

Real Junior Developer Interview Questions

From the video: <https://youtu.be/5NmPv42EwZ0>

Here is a list of the questions I talk about in the above video and some samples of the answers I gave when I took the interview.

I have tried to be as concise as possible with the answers and you might want to look into and elaborate on some of these points if the interviewer is looking for a more detailed response. Chances are you might only get asked a couple of technical questions in your interview but they could come from a broad range of backgrounds so having good all-round knowledge is essential.

Let me know if you have any questions and maybe send me a Tweet with your thoughts on these interview questions!

James

([@codebubb](#))

Question 1

Create an algorithm that prints the integers from 17 to 53. However for multiples of two print "Foo" instead of the number and for the multiples of five print "Bar". For numbers which are multiples of both two and five print "FooBar".

Pseudocode

Set counter to 17

Loop whilst counter is less than 54

 if counter modulus 2 is equal to 0 and counter modulus 5 is equal to 0

 print "FooBar"

 else if counter modulus 2 is equal to 0

 print "Foo"

 else if counter modulus 5 is equal to 0

 print "Bar"

 else

 print counter

JavaScript

```
for(var i=17; i< 54; i++){
  if(i % 2 === 0 && i % 5 === 0){
    console.log("FooBar");
  } else if(i % 2 === 0){
    console.log("Foo");
  } else if(i % 5 === 0){
    console.log("Bar");
  } else {
    console.log(i);
  }
}
```

Question 2

In SQL database tables, why is redundant data (i.e. the same data stored in multiple tables) generally a bad thing?

Redundant data may be considered bad as there are multiple places to keep track of data thereby making it harder to maintain consistency. For example, if a customer's address is stored in multiple tables and/or columns within the database then every instance of this would need to be updated if the customer's address changed. It would make more sense to store the address in one place (e.g. 'customers' table) and then refer back to it when required.

Question 3

In SQL database tables, why might redundant data be necessary in real world applications?

If the database is un-normalised then there may be a possibility of redundant data. The reason for having un-normalised data may be to make it easier to query the data or to allow for complex data to be stored in a row.

Question 4

Why won't the following Apache web server configuration work correctly? Can you suggest a solution to make it work?

Listen 33.33.33.33:443

```
<VirtualHost 33.33.33.33:443>
```

```
ServerName www.example1.com
```

```
DocumentRoot /var/html/www.example1.com
```

```
SSLEngine on
```

```
SSLCertificateFile /etc/apache2/ssl.crt/www.example1.com.crt
```

```
SSLCertificateKeyFile /etc/apache2/ssl.key/www.example1.com.crt
```

```
</VirtualHost>
```

```
<VirtualHost 33.33.33.33:443>
```

```
ServerName www.example2.com
```

```
DocumentRoot /var/html/www.example2.com
```

```
SSLEngine on
```

```
SSLCertificateFile /etc/apache2/ssl.crt/www.example2.com.crt
```

```
SSLCertificateKeyFile /etc/apache2/ssl.key/www.example2.com.crt
```

```
</VirtualHost>
```

I'm pretty sure that when I have installed SSL certs before they need to have their own unique IP address. The two entries have the same IP so one of them would need to be updated.

Question 5

In PHP, for what reasons might you initialise strings with single quotes (') instead of double quotes (")?

If you wanted to include double quotes within your strings in PHP then you would need to initialise the string with single quotes.

Question 6

In development teams multiple people are often involved in building and maintaining applications, often including the same set of files. They may be working on a single task together, or multiple tasks, and changes made by one developer may conflict with those of another. What system would you suggest to help manage this, and why would you choose your solution in particular?

Version control software such as Git or Subversion would handle the creation of software by a team of developers. Personally, I would choose Git as it is the technology I am most familiar with and

there are remote repository storage providers such as GitHub / BitBucket. These providers also integrate useful tools such as issue tracking.

Question 7

Modern websites often need to take into account the form of device a visitor is viewing the site on (e.g. desktop computer, tablet, smartphone). What design pattern would you suggest to make the site work across each device?

Responsive design. Using a framework such as Bootstrap/Foundation could be a quick way to make a site look good on all platforms it is being accessed on. Otherwise, custom media queries could be used to alter the way a site appears depending on device size.

Question 8

Explain what Object-oriented Programming is and the benefits of using an Object-oriented language.

Object-oriented programming is a paradigm that encourages developers to think of their application as a series of real objects. Data is usually encapsulated so that it can only be accessed by public methods (should the developer provide them). An object will also provide methods for interacting with itself and other objects. With OOP, each object is defined as a class and an instance of this class is created to form the object.

Some key benefits of OOP are: code-reusability, modularisation, inheritance (where a class is derived from another parent class), protection of data through encapsulation.

Question 9

In an ecommerce application, with discrete levels of stock, how would you manage a situation where two customers were interested in buying the last unit of stock for an item? How would your solution ensure the greatest possibility of the item being sold?

There would need to be some kind of locking in place to prevent multiple orders being placed for a single remaining item. If **customer A** was to place the item in their shopping basket, I would then remove the availability of that item to prevent **customer B** from also attempting to purchase. If **customer A** did not complete their transaction within a certain period of time (10 minutes for example) then the item would then be made available again. The locking period would depend on the likelihood of a collision such as this from occurring.

Furthermore, I think there are additional cues which could be added to the application to prevent this from happening. Visualising the remaining stock i.e. 'Last one in stock' or showing the number of visitors looking at the item in the last hour would encourage customers to purchase rather than procrastinating.

Question 10

Explain the benefits of using an MVC Framework

MVC is a design pattern that separates concerns in an application allowing different parts of the program to focus on a specific area. Using an MVC framework such as Laravel for PHP or Express for NodeJS allows an application to be created quickly and other developers who are familiar with the framework know where to look for things. Most MVC frameworks contain generators for quickly creating models, views and controllers and there are generally middleware packages (such as Passport for Express) that allow additional functionality to be added to the application quickly.

Finally, some MVC frameworks include database migrations which provide an element of version control to an application.