

Pro Angular

Chapter 02: Your First Angular App

14 July 2019

Q1. Which tool is used to access, download node packages and manage the dependencies between them in your angular application?

- A. libman
- B. angular-cli
- C. npm
- D. typescript

Answer: C

Q2. Which commands are valid command to check installed version of npm?

- A. npm -v
- B. npm --version
- C. npm -ls
- D. npm --list

Answer: A, B

Q3. Which command is valid to update npm?

- A. npm update -g npm@latest
- B. npm -u -g npm@latest
- C. npm init -g npm@latest
- D. npm install -g npm@latest

Answer: D

Q4. Which file is required by typescript compiler to transpile typescript file into JavaScript?

- A. package.json
- B. typings.json
- C. tsconfig.json
- D. module.js

Answer: C

Q5. Which keyword is used to identity data or types that you want to use elsewhere in the application?

- A. public
- B. export
- C. decorator
- D. @component

Answer: B

Q6. Which one is an data-binding example in an angular application?

- A. @title
- B. <%# title %>
- C. {{title}}
- D. Bind ("title")

Answer: C

Q7. Which one is responsible for managing a template and providing it with the data and logic it needs in an angular application?

- A. module
- B. component

-
- C. service
 - D. model

Answer: B

Q8. Which keyword is used to declare a dependency on the contents of a JavaScript module?

- A. using
- B. use
- C. import
- D. imports

Answer: C

Q9. Which one provides metadata about a class?

- A. import keyword
- B. export keyword
- C. decorator
- D. injector

Answer: C

Q10. Which decorator is used to identity a class as a component?

- A. @Injectable ({...})
- B. @NgModule ({...})
- C. @Component ({...})
- D. @Import ({...})

Answer: C

Q11. Which angular expression is used to repeat a region of content for each item in an array?

- A. *ngRepeat
- B. *ngFor
- C. *ngSwitch
- D. *ngIf

Answer: B

Q12. Which angular expression is a conditional statement that is used to insert different sets of elements into the document based on a specified value?

- A. *ngRepeat
- B. *ngFor
- C. *ngSwitch
- D. *ngIf

Answer: C

Q12. Which one is an example of input binding?

- A. <button (click)="logText(textbox.value)">Update Log</button>
- B. <input [value]="username">
- C. <input [(ngModel)]="username">
- D. All of the above

Answer: B

[A: output binding

C: two-way binding]

Q13. Which one is an example of output binding?

- A. <button (click)="logText(textbox.value)">Update Log</button>
- B. <input [value]="username">
- C. <input [(ngModel)]="username">
- D. <p>Hello {{username}}!</p>

Answer: A

[B: output binding

B: input binding

D: one-way/input binding]

Q14. Which one is an example of two-way binding?

A. `<button (click)="logText(textbox.value)">Update Log</button>`

B. `<input [value]="username">`

C. `<input [(ngModel)]="username">`

D. `<p>Hello {{username}}!</p>`

Answer: C

Q15. Which character is used to define a variable to refer to the element in the template's data bindings?

A. @

B. #

C. ?

D. !

Answer: B

`<input type='text' #todoText />`

`<button (click)="addItem(todoText.value)"> Add </button>`

Chapter 03: Putting Angular in Context

14 July 2019

Q1. Angular supports _____.

A. Model-View-Controller (MVC) pattern

B. Model-View-ViewModel (MVVM) pattern

C. Hierarchical Model-View-Controller (HMVC) pattern

D. Model View Adapter (MVA) pattern

Answer: A

[Page 31]

Q2. What can the HTML document produced by an Angular application contain?

A. Only standard markup

B. Only custom markup

C. A mix of standard and custom markup

D. Either standard or custom markup

Answer: C

[Page 31]

Q3. What type of web apps you use the angular for?

A. web apps developed following the round-trip model

B. web apps developed following the single-page model

C. web apps developed following the round-trip or single-page model

D. web apps developed following both the round-trip and single-page model

Answer: B

[Page 32]

Q4. Angular requires you to use _____.

A. JavaScript

-
- B. TypeScript
 - C. VBScript
 - D. Babel

Answer: B

Q5. Which one the model in an application built using the MVC pattern should **not** contain?

- A. the domain data
- B. the logic for creating, managing, and modifying the domain data
- C. a clean API that exposes the model data and operations on it
- D. logic for displaying data to the user

Answer: D

Q6. Which one acts as controllers in an angular web application?

- A. Module
- B. Component
- C. Service
- D. Decorator

Answer: B

[Page 37]

Q7. Views in Angular are known as _____.

- A. Components
- B. Modules
- C. Templates
- D. Injectables

Answer: C

[Page 38]

Q8. RESTful services use _____ to map operations?

- A. SOAP messages
- B. RPC
- C. a combination of the HTTP method and the URL
- D. only the URL

Answer: C

[Page 39]

Q9. Which HTTP method is generally used to create a new data object in RESTful services?

- A. GET
- B. POST
- C. PUT
- D. DELETE

Answer: B

[Page 40]

Q10. Which HTTP method is generally used to update the data object specified by the URL in RESTful services?

- A. GET
- B. POST
- C. PUT
- D. DELETE

Answer: C

[Page 40]

Chapter 05: JavaScript and TypeScript: Part 1

14 July 2019

Q1. Which code will **not** work?

- A.

```
function f() { /*...*/ }  
f();
```
- B.

```
f();  
function f() { /*...*/ }
```
- C.

```
let f = function () { /*...*/ }  
f();
```
- D.

```
f();  
let f = function () { /*...*/ }
```

Answer: D

[Page 67]

Ref. function hoisting at page 68]

Q2. How do you define rest parameters in function?

- A. First parameter of function prefixed with three periods, ...
- B. Last parameter of function prefixed with three periods, ...
- C. Any parameter of function prefixed with three periods, ...
- D. All of the above

Answer: B

[Page 70]

Q3. Which function of the following uses rest parameters?

- A.

```
function applicant(name, post, qualifications='') { /*..*/ }
```
- B.

```
function applicant(name, post, qualifications=[]) { /*..*/ }
```
- C.

```
function applicant(name, post, qualifications=null) { /*..*/ }
```
- D.

```
function applicant(name, post, ...qualifications) { /*...*/ }
```

Answer: D

[Page 70]

Q4. What is an arrow function?

- A. a function contains a single statement
- B. a function that does not accept any parameter
- C. a function contains that does not return a value (void function)
- D. an alternative way of defining functions and are often used as arguments to other functions

Answer: D

[Page 71]

Q5. Consider the code below

```
let p = (name, gender) => {  
  var m = 'Have a good day';  
  if(gender == 'M')  
  {  
    var m = 'Mr. ' + name;  
    console.log(m);  
  }  
  else  
  {  
    var m = 'Mrs. ' + name;
```

```
        console.log(m);
    }
    console.log(m);
}
p('Alam', 'M');
```

What will be the output?

- A. Mr. Alam
Have a good day
- B. Have a good day
Have a good day
- C. Mr. Alam
Mr. Alam
- D. Code will not work

Answer: C

[var is scoped to function so even if it is declared in the inner block]

Q6. Consider the code below

```
let p = (name, gender) => {
    Let m = 'Have a good day';
    if(gender == 'M')
    {
        let m = 'Mr. ' + name;
        console.log(m);
    }
    else
    {
        Let m = 'Mrs. ' + name;
        console.log(m);
    }
    console.log(m);
}
p('Alam', 'M');
```

What will be the output?

- A. Mr. Alam
Have a good day
- B. Have a good day
Have a good day
- C. Mr. Alam
Mr. Alam
- D. Code will not work

Answer: A

[let is scoped to the block it is declared]

Q7. Which one uses templated string correctly?

- A. var course = "ESAD"; var label = "Your Course " + course;
- B. var course = "ESAD"; var label = \$"Your Course {course}";
- C. var course = "ESAD"; var label = "Your Course \${course}";
- D. var course = "ESAD"; var label = `Your Course \${course}`;

Answer: D

Q8. You have an array variable

```
let colors= ['Red', 'Green', 'Blue']
```

Which one prints array items correctly using forEach function?

- A. colors.forEach(x=> console.log(x));
- B. colors.forEach(x=> console.log(colors[x]));
- C. colors.forEach((x,i)=> console.log(x[i]));
- D. colors.forEach(x=> console.log(i));

Answer: A

[Syntax

array.forEach(function(value){...}

array.forEach(function(value, index){...})

Q9. Which one useful to expand an array so that its contents can be combined with other arrays?

- A. forEach function
- B. join function
- C. Spread operator
- D. Arrow function

Answer: C

Q10. You have an array variable

let colors= ['Red', 'Green', 'Blue']

You want to create another array with all the items in colors array and some extra items.

Which code does this correctly?

- A. let colors4 = colors.join('Yellow');
- B. let colors4 = colors.concat('Yellow');
- C. let colors4 = [colors, 'Yellow'];
- D. let colors4 = [...colors, 'Yellow'];

Answer: D

Chapter 06: JavaScript and TypeScript: Part 2

14 July 2019

Q1. How do you define a constructor in a class in TypeScript?

By defining a function with the name same as the class

- A. By defining a method named new
- B. By defining a method named constructor
- C. By defining a method named this

Answer: C

[Page: 90]

Q2. Which one correctly defines a readable/writable property fullName in a class?

- A. fullName { get { /*..*/ } set { /*..*/ }}
- B. getFullName() { /*...*/ }
setFullName(value){ /*..*/ }
- C. get fullName { { /*...*/ }
set fullName { /*...*/ } }
- D. get fullName() { { /*...*/ }
set fullName(value) { /*...*/ } }

Answer: D

[Page: 92

Syntax

Getter: get propertyName() {return data; }

Setter: set propertyName(value) {data = value; }

Q3. Which keyword is used in the class definition to indicate that the class is derived from another superclass?

- A. inherits
- B. extends
- C. derived
- D. A colon (:) character

Answer: B

[Page: 93]

Q4. _____are used to manage the dependencies in a web application?

- A. JavaScript modules
- B. JavaScript classes
- C. JavaScript functions
- D. JavaScript files

Answer: A

[Page: 93]

Q5. You have defined a class Trainee in src/trainee.model.ts file

How do you make sure that class can be used outside of the file in which it is defined?

- A. Add the @Injectable decorator in the class
- B. Add the @Component decorator in the class
- C. Add the @Export decorator in the class
- D. Add the export keyword before the class definition

Answer: D

[Page: 94]

Q6. You have defined a class Trainee in src/trainee.model.ts file

What should you do to use the class in another TypeScript file?

- A. Add the class in the imports array in app.module.ts file for bootstrapping
- B. Add the import declaration in the file where you want to use the class
- C. Create a class and inject the trainee class in the constructor
- D. None of the above

Answer: B

[Page: 94]

import { Trainee } from "../trainee.module"]

Q7. Which one is the correct typescript variable with type annotation?

- A. number count;
- B. number:count
- C. count:number
- D. count as number

Answer: C

[Page: 98]

Q8. Which typescript function declaration of the following does not properly annotate types that it accepts a string and returns a number?

- A. function wordCount(s){ return s.length;}
- B. function wordCount(s:string):number{ return s.length;}
- C. let wordCount:(s:string)=>number = function(s) { return s.length;}
- D. let wordCount:(s:string)=>number = (s) => s.length;}

Answer: A

[Page: 98]

Q9. In typescript, to declare multiple types for a variable, you use_____

- A. a comma (,) character to separate types
- B. a colon (:) character to separate types
- C. a bar (|) character to separate types
- D. an ampersand (&) character to separate types

Answer: C

[Page: 101]

let m: number|boolean]

Q10. What is a union type in typescript?

- A. A variable for which type is unknown until runtime
- B. A variable for which type is determined from the assignment expression
- C. A variable which can contain multiple types of value
- D. A variable which can contain a set of distinct values

Answer: C

[Page: 101]

Q11. Which variable declaration is not valid?

- A. X:any
- B. X:number, string
- C. X:number|bool
- D. X:string[]

Answer: B

[Page: 101]

Q12. Which one is the type assertion syntax in typescript?

- A. (type)variable
- B. <type>variable
- C. variable<type>
- D. variable instanceof type

Answer: B

[Page: 101]

<type>variable or variable as type]

Q13. Which code block is **not** valid?

- A. let count:number = Number ("21")+1;
- B. let count:number = <number>("21")+1;
- C. let count:number =("21" as number)+1;
- D. let count:number = "21"+1;

Answer: D

[Page: 101]

Q14. What is a tuple?

- A. fixed-length arrays, where each item in the array is of a specified type
- B. fixed-length arrays, where all the items in the array are of specified type
- C. key/value pair collection
- D. sorted collection

Answer: A

[Page: 102]

Chapter 07: SportsStore: A Real Application

20 July 2019

Q1. Which one describes the application to the angular including which features are required to run the application?

- A. Root component
- B. Root module
- C. Feature module
- D. Injectable services

Answer: B

[Page: 114]

Two types of module: root module and feature module]

Q2. Which one is used to group related application functionality to make the application easier to manage?

- A. Root component
- B. Root module
- C. Feature module
- D. Injectable services

Answer: C

[Page: 114]

Q3. Which Angular building block manages the contents of the app element in the HTML document?

- A. Root component
- B. Root module
- C. Feature module
- D. Injectable services

Answer: A

[Page: 113]

Q4. How do you tell the root module which modules to load?

- A. In the imports property of the @NgModule decorator of the root module
- B. In the declarations property of the @NgModule decorator of the root module
- C. In the providers property of the @NgModule decorator of the root module
- D. In the bootstrap property of the @NgModule decorator of the root module

Answer: A

[Page: 114]

Q5. How do you tell the root module which components to load?

- A. In the imports property of the @NgModule decorator of the root module
- B. In the declarations property of the @NgModule decorator of the root module
- C. In the providers property of the @NgModule decorator of the root module
- D. In the bootstrap property of the @NgModule decorator of the root module

Answer: B

[Page: 114]

Q6. How do you tell the root module which classes should be used as services for the dependency injection feature?

- A. In the imports property of the @NgModule decorator of the root module
- B. In the declarations property of the @NgModule decorator of the root module
- C. In the providers property of the @NgModule decorator of the root module
- D. In the bootstrap property of the @NgModule decorator of the root module

Answer: C

[Page: 137]

Q7. How do tell Angular that a certain class will be used as a service?

- A. By applying export keyword before the class declaration
- B. By applying @Injectable decorator to the class
- C. By applying @Component decorator to the class
- D. By applying @NgModule decorator to the class

Answer: B

[Page: 137]

Q8. How do you tell the angular how to apply the component to HTML content?

- A. using the selector property of @Component decorator
- B. using the templateUrl property of @Component decorator
- C. using the template property of @Component decorator
- D. using the styleUrls property of @Component decorator

Answer: A

[Page: 129]

Q9. How do you tell the angular how to how to find the component's template?

- A. using the selector property of @Component decorator
- B. using the templateUrl property of @Component decorator
- C. using the template property of @Component decorator
- D. using the styleUrls property of @Component decorator

Answer: B

[Page: 129]

Q10. What are pipes in angular?

- A. Pipes are classes used to transform or prepare a data value for its use in a data binding
- B. Pipes are used to chain statements to run in sequence
- C. Pipes are used to attach methods to observable collection
- D. Pipes are used to subscribe to events

Answer: A

[Page: 123]

{{ strDate | date : 'dd-MM-yyyy' }}

Chapter 08: SportsStore: Orders and Checkout

21 July 2019

Q1. Which one of the following tools you use to initialize, develop, scaffold, and maintain Angular applications?

- A. node
- B. npm
- C. ng (angular-cli)
- D. bower

Answer: C

[Page: 135]

Q2. Which angular-cli command do you start the embedded HTTP server at its default port?

- A. ng open
- B. ng start
- C. ng serve
- D. ng build

Answer: C

[Page: 135]

Q3. What is the default port of the embedded HTTP server in angular application?

- A. 3000
- B. 4200
- C. 8080
- D. 80

Answer: B

Q4. Which angular-cli command do you start the embedded HTTP server at a specified port xxxx?

- A. ng serve xxxx
- B. ng serve --xxxx
- C. ng serve --port xxxx
- D. ng serve:xxxx

Answer: C

[Page: 135]

Q5. Which angular-cli command do you start the embedded HTTP server at a specified port 3000 and open the application with the default browser?

- A. ng serve --port 3000 open
- B. ng serve --port 3000 -open
- C. ng serve --port 3000 --open
- D. ng serve --port 3000 --start

Answer: C

[Page: 135]

Q6. Which class can be used as a service with dependency injection?

- A. export class DataService{}
- B. @service()
export class DataService{}
- C. @Injectable()
export class DataService{}
- D. @InjectableType<DataService>()
export class DataService{}

Answer: C

[Page: 137]

Q7. You have an injectable service DataService and you have registered service in root module.

To use the DataService with as dependency injection in component, which code block should you write in the component class?

- A. @Component(...)
export class UsefulComponent {
 constructor(private svc:DataService){}
- B. @Component(...)
export class UsefulComponent {
 svc:DataService;
 ngOnInit(svc:DataService) {
 this.svc = svc;
 }
}
- C. @Component(...)
export class UsefulComponent {
 svc:DataService;
 set serive(svc){
 this.svc=svc;

```
    }  
  }  
D. @Component(...)  
  export class UsefulComponent {  
    svc:DataService;  
    get service() {  
      return this.svc;  
    }  
    set serive(svc){  
      this.svc=svc;  
    }  
  }  
}
```

Answer: A

Q8. Which angular feature allows to select the components based on the URL displayed by the browser?

- A. Dependency injection
- B. Reusable components
- C. URL routing
- D. Route guards

Answer: C

[Page: 142]

Q9. Which angular class enables navigation from one view to the next as users perform application tasks?

- A. module
- B. component
- C. router
- D. Http utility class

Answer: C

Q10. Which one do you in an anchor element for a navigation by the users?

- A. href attribute
- B. ink attribute
- C. routerLink attribute
- D. router attribute

Answer: C

[Page: 183]

Q11. Which features do you use to control the routing system such preventing user from navigating to a url without authentication?

- A. Dependency injection
- B. Reusable components
- C. URL routing
- D. Route guards

Answer: D

[Page: 148]

Q12. How do you implement a route guard?

- A. Use the the RouterModule
- B. Use the Router object's navigateByUrl
- C. Create a class that defines a canActivate method
- D. None of the above

Answer: C

[Page: 149]

Chapter 09: SportsStore: Administration

21 July 2019

Q1. JWT stands for _____

- A. Java Web Token
- B. JavaScript Web Token
- C. JSON Web Token
- D. JSON Web Token

Answer: C

Q2. Why do you use JWT?

- A. In SPA, for authentication via third-party like facebook or google
- B. In SPA, for role-based authentication only
- C. In SPA, for authentication in RESTful services
- D. In SPA, for Azure Active Directory based authentication

Answer: C

Q3. What is the format of header that must be sent to RESTful services that implements JWT bearer token for security?

- A. Authorization: token-string
- B. Authorization: Bearer token-string
- C. Authorization: Bearer: token-string
- D. Bearer: token-string

Answer: B

Q4. What is required to prevent users from navigating to certain components without login?

- A. Route configuration
- B. Route guard
- C. JWT
- D. ActiveRoute

Answer: B

Q5. Which one should you use at minimum for data from a RESTful service?

- A. `import { HttpClient } from "@angular/common/http";`
- B. `import { HttpHeaders } from '@angular/common/http';`
- C. `import { Router } from "@angular/router";`
- D. `import { FormsModule } from "@angular/forms";`

Answer: A

[With earlier version (<6)

`import { Http } from "@angular/http"`

Now new improved HttpClient is introduced]

Q6. You have a RESTful at url `http://remoteserver:9110/products`

You have the service like below

`@Injectable()`

```
export class MyService {  
  constructor(private http: HttpClient) {  
  }  
}
```

Which one is the correct way to call for fetching data from the REST service?

-
- A. `getProducts():any{ return this.http("http://remoteserver:9110/products");}`
 - B. `getProducts():any{ return this.http.get("http://remoteserver:9110/products");}`
 - C. `getProducts():any{ return this.http("GET", "http://remoteserver:9110/products");}`
 - D. All of the above

Answer: B

Q7. What is the correct way of calling the `get` method of `HttpClient` of the `HttpClient` class with headers?

- A. `get(url)`
- B. `get(url, httpOptions)`
- C. `get(httpOptions, url)`
- D. `get(url, data, httpOptions)`

Answer: B

Q8. What is the correct way of calling the `get` method of `HttpClient` of the `HttpClient` class with data?

- A. `post(url, data)`
- B. `post(data, url)`
- C. `post(url, {data})`
- D. `post(url, data, format)`

Answer: A

Chapter 10: SportsStore: Progressive Features and Deployment

21 July 2019

Q1. What is a progressive web application (PWA)?

- A. A web application is initially support limited functions and gradually features are added to it
- B. A web application is initially support limited function on older browsers but on modern browsers it supports full functionalities
- C. A web application that is developed using iterative life cycle model
- D. A web application that behaves more like a native application, which means it can continue working when there is no network connectivity, its code and content are cached so it can start immediately, and it can use features such as notifications

Answer: D

[Page: 193]

Q2. Which package do you need to install to add Progressive Features to your angular applications?

- A. `angular-cli`
- B. `angular-cache`
- C. `angular/pwa`
- D. `RxJS/MemoryCache`

Answer: C

[Page: 194]

Q3. What does the `@angular/pwa` package do to your angular applications?

- A. It caches Observable collection that you fetch remote from RESTful services in your angular application
- B. It enables animation during view transitions
- C. It as reactive features to angular applications

-
- D. It configures the application so that HTML, JavaScript, and CSS files are cached, which will allow the application to be started even when there is no network available

Answer: D

[Page: 194]

Q4. Which configuration file specifies which files and data URLs the Angular service worker should cache and how it should update the cached files and data?

- A. src/ngsw-config.json
- B. src/tsconfig.app.json
- C. src/tsconfig.spec.json
- D. src/tslint.json

Answer: A

[Page: 194]

Q5. Which section of the ngsw-config file specifies which application data to be cached?

- A. assetGroups
- B. dataGroups
- C. cacheConfig
- D. resources

Answer: B

[Page: 195]

Q6. Look at the ngsw-config.json section

```
"dataGroups": [  
  {  
    "name": "...",  
    "urls": [...],  
    "cacheConfig": {  
      "maxSize": 100,  
      "maxAge": "5d"  
    }  
  }  
]
```

What does it specify?

- A. the cache will contain data maximum 100MB data and that data will be valid for 5 days
- B. the cache will contain data of 100 objects and that data will be valid for 5 days
- C. the cache data will be valid for a minimum of 5 days and maximum 100 days
- D. the cache will contain data from 100 requests and that data will be valid for 5 days

Answer: D

[Page: 195]

Q7. Which of the following do you use to determine the connection status of the application?

- A. the browser's navigator.onLine
- B. the browser's navigator.offLine
- C. the browser's navigator.connection
- D. the browser's navigator.isOnline

Answer: A

[Page: 196]

Q8. Which command do you use to build your angular application to be deployed to a remote server?

- A. ng build
- B. ng build --prod
- C. ng build --watch
- D. ng build --aot

Answer: B

[Page: 200]

[ng build for development environment

ng build --prod for production environment

ng build --watch if you want regenerates output files when source files change

ng build --aot for Ahead of Time compilation]

Q9. Which configuration file is required if you want to download the additional packages required by the application during deployment?

A. ts-config.json

B. config.json

C. package.json

D. Deploy-package.json

Answer: D

[Page: 203]

Chapter 11: Creating an Angular Project

22 July 2019

Q1. Which command should you use to create an angular application named myapp?

A. ng new myapp

B. ng create myapp

C. ng init myapp

D. npm init myapp

Answer: A

[Page: 211]

Q2. Which command should you use to create an angular application named myapp with routing module?

A. ng new myapp routing

B. ng new myapp --routing

C. ng new myapp -il routing

D. npm new myapp --add routing

Answer: B

[Page: 212]

Q3. Which file contains the configuration for the Angular development tools?

A. angular.json

B. package.json

C. tsconfig.json

D. tslint.json

Answer: A

[Page: 213]

Q1. used to specify the packages that the application relies on, including Angular itself. one is not a section required in package.json file in an angular application?

A. dependencies

B. devDepedencies

C. scripts

D. resources

Answer: D
[Page: 217]

Q2. Which section of the package.json files is used to specify the packages that the application relies on, including Angular itself?

- A. dependencies
- B. devDepedencies
- C. scripts
- D. resources

Answer: A
[Page: 218]

Q3. Which section of the package.json files is used to specify the packages that are used in development, such compilers and test frameworks

- A. dependencies
- B. devDepedencies
- C. scripts
- D. resources

Answer: B
[Page: 218]

Q4. Which file contains the details of dependency packages required by the angular application and the development tools and defines the commands that run the development tool

- A. package.json
- B. tsconfig.json
- C. ngsw-configuration.json
- D. angular.json

Answer: A
[Page: 213]

Q5. Which file the TypeScript compiler require to prepare an Angular project for use?

- A. package.json
- B. tsconfig.json
- C. ngsw-configuration.json
- D. angular.json

Answer: B
[Page: 213]

Q6. Which folder contains the application's source code, resources, and configuration files of the angular application?

- A. e2e
- B. src
- C. node_modules
- D. the root folder of the app

Answer: B
[Page: 213]

Q7. Which folder in an angular application contains the source code and contents?

- A. src
- B. e2e
- C. src/app
- D. src/assets

Answer: C
[Page: 216]

Q8. Which folder in an angular application is used for the static resources required by the application, such as images?

- A. src
- B. e2e
- C. src/app
- D. src/assets

Answer: D

[Page: 216]

Q9. Which file in angular application is used to provide support for features that are not available natively in some browsers, especially Internet Explorer?

- A. main.ts
- B. polyfills.ts
- C. karma.conf.js
- D. tests.ts

Answer: B

[Page: 216]

Q10. Which file contains the TypeScript statements that start the application when they are executed?

- A. main.ts
- B. polyfills.ts
- C. karma.conf.js
- D. tests.ts

Answer: A

[Page: 216]

Q11. Which command will start the development server in an angular application?

- A. npm start
- B. node start
- C. npm serve
- D. ng serve

Answer: D

[Page: 220]

Q12. What is hot module replacement (HMR)?

- A. A webpack feature that bundles all java scripts to use in browser
- B. A module loader
- C. A built-in webpack feature to start development server at a default port
- D. A webpack feature that detect changes to the application's source or content files, the altered file is compiled, put into a bundle by webpack, and sent to the browser.

Answer: D

[Page: 212]

Q13. Which section of the package.json files is used to define commands that are used to compile, test, or run the angular application?

- A. dependencies
- B. devDepedencies
- C. scripts
- D. resources

Answer: C

[Page: 220]

Q14. Which dependency package implements a reflection API that is used to inspect decorators on classes in angular web apps?

- A. reflect-metadata
- B. rxjs
- C. zone.js
- D. core-js

Answer: A

Q15. Which dependency package is used to implement the Angular change-detection system that is used in data bindings?

- A. reflect-metadata
- B. rxjs
- C. zone.js
- D. core-js

Answer: B

Q16. Which dependency package provides an execution context for asynchronous tasks and is used to evaluate template expressions?

- A. reflect-metadata
- B. rxjs
- C. zone.js
- D. core-js

Answer: C

Q17. Which dependency package provides support for new JavaScript features for browsers that don't implement them?

- A. reflect-metadata
- B. rxjs
- C. zone.js
- D. core-js

Answer: D

Q18. Which dependency package is the module loader?

- A. systemjs
- B. @angular/core
- C. classlist.js
- D. core-js

Answer: A

Q19. Which package provides the type information for popular JavaScript packages so they can be used more conveniently with TypeScript?

- A. typescript
- B. typings
- C. rxjs
- D. classlist.js

Answer: B

Chapter 12: Using Data Bindings

23 July 2019

Q1. Which data binding is used to generate content for the user and are the basic building block for Angular templates?

- A. One-way data binding
- B. Two-way data binding
- C. Event binding
- D. Changes tracking

Answer: A

[Page: 239]

Q2. Which one is not a one-way data binding?

- A. `{{Title}}`
- B. `<input [value]="firstName" />`
- C. `<button (click)="">Change </button>`
- D. `<input [(ngModel)]=p.Name" #f />`

Answer: D

[Page: 240]

Q3. Which Data Binding expression indicates that data flows from the expression to the target?

- A. `[target]="expression"`
- B. `{{expression}}`
- C. `(target) ="expression"`
- D. `[(target)] ="expression"`

Answer: A

[Page: 240]

[A. Input binding

B. String interpolation binding

C. Output binding

D. Two-way binding]

Q4. Which one is string interpolation binding?

- A. `[target]="expression"`
- B. `{{expression}}`
- C. `(target) ="expression"`
- D. `[(target)] ="expression"`

Answer: B

[Page: 244]

Q5. Which Data Binding expression indicates that the data flows from the target to the destination specified by the expression?

- A. `[target]="expression"`
- B. `{{expression}}`
- C. `(target) ="expression"`
- D. `[(target)] ="expression"`

Answer: C

[Page: 244]

Q6. Which Data Binding expression indicates that the data flows in both directions between the target and the destination specified by the expression?

- A. `[target]="expression"`
- B. `{{expression}}`
- C. `(target) ="expression"`
- D. `[(target)] ="expression"`

Answer: D

[Page: 244]

Q7. Which one is null conditional operator?

- A. #
- B. ?
- C. ??
- D. ||

Answer: B

[Page: 251]

Q8. You want to bind the name property of the member model object if not null otherwise show text "input name".

Which template does this with null conditional operator?

- A. `<input [value]="member.name" placeholder='input name' />`
- B. `<input [value]="member.name || 'input name'" />`
- C. `<input [value]="member?.name || 'input name'" />`
- D. `<input [value]="member.name ? 'input name'" />`

Answer: C

[Page: 245]

Q9. Which binding does not work?

- A. `<div>{{model.fullName}}</div>`
- B. `<div>Full Name: {{model.fullName}}</div>`
- C. `<div [text]="model.fullName"></div>`
- D. `<div [textContent]="Full Name: ' + model.fullName"></div>`

Answer: C

[Page: 245]

[Angular binding supports DOM property not standard
Not all standard attributes are not supported]

Q10. Which property is used to set the content of HTML elements?

- A. text
- B. content
- C. html
- D. textContent

Answer: D

[Page: 246]

Q11. Which attribute binding will generate errors?

- A. `<td [colspan]="1+1">merged</td>`
- B. `<td [colSpan]="1+1">merged</td>`
- C. `<td [attr.colspan]="1+1">merged</td>`
- D. All of the above

Answer: A

[Page: 248]

[Angular supports DOM property
colspan doesn't work but colSpan does (camel casing colSpan, textContent)
But attr.colspan is ok]

Q12. Which one does not properly set CSS class to div element evaluating the getClass method?

- A. `<div [class]="getClass ()">div content</div>`
- B. `<div [style.class]="getClass ()">div content</div>`

-
- C. `<div [attr.class]="getClass ()">div content</div>`
 - D. `<div [ngClass]="getClass ()">div content</div>`

Answer: B

[Page: 249]

Q13. Which one append a class "text-danger" to an element without overwriting the classes already on the element when the template expression evaluates to truthy?

- A. `<div class="text-right" [class]="isLow? 'text-danger': "">div content</div>`
- B. `<div class="text-right" [attr.class]="isLow? 'text-danger': "">div content</div>`
- C. `<div class="text-right" [class.text-danger]="isLow">div content</div>`
- D. `<div class="text-right" [ngClass]="isLow">div content</div>`

Answer: C

[Page: 251]

Q14. You have an item in your component

`fontSize:string="10";`

You want an element's font size in pixel to the `fontSize` member value.

What template expression should you write?

- A. `first`
- B. `first`
- C. `first`
- D. All of the above

Answer: B

[Page: 257]

Chapter 13: Using the Built-in Directives

14 July 2019

Q1. Which built-in directive is used to include an element and its content in the HTML document if the expression evaluates as true?

- A. `ngIf`
- B. `ngSwitch`
- C. `ngTemplate`
- D. `ngFor`

Answer: A

[Page-264]

Q2. Which built-in directive is used to choose between multiple elements to include in the HTML document based on the result of an expression?

- A. `ngIf`
- B. `ngSwitch`
- C. `ngTemplate`
- D. `ngFor`

Answer: B

[Page-264]

Q3. Which one is the correct syntax of `ngFor` directive?

- A. `<div *ngFor="let item in expr"></div>`
- B. `<div *ngFor="let item of expr"></div>`
- C. `<div *ngFor="let item to expr"></div>`
- D. `<div *ngFor="let item from expr"></div>`

Answer: B
[Page-270]

Q4. Which one is not a ngFor supported local template value?

- A. index
- B. current
- C. odd
- D. even

Answer: B

[Page-272]

index, odd, even, first, last]

Q5. How can you improve ngFor template recreation?

- A. using index with ngFor
- B. using trackBy with ngFor
- C. using local cache of template
- D. using ngIf with with ngFor

Answer: B

[Page-279]

Q6. Which one is used to repeat a block of content at a specified location?

- A. ngTemplate directive
- B. ngTemplateOutlet directive
- C. ngTemplateOutletContext directive
- D. ngSwitch directive

Answer: B

[Page-280]

Q7. Which one should you use when you need to generate the same content in different places?

- A. ngTemplate directive
- B. ngTemplateOutlet directive
- C. ngTemplateOutletContext directive
- D. ngSwitch directive

Answer: B

[Page-280]

Q8. You have model element in you component

count: number = 0;

Now which binding expression will produce error in your template?

- A. {{count}}
- B. {{count+1}}
- C. Count: {{count}}
- D. {{count++}}

Answer: D

[Page-284]

No assignment is allowed

Also one-way data bindings cannot modify data after their expressions are evaluated]

Q9. You have model element in you component

price: number = 49.99;

discount:number = .10

Now which binding expression will produce error in your template?

- A. {{price-price*discount}}

-
- B. `{{(1-price)*discount}}`
 - C. `{{Math.floor (price-price*discount)}}`
 - D. Current price: `{{(1-price)*discount}}`

Answer: C

[Page-287]

Cannot access outside of component in binding

Cannot access global namespace in binding]

Chapter 14: Using Events and Forms

23 July 2019

Q1. Which one use to events binding?

- A. Round brackets
- B. Angle brackets
- C. Curly braces
- D. Banana in a box, [(event)]

Answer: A

[Page: 295]

Q2. Which character is used to define template reference variables?

- A. ?
- B. #
- C. \$
- D. &

Answer: B

[Page: 301]

Q3. Which one correctly uses template reference variable?

- A. `<input id="t" />{{t.value}}`
- B. `<input id="t" />{{#t.value}}`
- C. `<input ame="t" />{{t.value}}`
- D. `<input #t />{{t.value}}`

Answer: D

[Page: 301]

Q4. You want to pass a text input element's value to method when clicked on a button. Which code does this correctly?

- A. `<input type="text" id="t">`
`<button (click)="show(t)">Show</button>`
- B. `<input type="text" name="t">`
`<button (click)="show(t)">Show</button>`
- C. `<input type="text" [(ngModel)]= "t">`
`<button (click)="show(t)">Show</button>`
- D. `<input type="text" #t>`
`<button (click)="show(t)">Show</button>`

Answer: D

[Page: 307]

Q5. Which template properly sets two-way binding to a model element?

- A. `<input [(input)]= "newProduct.name" [(value)]= "newProduct.name"/>`
- B. `<input [ngModel]= "newProduct.name" />`

-
- C. `<input (ngModel)="newProduct.name" />`
 - D. `<input [(ngModel)]="newProduct.name" />`

Answer: D

[Page: 307]

Q6. Which one is not a built-in angular validation attribute?

- A. `required`
- B. `minlength`
- C. `maxlength`
- D. `number`

Answer: D

[Page: 309]

`required minlength maxlength pattern`]

Q7. Which angular form validation class is assigned if element's contents have not been changed by the user?

- A. `ng-untouched`
- B. `ng-touched`
- C. `ng-pristine`
- D. `ng-dirty`

Answer: C

[Page: 310]

Q8. Which angular form validation class is assigned if element's contents have been changed by the user but the user then returns to the previous contents?

- A. `ng-untouched`
- B. `ng-touched`
- C. `ng-pristine`
- D. `ng-dirty`

Answer: D

[Page: 314]

Q9. Which template is appropriate to work field level validation?

- A. `<input name="name" [(ngModel)]="newProduct.name" required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`
- B. `<input name="name" [(ngModel)]="newProduct.name" id="name" required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`
- C. `<input name="name" [(ngModel)]="newProduct.name" #name="ngModel" required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`
- D. `<input name="name" [(ngModel)]="newProduct.name" #name required />`
`<div class="text-danger" *ngIf="name.dirty && name.invalid && name.errors.required">You must enter a product name </div>`

Answer: C

[Page: 313]

Chapter 15: Creating Attribute Directives

14 July 2019

Q1. What is required to create an attribute directive?

- A. Create a class with name suffixed with Attribute
- B. Create a class and save it in src/directives folder
- C. Create a class that inherits from Element Ref class
- D. Create a class decorated with @Directive decorator

Answer: D

[Page: 340]

Q2. You created a directive using code

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

```
@Directive({  
  selector: '[app-highlight]'  
})  
export class HighlightDirective {  
  constructor() { /*...*/ }  
}
```

How do you use the directive?

- A. `text`
- B. `text`
- C. `text`
- D. `text`

Answer: C

[Page: 345]

Q3. You want to create an attribute directive that will make an element's black color context to be yellow. Your code is like below

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

```
@Directive({  
  selector: '[appHighlight]'  
})  
export class HighlightDirective {  
  constructor(el: ElementRef) {  
  
  }  
}
```

Which code should you use in the constructor?

- A. `el.backgroundColor = 'yellow';`
- B. `el.style.backgroundColor = 'yellow';`
- C. `el.nativeElement.backgroundColor = 'yellow';`
- D. `el.nativeElement.style.backgroundColor = 'yellow';`

Answer: D

[Page: 346]

Q4. Which directive is used to read host element's attribute value in an attribute directive?

- A. @Directive
- B. @Input
- C. @Attribute
- D. @Value

Answer: C

[Page: 347]

Q5. You want to create an attribute directive that will read a highlight-class attribute value from host element
Your code is like below

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

```
@Directive({
  selector: '[app-highlight]'
})
export class HighlightDirective {
  constructor(el: ElementRef) {}
}
```

How should you change the constructor?

- A. constructor(el: ElementRef, highlight-class: string) {
- B. constructor(el: ElementRef, @Attribute() hClass: string) {}
- C. constructor(el: ElementRef, @Attribute("string") hClass: string) {}
- D. constructor(el: ElementRef, @Attribute("highlight-class") hClass: string) {}

Answer: D

[Page: 346]

Q6. You want to create an attribute directive that will make an element's black color context to be yellow.
Your code is like below

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

```
@Directive({
  selector: '[appHighlight]'
})
export class HighlightDirective {
  @Input() highlightColor: string;
  constructor(el: ElementRef) {
    el.nativeElement.style.backgroundColor = highlightColor: string;
  }
}
```

Which code should use the directive correctly?

- A. <p appHighlight highlightColor="orange">orange</p>
- B. <p appHighlight [highlightColor]="orange">orange</p>
- C. <p appHighlight (highlightColor)="orange">orange</p>
- D. <p appHighlight [(highlightColor)]="orange">orange</p>

Answer: B

[Page: 348]

Chapter 16: Creating Structural Directives

14 July 2019

Q1. Which directive change the layout of the HTML document by adding and removing elements?

- A. Attribute directive
- B. Structural directive
- C. Conditional directive
- D. Repeatable directive

Answer: B

[Page 365]

Q2. Which @angular/core symbols you import to create structural directive?

- A. Input
- B. TemplateRef
- C. ViewContainerRef
- D. ElementRef

Answer: A, B, C

[Page 368]

Q3. You created a structural directive using the code

```
import { Directive, Input, TemplateRef, ViewContainerRef } from '@angular/core';
```

```
export class UnlessDirective {  
}
```

Which code you should you add above the class?

- A. @Directive()
- B. @Directive({ template: '[appUnless]'})
- C. @Directive({ selector: '[appUnless]'})
- D. @Directive({ style: '[appUnless]'})

Answer: C

[Page 368]

Q4. Which method of the ViewContainerRef is called to create a new view using a template?

- A. createEmbeddedView
- B. Insert
- C. element
- D. indexOf

Answer: A

[Page 370]

Q5. Which method of the ViewContainerRef is called to remove all the views from the container?

- A. createEmbeddedView
- B. clear
- C. remove
- D. indexOf

Answer: B

[Page 370]

Q6. How do you tell angular that the directive needs to query the host element's content and assign the first result of the query to the property?

- A. using @Attribute directive
- B. using @Input directive
- C. using @ContentChild directive
- D. using @Output directive

Answer: C

[Page 393]

Chapter 17: Understanding Components

14 July 2019

Q1. What is the root component in an angular application?

- A. the module that you bootstrap to launch the application
- B. the entry point specified in the Angular module.
- C. the component that loads other components
- D. the component has one or more child component

Answer: B

[Page: 403]

Q2. How do you describe a component?

- A. Using the declarations property of the @NgModule decorator of the module that encapsulates it
- B. Using the @Component directive
- C. Using the selector property
- D. Using the templateUrl property

Answer: B

[Page: 405]

Q3. Which property of the @Component decorator is used to specify the CSS selector used to match host elements?

- A. styles
- B. styleUrls
- C. selector
- D. template

Answer: C

[Page: 405]

Q5. Which property of the @Component decorator is used to specify an inline template?

- A. selector
- B. Template
- C. templateUrl
- D. styles

Answer: B

[Page: 405]

Q6. What is required to do to use a component in a module?

- A. Bring the component in scope using an import statement
- B. Add the component to the NgModule decorator's declarations array
- C. Add the component to the NgModule decorator's imports array
- D. Add the component to the NgModule decorator's providers array

Answer: A, B

[Page: 406]

Q7. Which component decoration is correct?

- A. @Component({selector: "<app></app>", templateUrl: "<div>loading.</div>"})
- B. @Component({selector: "<app></app>", template: "<div>loading.</div>"})
- C. @Component({selector: "<app></app>", styles: "<div>loading.</div>"})
- D. @Component({selector: "<app></app>", styleUrls: "<div>loading.</div>"})

Answer: B

[Page: 409]

Q8. How do you pass data from parent component to child component?

- A. Using Input properties
- B. Using Output properties
- C. Using attribute directive
- D. Using structural directive

Answer: A
[Page: 411]

- Q9. How do child components define custom events to signal important changes to parent components?
- A. Using Input properties
 - B. Using Output properties
 - C. Using attribute directive
 - D. Using structural directive

Answer: B
[Page: 415]

- Q10. Which event declaration is correct?
- A. `@Output() newName = event;`
 - B. `@Output() newName = $event;`
 - C. `@Output() newName = $event.value;`
 - D. `@Output() newName = new EventEmitter<string>();`

Answer: D
[Page: 415]

Chapter 18: Using and Creating Pipes

14 July 2019

- Q1. What are pipes in angular?
- A. small fragments of code to transform data values to be displayed in templates
 - B. small fragments of code to filter out data based on logic supplied
 - C. properties to pass data from parent component to child component
 - D. custom events that are used by child components to signal important changes to parent components

Answer: A

- Q2. Which character is used to apply pipes?
- A. ?
 - B. \$
 - C. =>
 - D. |

Answer: D

- Q3. Which pipe shows price with us \$ symbol?
- A. `{{item.price | currency }}`
 - B. `{{item.price | currency:"USD" }}`
 - C. `{{item.price | currency:"USD":"symbol" }}`
 - D. `{{item.price | currency:"$" }}`

Answer: C

- Q4. Which decorator should you use to create a custom pipe?
- A. `@Dective()`
 - B. `@Input()`
 - C. `@Pipe()`
 - D. `@Transform`

Answer: C

- Q5. Which method you must override in your custom pipe?

-
- A. transform method that accepts an ElementRef and returns the transformed value
 - B. transform method that accepts an Attribute and returns the transformed value
 - C. transform method that returns the transformed value
 - D. transform method that accepts an input value followed by optional parameters and returns the transformed value

Answer: D

Q6. Which @Pipe decorator property controls that this pipe is reevaluated only when its input value or its arguments are changed?

- A. name
- B. pure
- C. input
- D. output

Answer: B

[When true]

Q7. You created a custom pipe using the code

```
import { Pipe, PipeTransform } from '@angular/core';
@Pipe({name: 'exponentTo'})
export class ExponentToPipe implements PipeTransform {
  transform(value: number, exponent?: number): number {
    return Math.pow(value, isNaN(exponent) ? 1 : exponent);
  }
}
```

How do you use the pipe?

- A. {{2 | exponentTo (10)}}
- B. {{2 | exponentTo , exponent10}}
- C. {{2 | exponentTo : 10}}
- D. {{2 | exponentTo, 10}}

Answer: C

Q8. which pipe transforms an object into a JSON string?

- A. toString
- B. json
- C. parse
- D. stringify

Answer: B

Q9. Which pipe subscribes to an observable or promise?

- A. json
- B. slice
- C. async
- D. promise

Answer: C

[You used this in class lessons *ngFor="let p of getProducts()|async"]

Q10. What is the correct format number pipe?

- A. <minFactionDigits>-<maxFactionDigits>
- B. <minIntegerDigits>.<maxFactionDigits>
- C. <minIntegerDigits>.<minFactionDigits>
- D. <minIntegerDigits>.<minFactionDigits>-<maxFactionDigits>

Answer: D