CountPairs : https://codingbat.com/prob/p154048

public int countPairs(String str) {

if(str.length() <= 2)

return 0;

if(str.charAt(0) == str.charAt(2))

return 1 + countPairs(str.substring(1));

return countPairs(str.substring(1));

}

count7: https://codingbat.com/prob/p101409

public int count7(int n)

{

int c = 0;

if (7 > n)

{

return 0;

}

else

{

if ( 7 == n % 10)

{

c = 1;

}

else

{

c = 0;

}

}

return c + count7(n / 10);

}

OR

public int count7(int n)

{

return (7 > n) ? 0 : ( ( 7 == n % 10) ? 1 + count7(n / 10) : 0 + count7(n / 10));

}

Blackjack: https://codingbat.com/prob/p117019

public int blackjack(int a, int b) {

if (a > 21 && b > 21) {

return 0;

}else if (a > 21) {

return b;

} else if (b > 21) {

return a;

}

int sumA = 21 - a;

int sumB = 21 - b;

if (sumA > sumB) {

return b;

} else {

return a;

}}

answerCell: https://codingbat.com/java/Logic-1

public boolean answerCell(boolean isMorning, boolean isMom, boolean isAsleep) {

if(isAsleep)

return false;

if(isMorning && !isMom)

return false;

return true;

}

wordsWithoutList: https://codingbat.com/prob/p183407

public List wordsWithoutList(String[] words, int len) {

ArrayList<String> list = new ArrayList<String>();

for(int i = 0; i < words.length; i++) {

if(words[i].length() != len)

list.add(words[i]);

}

return list;

}

(https://stackoverflow.com/questions/74202343/added-border-in-css-go-over-the-last-div)

JS Fiddle Question:

<https://jsfiddle.net/q6pmaebj/>

**BuilderDesignPattern**

Method chains() like steam().filter().map().collect()

**2. What are the Spring Boot key components?**

Below are the four key components of spring-boot:

* Spring Boot auto-configuration.
* Spring Boot CLI.
* Spring Boot starter POMs.
* Spring Boot Actuators

## Angular HttpClient Methods

* request()
* delete()
* get()
* patch()
* post()
* put()
* head()
* jsonp()
* options()

To make the HTTP request to communicate with the server, we first import the HttpClientModule service in our angular app. (@angular/common/http)

How to achive relationship between two entities in Hibernate

* What can you tell us about yourself? ...
* How would a colleague describe you? ...
* Have you ever disagreed with a boss or colleague? ...
* Why do you want to work for us? ...
* Why should we hire you? ...
* Do you have any questions for us?

**What is COALESCE function?**

COALESCE function is used to return the value which is set to be not null in the list. If all values in the list are null, then the coalesce function will return NULL.

Coalesce(value1, value2,value3,…)

### What is BLOB datatype?

A BLOB data type is a varying length binary string which is used to store **two gigabytes memory**. Length should be specified in Bytes for BLO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmpId** | **FullName** | **ManagerId** | **DateOfJoining** | **City** |
| 121 | John Snow | 321 | 01/31/2014 | Toronto |
| 321 | Walter White | 986 | 01/30/2015 | California |
| 421 | Kuldeep Rana | 876 | 27/11/2016 | New Delhi |

1. [Write an SQL query to fetch all the Employees who are also managers.](https://artoftesting.com/sql-queries-for-interview#employee_also_manager)

SELECT DISTINCT E.FullName FROM EmployeeDetails E INNER JOIN EmployeeDetails M ON E.EmpID = M.ManagerID;

**Write an SQL query to fetch duplicate records from EmployeeDetails (without considering the primary key – EmpId**).

SELECT FullName, ManagerId, DateOfJoining, City, COUNT(\*) FROM EmployeeDetails GROUP BY FullName, ManagerId, DateOfJoining, City HAVING COUNT(\*) > 1;

**Write an SQL query to fetch only even rows from the table.**

SELECT \* FROM EmployeeDetails WHERE MOD (EmpId, 2) = 0;

**Write an SQL query to create a new table with data and structure copied from another table.**

CREATE TABLE NewTable SELECT \* FROM EmployeeSalary;

**Write an SQL query to find the nth highest salary from table.**

**SELECT TOP 1 Salary FROM ( SELECT DISTINCT TOP N Salary FROM Employee ORDER BY Salary DESC ) ORDER BY Salary ASC;**

OR

SELECT Salary FROM Employee ORDER BY Salary DESC LIMIT N-1,1;

### What are starter dependencies?

Spring boot starter is a maven template that contains a collection of all the relevant transitive dependencies that are needed to start a particular functionality.  
Like we need to import spring-boot-starter-web dependency for creating a web application.

<dependency>

<groupId> org.springframework.boot</groupId>

<artifactId> spring-boot-starter-web </artifactId>

</dependency>

### What are the most common Spring Boot CLI commands?

-run, -test, -grap, -jar, -war, -install, -uninstall, --init, -shell, -help.

To check the description, run spring --help from the terminal.

### What Are the Basic Annotations that Spring Boot Offers?

The primary annotations that Spring Boot offers reside in its org.springframework.boot.autoconfigure and its sub-packages. Here are a couple of basic ones:

@EnableAutoConfiguration – to make Spring Boot look for auto-configuration beans on its classpath and automatically apply them.

@SpringBootApplication – used to denote the main class of a Boot Application. This annotation combines @Configuration, @EnableAutoConfiguration, and @ComponentScan annotations with their default attributes

### How to disable a specific auto-configuration class?

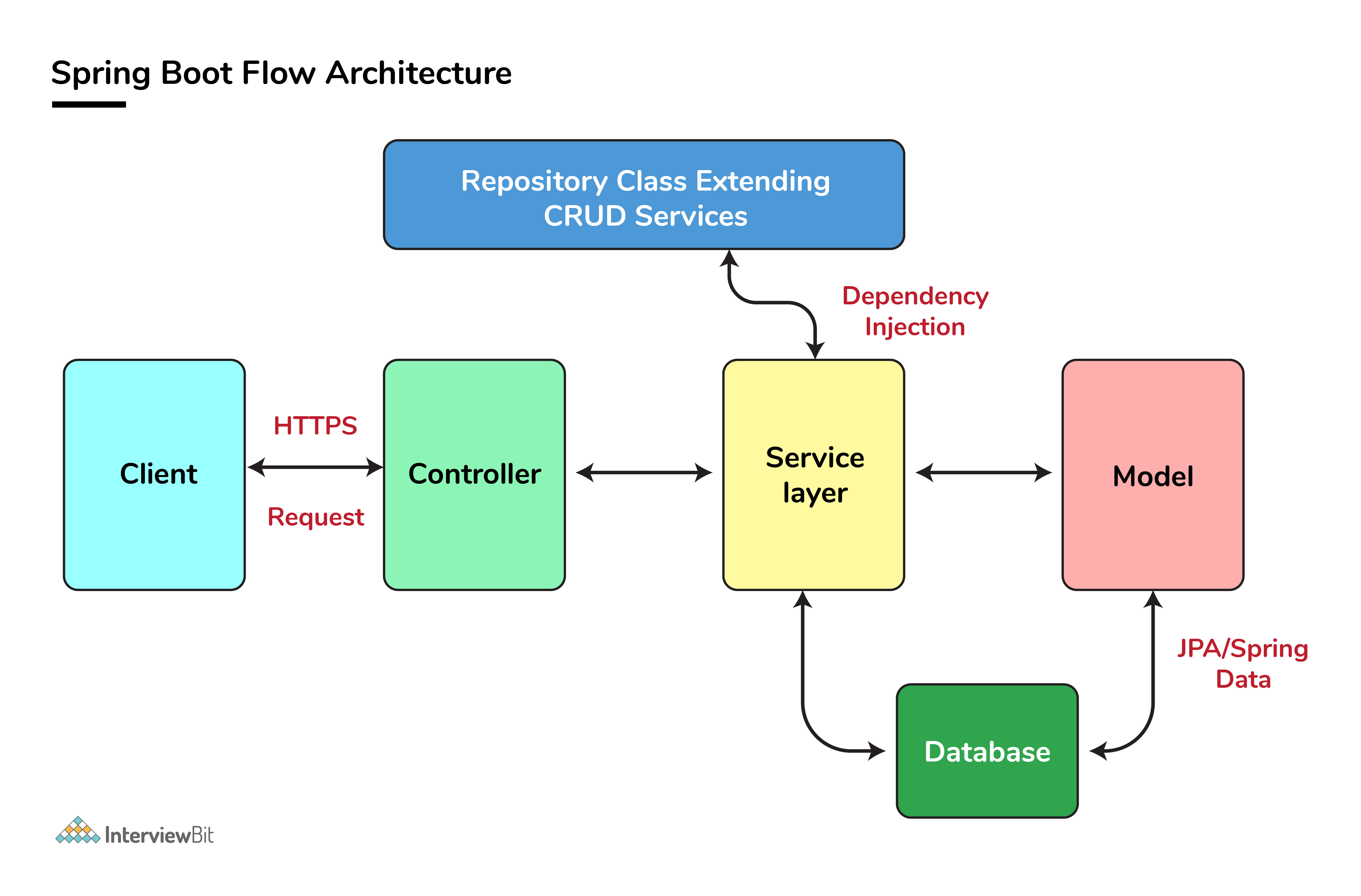
You can use exclude attribute of @EnableAutoConfiguration if you want auto-configuration not to apply to any specific class.

//use of exclude

@EnableAutoConfiguration(exclude={className})

### Describe the flow of HTTPS requests through the Spring Boot application?

On a high-level spring boot application follow the MVC pattern which is depicted in the below flow diagram.

Spring Boot Flow Architectur

### What is Spring Actuator? What are its advantages?

An actuator is an additional feature of Spring that helps you to monitor and manage your application when you push it to production. These actuators include auditing, health, CPU usage, HTTP hits, and metric gathering, and many more that are automatically applied to your application

### How to enable Actuator in Spring boot application?

To enable the spring actuator feature, we need to add the dependency of “spring-boot-starter-actuator” in pom.xml.

<dependency>

<groupId> org.springframework.boot</groupId>

<artifactId> spring-boot-starter-actuator </artifactId>

</dependency>

**What are the actuator-provided endpoints used for monitoring the Spring boot application?**

Actuators provide below pre-defined endpoints to monitor our application -

* Health
* Info
* Beans
* Mappings
* Configprops
* Httptrace
* Heapdump
* Threaddump
* Shutdown

### How to check the environment properties in your Spring boot application?

Spring Boot actuator “/env” returns the list of all the environment properties of running the spring boot application.

**What is dependency Injection?**

The process of injecting dependent bean objects into target bean objects is called dependency injection.

* Setter Injection: The IOC container will inject the dependent bean object into the target bean object by calling the setter method.
* Constructor Injection: The IOC container will inject the dependent bean object into the target bean object by calling the target bean constructor.
* Field Injection: The IOC container will inject the dependent bean object into the target bean object by Reflection API.

**What is the starter dependency of the Spring boot module?**

Spring boot provides numbers of starter dependency, here are the most commonly used -

* Data JPA starter.
* Test Starter.
* Security starter.
* Web starter.
* Mail starter.
* Thymeleaf starter

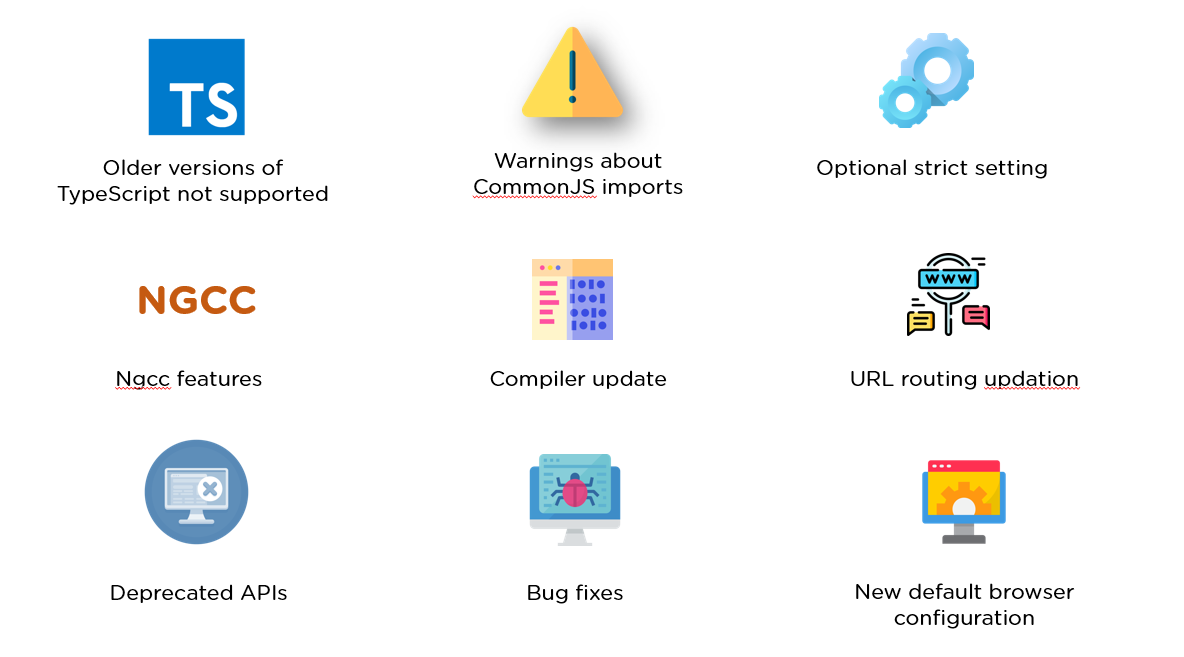
### Differentiate between Angular and AngularJS

What are decorators in Angular?

Decorators are a design pattern or functions that define how Angular features work. They are used to make prior modifications to a class, service, or filter. Angular supports four types of decorators, they are:

1. Class Decorators
2. Property Decorators
3. Method Decorators
4. Parameter Decorators

### What are the new updates with Angular10?



What are Directives in Angular?

Directives are attributes that allow the user to write new HTML syntax specific to their applications. They execute whenever the Angular compiler finds them in the DOM. Angular supports three types of directives.

1. Component Directives
2. Structural Directives
3. Attribute Directives

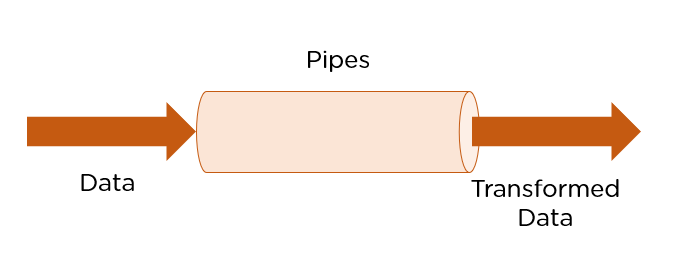
 What is an AOT compilation? What are its advantages?

The Ahead-of-time (AOT) compiler converts the Angular HTML and TypeScript code into JavaScript code during the build phase, i.e., before the browser downloads and runs the code.

Some of its advantages are as follows.

1. Faster rendering
2. Fewer asynchronous requests
3. Smaller Angular framework download size
4. Quick detection of template errors
5. Better security

### What are Pipes in Angular?



[Pipes are simple functions](https://www.simplilearn.com/tutorials/angular-tutorial/angular-pipes) designed to accept an input value, process, and return as an output, a transformed value in a more technical understanding. Angular supports several built-in pipes. However, you can also create custom pipes that cater to your needs.

Some key features include:

1. Pipes are defined using the pipe “|” symbol.
2. Pipes can be chained with other pipes.
3. Pipes can be provided with arguments by using the colon (:) sign.

### What are Pure Pipes?

These pipes are pipes that use pure functions. As a result of this, a pure pipe doesn't use any internal state, and the output remains the same as long as the parameters passed stay the same. Angular calls the pipe only when it detects a change in the parameters being passed. A single instance of the pure pipe is used throughout all components.

### 17. What are Impure Pipes?

For every change detection cycle in Angular, an impure pipe is called regardless of the change in the input fields. Multiple pipe instances are created for these pipes. Inputs passed to these pipes can be mutable.

By default, all pipes are pure. However, you can specify impure pipes using the pure property, as shown below.

@Pipe({

name: 'demopipe',

pure : true/false

})

export class DemopipePipe implements PipeTransform {

What are filters in Angular? Name a few of them.

Filters are used to format an expression and present it to the user. They can be used in view templates, controllers, or services. Some inbuilt filters are as follows.

* date - Format a date to a specified format.
* filter - Select a subset of items from an array.
* Json - Format an object to a JSON string.
* limitTo -  Limits an array/string, into a specified number of elements/characters.
* lowercase - Format a string to lowercase.

Explain the lifecycle hooks in Angular

In Angular, every component has a lifecycle. Angular creates and renders these components and also destroys them before removing them from the DOM. This is achieved with the help of lifecycle hooks. Here's the list of them -

1. ngOnChanges() - Responds when Angular sets/resets data-bound input properties.
2. ngOnInit() - Initialize the directive/component after Angular first displays the data-bound properties and sets the directive/component's input properties/
3. ngDoCheck() - Detect and act upon changes that Angular can't or won't detect on its own.
4. ngAfterContentInit() - Responds after Angular projects external content into the component's view.
5. ngAfterContentChecked() - Respond after Angular checks the content projected into the component.
6. ngAfterViewInit() - Respond after Angular initializes the component's views and child views.
7. ngAfterViewChecked() - Respond after Angular checks the component's views and child views.
8. ngOnDestroy - Cleanup just before Angular destroys the directive/component.

### What are Template statements?

Template statements are properties or methods used in HTML for responding to user events. With these template statements, the application that you create or are working on, can have the capability to engage users through actions such as submitting forms and displaying dynamic content.

For example,

<button (click)="deleteHero()">Delete hero</button>

The template here is deleteHero. The method is called when the user clicks on the button

### What is the difference between AOT and JIT?

Ahead of Time (AOT) compilation converts your code during the build time before the browser downloads and runs that code. This ensures faster rendering to the browser. To specify AOT compilation, include the --aot option with the ng build or ng serve command.

The Just-in-Time (JIT) compilation process is a way of compiling computer code to machine code during execution or run time. It is also known as dynamic compilation. JIT compilation is the default when you run the ng build or ng serve CLI commands.

**Functional Programming:**

**State**means

we have a variable whose value will change while program executes ex: Sum

Terminal operations what are available terminal operatiors available

like findFirst()

Intermediate Operations are Lazy

peek

filter

ect..

**Higher Order**Function

is a function that returns another Function as return value

**Constructer Reference**we can create new object something like this

String**::**new

**Method Reference (::)**

this can also work for both static and members(non static) as well.

**Type Inference**with Lambda expression we need not to specify the type of the variable it is automatically take care by Java compiler is called **Type Inference**

Ex: (x,y) -> x+y

(Integer x, Integer y) -> x+y

above both are same

**Supplier**Functional Interface is no input but return something back to us

Implementation something like empty paranthesis followed by lambda

() -> return statement

**Behavior**of the method as a argument is called **Behavior Parameterization**

**Function Descriptor**

Is a abstract method in Functional Interface.

Only one Abstract method in Functional Interface

Predicate

Function

Consumer

Terminal Operations returns something other than the stream.

these we use at the end of stream

ex: reduce

forEach

collect

etc

Intermediate Operations are called whatever that takes Stream and return Streams.

Ex: Distinct()

sorted()

map() etc

 Give a briefing on the life cycle of a thread.

The life cycle of a thread includes five stages, as mentioned below.

1. New Born State
2. Runnable State
3. Running State
4. Blocked State
5. Dead State

Explain the difference between >> and >>> operators.

Although they look similar, there is a massive difference between both.

* >> operator does the job of right shifting the sign bits
* >>> operator is used in shifting out the zero-filled bits

### What is the Daemon Thread?

The Daemon thread can be defined as a thread with the least priority. This Daemon thread is designed to run in the background during the Garbage Collection in Java.

The setDaemon() method creates a Daemon thread in Java.

### Why is Java is Dynamic?

Java is designed to adapt to an evolving environment. Java programs include a large amount of runtime information that is used to resolve access to objects in real-time.

### . How many times is the finalize method called?

The finalize method is called the Garbage collector. For every object, the Garbage Collector calls the finalize() method just for one time.

### What is the difference between BeanFactory and ApplicationContext?

BeanFactory is the **basic container** whereas ApplicationContext is the **advanced container**. ApplicationContext extends the BeanFactory interface. ApplicationContext provides more facilities than BeanFactory such as integration with spring AOP, message resource handling for i18n etc.

### What is the difference between constructor injection and setter injection?

|  |  |  |
| --- | --- | --- |
| **No.** | **Constructor Injection** | **Setter Injection** |
| 1) | No Partial Injection | Partial Injection |
| 2) | Desn't override the setter property | Overrides the constructor property if both are defined. |
| 3) | Creates new instance if any modification occurs | Doesn't create new instance if you change the property value |
| 4) | Better for too many properties | Better for few properties. |

What is autowiring in spring? What are the autowiring modes?

Autowiring enables the programmer to inject the bean automatically. We don't need to write explicit injection logic.

Let's see the code to inject bean using dependency injection.

1. <bean id="emp" **class**="com.javatpoint.Employee" autowire="byName" />

The auto-wiring modes are given below:

|  |  |  |
| --- | --- | --- |
| **No.** | **Mode** | **Description** |
| 1) | no | this is the default mode, it means autowiring is not enabled. |
| 2) | byName | injects the bean based on the property name. It uses setter method. |
| 3) | byType | injects the bean based on the property type. It uses setter method. |
| 4) | constructor | It injects the bean using constructor |

### What are the advantages of JdbcTemplate in spring?

**Less code**: By using the JdbcTemplate class, you don't need to create connection,statement,start transaction,commit transaction and close connection to execute different queries. You can execute the query directly.

### What is AOP?

AOP is an acronym for Aspect Oriented Programming. It is a methodology that divides the program logic into pieces or parts or concerns.

It increases the modularity and the key unit is Aspect.

### 19) What are the advantages of spring AOP?

AOP enables you to dynamically add or remove concern before or after the business logic. It is **pluggable** and **easy to maintain**.

### 20) What are the AOP terminology?

AOP terminologies or concepts are as follows:

* JoinPoint
* Advice
* Pointcut
* Aspect
* Introduction
* Target Object
* Interceptor
* AOP Proxy
* Weaving

### 21) What is JoinPoint?

JoinPoint is any point in your program such as field access, method execution, exception handling etc.

### 22) Does spring framework support all JoinPoints?

No, spring framework supports method execution joinpoint only.

### 23) What is Advice?

Advice represents action taken by aspect.

**Describe the differences between fail-fast and fail-safe iterators.** Compare fail-fast and fail-safe iterators in terms of their behavior when the underlying collection is modified during iteration.