<b>Square root of 25 is: {{25 | sqrt}}</b>
<br/>
```

```
<b>Square root of 729 is: {{729 | sqrt}}</b>
```