WAD IMP Questions for viva

1. What is html css and javascript?
2. What type of language is HTML CSS JavaScript?
   [Ans : HTML and CSS are markup languages whereas Javascript is a programming language. This means you can use it to store variables, carry out mathematical functions and perform many advanced functions.]
3. What is CSS used for? [Ans : CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.]
4. What are the three types of CSS? [Ans : here are three ways you can use to implement CSS into your HTML: internal, external, and inline styles.]

5. The Difference Between Inline, External and Internal CSS Styles

Ans: Internal CSS Internal or embedded CSS requires you to add **<style>** tag in the **<head>** section of your HTML document.

This CSS style is an effective method of styling a single page. However, using this style for multiple pages is time-consuming as you need to put CSS rules on every page of your website.

Here's how you can use internal CSS:

1. Open your HTML page and locate **<head>** opening tag.
2. Put the following code right after the **<head>** tag

```
<style type="text/css">
```

3. Add CSS rules on a new line. Here's an example:

```
body {
   background-color: blue;
```

```
}
h1 {
   color: red;
   padding: 60px;
}
```

4. Type the closing tag:

```
</style>
```

Your HTML file will look like this:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
   background-color: blue;
}
h1 {
   color: red;
   padding: 60px;
}
</style>
</head>
<body>

<h1>Hostinger Tutorials</h1>
<p>This is our paragraph.</p>

</body>
</html>
```

Advantages of Internal CSS:

- You can use class and **ID selectors** in this style sheet. Here's an example:

```
.class {
   property1 : value1;
   property2 : value2;
   property3 : value3;
}

#id {
   property1 : value1;
   property2 : value2;
   property3 : value3;
}
```

- Since you'll only add the code within the same HTML file, you don't need to upload multiple files.

Disadvantages of Internal CSS:

- Adding the code to the HTML document can increase the page's size and loading time.

External CSS

With external CSS, you'll link your web pages to an external **.css** file, which can be created by any text editor in your device (e.g., **Notepad++**).
This CSS type is a more efficient method, especially for styling a large website. By editing one **.css** file, you can change your entire site at once.
Follow these steps to use external CSS:

1. Create a new **.css** file with the text editor, and add the style rules. For example:

```
.xleftcol {
```

```
  float: left;
  width: 33%;
  background:#809900;
}
.xmiddlecol {
  float: left;
  width: 34%;
  background:#eff2df;
}
```

2. In the **<head>** section of your HTML sheet, add a reference to your external **.css** file right after **<title>** tag:

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

Don't forget to change **style.css** with the name of your **.css** file.

Advantages of External CSS:

- Since the CSS code is in a separate document, your HTML files will have a cleaner structure and are smaller in size.
- You can use the same **.css** file for multiple pages.

Disadvantages of External CSS:

- Your pages may not be rendered correctly until the external CSS is loaded.
- Uploading or linking to multiple CSS files can increase your site's download time.

Inline CSS

Inline CSS is used to style a specific HTML element. For this CSS style, you'll only need to add the **style** attribute to each HTML tag, without using selectors.

This CSS type is not really recommended, as each HTML tag needs to be styled individually. Managing your website may become too hard if you only use inline CSS.

However, inline CSS in HTML can be useful in some situations. For example, in cases where you don't have access to CSS files or need to apply styles for a single element only.

Let's take a look at an example. Here, we add an inline CSS to the **<p>** and **<h1>** tag:

```
<!DOCTYPE html>
<html>
<body style="background-color:black;">

<h1 style="color:white;padding:30px;">Hostinger Tutorials</h1>
<p style="color:white;">Something usefull here.</p>

</body>
</html>
```

Advantages of Inline CSS:

- You can easily and quickly insert CSS rules to an HTML page. That's why this method is useful for testing or previewing the changes, and performing quick-fixes to your website.
- You don't need to create and upload a separate document as in the external style.

Disadvantages of Inline CSS:

- Adding CSS rules to every HTML element is time-consuming and makes your HTML structure messy.
- Styling multiple elements can affect your page's size and download time.]

6. Explain meta tags in HTML

Answer

- **Meta tags** always go inside **head tag** of the HTML page
- **Meta tags** is always passed as name/value pairs
- **Meta tags** are not displayed on the page but intended for the browser
- **Meta tags** can contain information about **character encoding**, **description**, **title** of the document etc,

7. What Is Load Balancing?

Answer
Load balancing is simple technique for distributing workloads across multiple machines or clusters. The most common and simple load balancing algorithm is Round Robin. In this type of load balancing the request is divided in circular order ensuring all machines get equal number of requests and no single machine is overloaded or underloaded.

The Purpose of load balancing is to

- Optimize resource usage (avoid overload and under-load of any machines)
- Achieve Maximum Throughput
- Minimize response time

Most common load balancing techniques in web based applications are

1. Round robin
2. Session affinity or sticky session
3. IP Address affinity

8. Explain the CSS `box model` and the layout components that it consists of

Answer
The CSS box model is a rectangular layout paradigm for HTML elements that consists of the following:

- **Content** - The content of the box, where text and images appear
- **Padding** - A transparent area surrounding the content (i.e., the amount of space between the border and the content)
- **Border** - A border surrounding the padding (if any) and content
- **Margin** - A transparent area surrounding the border (i.e., the amount of space between the border and any neighboring elements)

9. What is *Sass*?

Answer

Sass or **Syntactically Awesome StyleSheets** is a *CSS* preprocessor that adds power and elegance to the basic language. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS-compatible syntax. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly.

A *CSS preprocessor* is a scripting language that extends CSS by allowing developers to write code in one language and then compile it into CSS.

10. What is a *CSS rule*?

Answer
Web browsers apply **CSS rules** to a document to affect how they are displayed. A CSS rule is formed from:

- A **set of properties**, which have values set to update how the HTML content is displayed,
- A **selector**, which selects the element(s) you want to apply the updated property values to.

A set of CSS rules contained within a stylesheet determines how a webpage should look.

11. What is the difference between span and div?

Answer

- div is a block element
- span is inline element

For bonus points, you could point out that it's illegal to place a block element inside an inline element, and that while div can have a p tag, and a p tag can have a span, it is not possible for span to have a div or p tag inside.

12. How does *concurrency* work in Node.js?

Answer

The thing with node.js is that *everything runs concurrently, except for your code*.

So, what that means is that there are actually lots of threads running inside Node.js virtual machine (or a thread pool if you wish), and those threads are utilized whenever you call an async function like performing i/o operations on files, accessing databases, requesting urls, etc.

However, for your code, there is only a single thread, and it processes events from an event queue. So, when you register a callback its reference is actually passed to the background worker thread, and once the async operation is done, new event is added to the event-queue with that callback

When Node gets I/O request it creates or uses a thread to perform that I/O operation and once the operation is done, it pushes the result to the **event queue**. On each such event, **event loop** runs and checks the queue and if the execution stack of Node is empty then it adds the queue result to execution stack.

This is how Node manages concurrency.

13. Name fundamental principles of design

Answer
The fundamental principles of design are:

- **BALANCE**—Balance in design is similar to balance in physics. A large shape close to the center can be balanced by a small shape close to the edge. Balance provides stability and structure to a design. It's the weight distributed in the design by the placement of your elements.
- **PROXIMITY**—Proximity creates a relationship between elements. It provides a focal point. Proximity doesn't mean that elements have to be placed together, it means they should be visually connected in some way.
- **ALIGNMENT**—Allows us to create order and organization. Aligning elements allows them to create a visual connection with each other.
- **REPETITION**—Repetition strengthens a design by tying together individual elements. It helps to create association and consistency. Repetition can create rhythm (a feeling of organized movement).
- **CONTRAST**—Contrast is the juxtaposition of opposing elements (opposite colors on the color wheel, or value light/dark, or direction—horizontal/vertical). Contrast allows us to emphasize or highlight key elements in your design.
- **SPACE**—Space in art refers to the distance or area between, around, above, below, or within elements. Both positive and negative space are important factors to be considered in every design.

1. Give the difference between HTML and HTML 5.0.
2. What is HTML? Give the reasons why we use HTML?
3. What are the types of lists used in HTML explain with suitable example?
4. How divide and pound symbol can be put on HTML document?
5. How will you create password filled in a HTML form
6. What is the difference between group of checkbox button and group of radio buttons?
7. What is CSS selector explain any three selectors with example?
8. Explain the need for CSS also in least features of CSS
9. Write a CSS rule that places a background image Halfway down the page tilting it horizontally the image should remain in place when the use scroll up or down.
10. Explain any eight CSS text properties.
11. Explain the bootstrap grid system in detail.
12. What does inline embedded and external JavaScript explain with advantages and drawbacks.
13. Explain JavaScript objects in detail.
14. How to create and modify the objects in JavaScript.
15. Explain the event handling in JavaScript.
16. Compare and contrast the traditional and web application architecture and ajax based web application architecture.
17. List the pros and cons of ajax
18. List and explain different methods for DOM manipulation with jQuery
19. List and explain the form selectors used in jQuery
20. What is the web Framework? Give the reason for using web framework.
21. List and explain the features of any three popular web frameworks.
22. Explain MVC architecture with a suitable diagram.
23. Enlist the features of MVC framework.
24. What is typescript? Give the advantages and disadvantages of using it.
25. List and explain the features of typescript.
26. What is Angular JS and explain its features.
27. What is the use of src folder and package.json file in Angular?
28. What is React JS enlist the features of it.
29. Write a simple react JS application that displays welcome user message.
30. Given the difference between functional and class components.
31. Explain the basic building blocks of reduction architecture with suitable block diagram.
32. What is hook? Explain the basic hooks in React JS with simple demo application.
33. What is node.JS? Enlist the features of node.js.
34. Explain any four methods of console object in node.js with suitable examples.

35. Explain the callbacks in node.js with suitable example.
36. State and explain any three built in models in node.js.
37. Explain the concept of NPM.
38. What is socket? Write a client server communication in node.JS using socket programming.
39. What is NoSQL? What is the need for it in least various features of NoSQL?
40. List and explain various features of MongoDB.
41. List and explain various steps for MongoDB node.JS communication.
42. Explain insertion and deletion database operation using node.JS.
43. Write a program to display the sorted data in the database using node.JS.
44. What is CRUDE? Explain the crude CRUDE in node.JS.
45. Write short note on Mongoose ODM.
46. List and explain the features of advantage advanced MongoDB.
47. What is replication? Enlist the advantages of replication in database system.
48. What is the purpose of map reduce? Explain it with suitable example.
49. Explain the concept of sharding.
50. What is mobile-first?
51. What is user interface?
52. What is mobile web?
53. What is difference between website and web app?
54. What are the mobile devices?
55. What is desktop device?
56. What is jQuery mobile?
57. Is jQuery mobile framework? If yes, explain it in brief.
58. What are the features of jQuery mobile?
59. What does mean by progressive enhancement in jQuery mobile?
60. How to style the web page in jQuery mobile?
61. What are the advantages and disadvantages of jQuery mobile?
62. How to setup jQuery mobile?
63. What are the required files to start with jQuery mobile?
64. What is the difference between jQuery and jQuery mobile?
65. What is CDN? How it is powerful in web development?
66. What is jQuery architecture?
67. What is “role” in jQuery mobile?
68. List all rolls in jQuery mobile.
69. What is the attribute name to define role?
70. What is page? Write a code to create a page in jQuery mobile.
71. How to define header and footer in jQuery mobile?
72. Write a code to create a header and footer in jQuery mobile.
73. What is content? Write a code to create content div in jQuery mobile

74. What is navigation? Write a code to navigate from one page to another page in jQuery mobile.
75. Write a code to create login form.
76. List any five CSS classes for jQuery mobile.
77. What is cloud computing?
78. What are the benefits of cloud computing?
79. What are the types of cloud computing?
80. What is AWS cloud?
81. What are the services provided by AWS?
82. What is EC2 service?
83. What are the EC2 types?
84. What are the advantages of Elastic beanstalk?
85. Difference between elastic beanstalk and EC2?
86. List steps to deploy the application on elastic beanstalk.