

1)What is the difference between “display: none” and “visibility: hidden”, when used as attributes to the HTML element.

Ans- display:none means that the tag in question will not appear on the page at all (although you can still interact with it through the dom). There will be no space allocated for it between the other tags.

visibility:hidden means that unlike display:none, the tag is not visible, but space is allocated for it on the page. The tag is rendered, it just isn't seen on the page.

So, the difference between **display: "none"**; and **visibility: "hidden"**; right from the name itself we can tell the difference as display: "none", completely gets rid of the tag, as it had never existed in the HTML page whereas visibility: "hidden"; just makes the tag invisible, it will still on the HTML page occupying space it's just invisible.

2)In how many ways can we specify the CSS styles for the HTML element?

Ans- There are three ways to add CSS to HTML. You can **add inline CSS** in a style attribute to style a single HTML element on the page. You can embed an **internal stylesheet** by adding CSS to the head section of your HTML doc. Or you can link to an **external stylesheet** that will contain all your CSS separate from your HTML.

3)What are forms and how to create forms in HTML?

Ans- HTML Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.

Forms contain special elements called controls like inputbox, checkboxes, radio-buttons, submit buttons, etc. Users generally complete a form by modifying its controls e.g. entering text, selecting items, etc. and submitting this form to a web server for further processing.

The **<form> tag** is used to create an HTML form

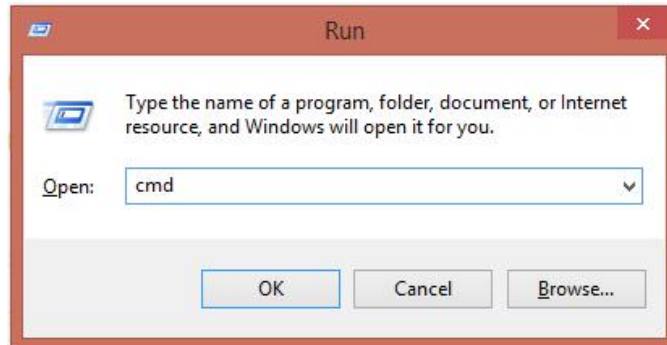
4)How to handle events in HTML?

Ans- Events are actions or occurrences that happen in the system you are programming, which the system tells you about so your code can react to them. .Basically, to handle events in HTML, you just need to add the function in the HTML tag which is going to be executed in JavaScript when any event in HTML is fired or triggered. For example, if the user clicks a button on a webpage, you might want to react to that action by displaying an information box.

5)How do you create a simple server in Node.js that returns Hello World?

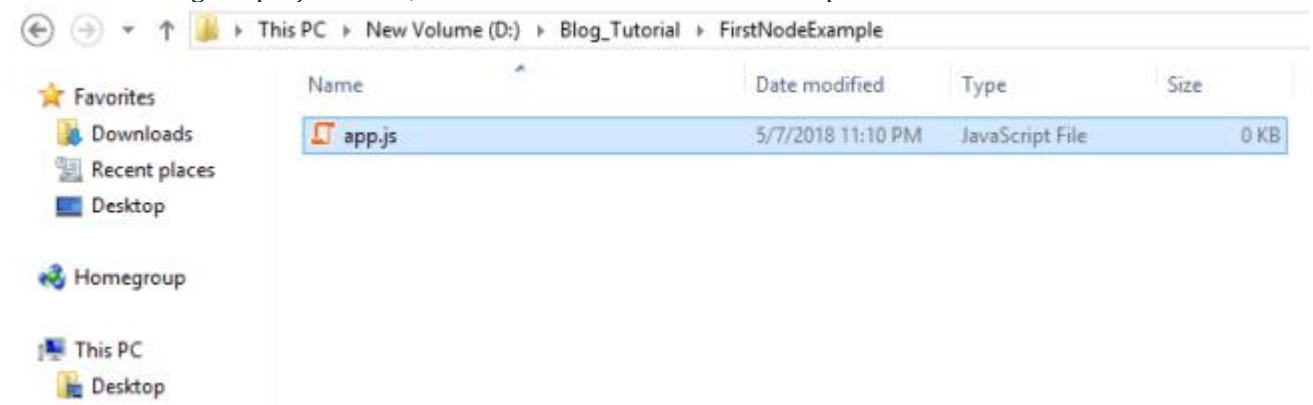
Ans- Step 1

First, open the command prompt by pressing Windows + R key and type cmd for opening the command prompt for running our sample application.



Step 2

After opening the command prompt, create one node.js project folder in any drive for saving this example file. For example, I am using D: drive Tutorial folder. Inside this folder, create one more folder for saving the project. Here, we will create the FirstNodeExample folder.



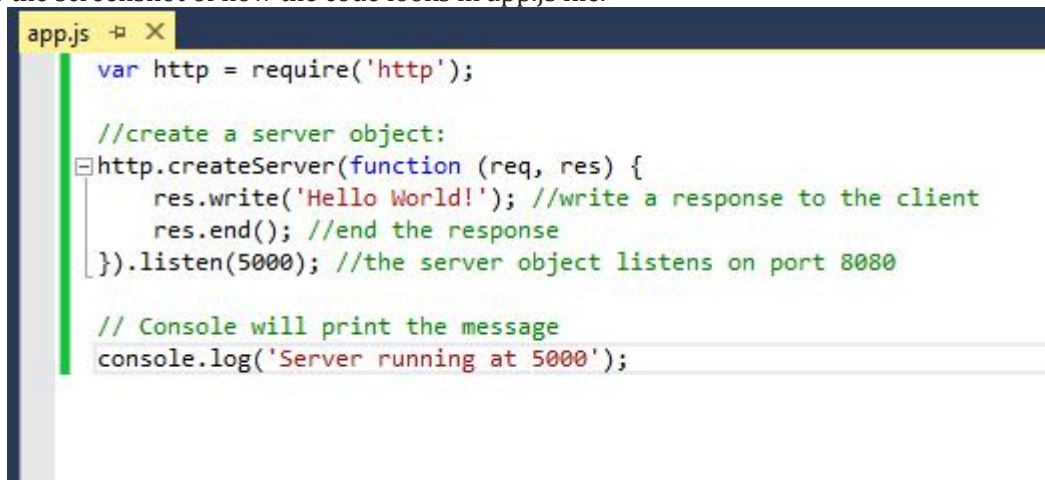
Step 3

Now, create one JavaScript file and name that app.js file inside FirstNodeExample folder. Now, open this file in any editor like Notepad, Notepad ++, Visual Studio. Here, I am using Visual Studio to write this code. Now, write the code in the app.js file and save that file.

Write the node.js code to display "Hello World!" in the browser.

```
1.     var http = require('http');
2.     //create a server object:
3.
4.     http.createServer(function (req, res) {
5.         res.write('Hello World!'); //write a response to the client
6.         res.end(); //end the response
7.     }).listen(5000); //the server object listens on port 8080
8.
9.     // Console will print the message
10.    console.log('Server running at 5000');
```


See the screenshot of how the code looks in app.js file.



```
app.js ✕  
var http = require('http');  
  
//create a server object:  
http.createServer(function (req, res) {  
  res.write('Hello World!'); //write a response to the client  
  res.end(); //end the response  
}).listen(5000); //the server object listens on port 8080  
  
// Console will print the message  
console.log('Server running at 5000');
```

Step 4

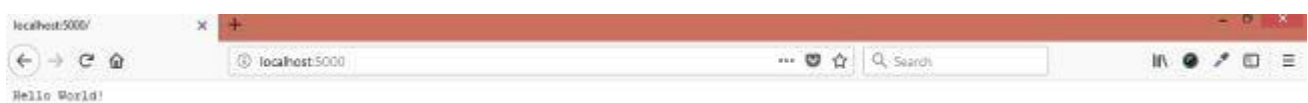
Now, again open your command prompt and type node app.js and press enter for running the node js file. See the below example screenshot.



```
C:\Windows\system32\cmd.exe - node app.js  
D:\Blog_Tutorial\FirstNodeExample>node app.js  
Server running at 5000
```

Step 5

Now, the Server is running. Here, we will create the server that is running on 5000 port for running this example. See the below screenshot for how to run this example. Just type localhost:5000 in your browser address bar and press Enter. See the below output screenshot.



6)What do you understand by callback hell?

Ans- Callback hell refers to a coding pattern where there is a lot of nesting of callback functions. The code forms a pyramid like structure and it becomes difficult to debug. Callback hell refers to a coding pattern where there is a lot of nesting of callback functions. The code forms a pyramid like structure and it becomes difficult to debug. It can be avoided by: a. Using promises b. Yield operator and Generator functions from ES6 c. Modularizing code d. Using async library (<http://caolan.github.io/async/>)

7)How does Node.js overcome the problem of blocking of I/O operations?

Ans- By putting the event-based model at its core and using an event loop instead of threads, Node.js overcomes the problem of blocking I/O operations.

8)What is an Event Emitter in Node.js?

Ans- The EventEmitter is a module that facilitates communication/interaction between objects in Node

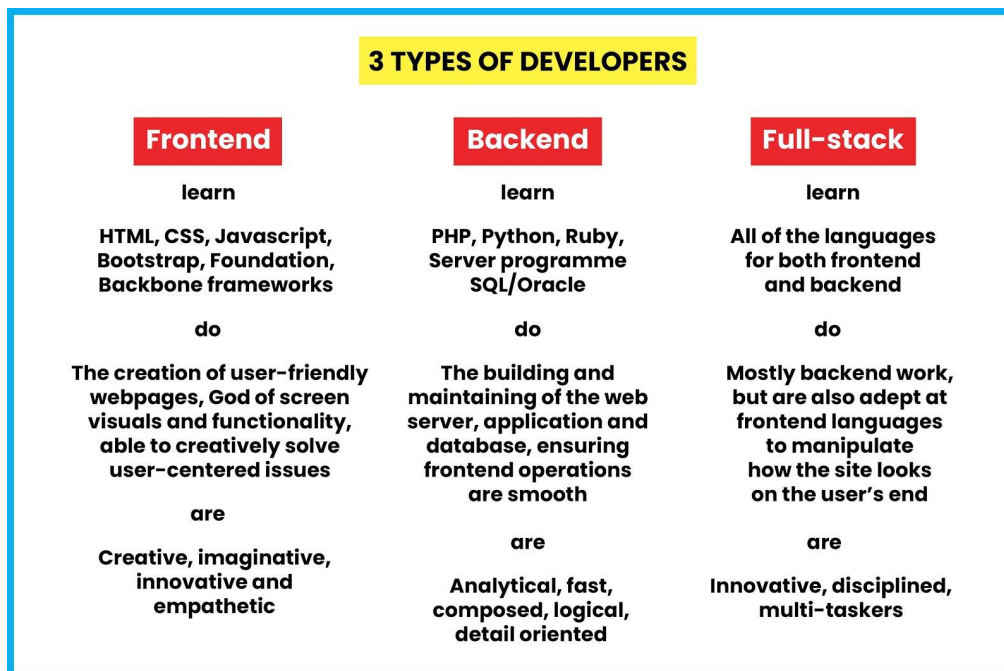
9)What are the modules in Node.js?

Ans- Module in Node.js is a simple or complex functionality organized in single or multiple JavaScript files which can be reused throughout the Node.js application.

Node.js includes three types of modules:

1. Core Modules
2. Local Modules
3. Third Party Modules

10)Explain the difference between frontend and backend development?



Ans-

11)What is the difference between Angular and Node.js?

Ans-

Angular	Node.js
1. It is an open source web application development framework	1. It is a cross-platform run-time environment for applications
2. It is written in TypeScript	2. It is written in C, C++ and JavaScript languages
3. Used for building single-page client-side web applications	3. Used for building fast and scalable server-side networking applications
4. Angular itself is a web application framework	4. Node.js has many different frameworks like Sails.js, Partial.js, and Express.js, etc.
5. Ideal for creating highly active and interactive web apps	5. Ideal for developing small size projects
6. Helpful in splitting an app into MVC components	6. Helpful in generating database queries
7. Suitable for developing real-time applications	7. Suitable in situations where something faster and more scalable is required

12)What is data binding? Which type of data binding does Angular deploy?

Ans- Data Binding is the mechanism that binds the applications UI or User Interface to the models. Using Data Binding, the user will be able to manipulate the elements present on the website using the browser.

In **one-way databinding**, the value of the Model is used in the View (HTML page) but you can't update Model from the View. Angular Interpolation / String Interpolation, Property Binding, and Event Binding are the example of one-way databinding.

In **two-way databinding**, automatic synchronization of data happens between the Model and the View. Here, change is reflected in both components. Whenever you make changes in the Model, it will be reflected in the View and when you make changes in View, it will be reflected in Model.

13)What are decorators in Angular?

Ans- Decorators are design patterns used to isolate the modification or decoration of a class without modifying the source code.

In AngularJS, decorators are functions that allow a **service, directive, or filter** to be modified before it is used.

14)What are Annotations in Angular?

Ans- Annotation creates the attribute 'annotations' that stores arrays and pass metadata to the constructor of the annotated class.

15)What are Components in Angular?

Ans- Components are the main building block for Angular applications. Each component consists of:

- An HTML template that declares what renders on the page
- A TypeScript class that defines behavior
- A CSS selector that defines how the component is used in a template
- Optionally, CSS styles applied to the template

16)what are Pipes in Angular?

Ans- Pipes are a useful feature in Angular. They are a simple way to transform values in an Angular template. There are some built in pipes, but you can also build your own pipes. A pipe takes in a value or values and then returns a value.

17)Explain the lifecycle hooks in Angular?

Ans- Lifecycle hooks are a special functionality in Angular that allow us to "hook into" and run code at a specific lifecycle event of a component or directive.

Angular manages components and directives for us when it creates them, updates them, or destroys them.

18)What are Services in Angular?

Ans- An Angular service is a stateless object and provides some very useful functions. These functions can be invoked from any component of Angular, like Controllers, Directives, etc. This helps in dividing the web application into small, different logical units which can be reused.

19)What are Template and Reactive forms?

Ans- Template Driven Forms are based only on template directives, while Reactive forms are defined programmatically at the level of the component class. Reactive Forms are a better default choice for new applications, as they are more powerful and easier to use.

20)What is a Document in MongoDB?

Ans- MongoDB is a document-oriented database. A document in MongoDB is a basic unit or basic building block of data. **MongoDB Document** is an entity in which zero or more ordered field-value pairs are stored.

21)What is a Collection in MongoDB?

Ans- A collection is a grouping of MongoDB documents. Documents within a collection can have different fields. A collection is the equivalent of a table in a relational database system. A collection exists within a single database

22)How to perform queries in MongoDB?

Ans- <https://www.digitalocean.com/community/tutorials/how-to-create-queries-in-mongodb>

23)What are two options to download jQuery Mobile library?

Ans- Download jQuery Library from CDNs. A CDN (Content Delivery Network) is a network of servers designed to serve files to the users. If you use a CDN link in your webpage, it moves the responsibility of hosting files from your own servers to a series of external ones.

24)How to set the icon and icon position in the button ?

Ans- The jQuery Mobile framework includes a selected set of icons most often needed for mobile apps. To minimize download size, jQuery Mobile includes a single white icon sprite, and automatically adds a semi-transparent black circle behind the icon to ensure that it has good contrast on any background color.

An icon can be added to a button by adding a **data-icon** attribute on the anchor specifying the icon to display. For example, the following markup:

```
<a href="index.html" data-role="button" data-icon="delete">Delete</a>
```

Icon positioning

By default, all icons in buttons are placed to the left of the button text.

Delete

This default may be overridden using the **data-iconpos** attribute to set the icon to the right, above (top) or below (bottom) the text. For example, the markup:

```
<a href="index.html" data-role="button" data-icon="delete" data-iconpos="right">Delete</a>
```

You can also create an icon-only button, by setting the **data-iconpos** attribute to **notext**. The button plugin will hide the text on-screen, but add it as a **title** attribute on the link to provide context for screen readers and devices that support tooltips. For example, replacing **data-iconpos="right"** on the previous example with **data-iconpos="notext"**:

```
<a href="index.html" data-role="button" data-icon="delete" data-iconpos="notext">Delete</a>
```

25)How to create two column grid and two column layout?

Ans- The layout is very important while designing a website. [CSS grid](#) layout is used to add columns and rows to the webpage. In this tutorial, we will be creating a 2-column layout grid in CSS.

Using CSS flexbox

We can create a 2-column layout grid using [CSS flexbox](#). Add the **display: flex** property to the container class. Use **flex** property to the column.

Example: Create a 2-column layout grid with CSS

Here is an example to create a 2-column layout using flexbox properties. Both the column has equal width.

```
<!DOCTYPE html><html lang="en"><head><meta charset="utf-8"><title>HTML</title><style>

    .row {

        display: flex;

    }

    .column {

        flex: 50%;

        padding: 16px;

        height: 250px;

    }</style></head><body>

<h2>Creating two column layout </h2>

<div class="row">

    <div class="column" style="background: #CCCCCC;">

        <h2> column 1 </h2>

        <p> some contents in column 1</p>

    </div>

    <div class="column" style="background: #d3d3d3;">
```

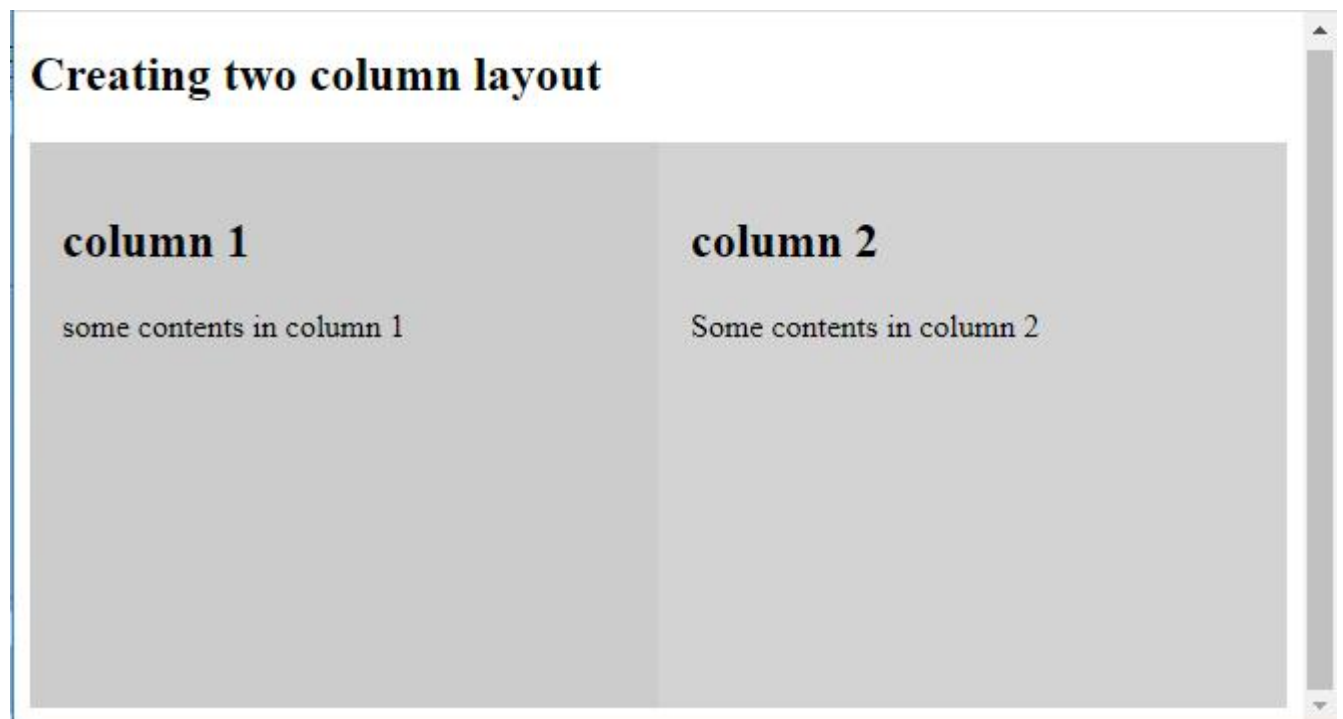
```
<h2>column 2</h2>

<p> Some contents in column 2 </p>

</div>

</div></body></html>
```

Here is the output of the above program.



Using CSS grid property

The grid property is used to layout major page areas. We can create a grid of fixed track sizes by creating a grid container using `display:grid` and add two columns using `grid-template-columns`.

26)What are mobile events?

Ans- jQuery Mobile Events

You can use any standard jQuery events in jQuery Mobile.

In addition, jQuery Mobile also offers several events that are tailor-made for mobile-browsing:

- Touch events - triggers when a user touches the screen (tap and swipe)
- Scroll events - triggers when a user scrolls up and down
- Orientation events - triggers when the device rotates vertically or horizontally
- Page events - triggers when a page is shown, hidden, created, loaded and/or unloaded

27)How to display date picker in the page?

Ans- DatePickers are needed whenever we want to ask for the date from the user. We can store the data and use it for future uses. jQuery provides an easy method to create our DatePicker. The DatePicker of jQuery is inline, so the UI is very simple yet attractive.

Syntax:

We need an input element and then we are going to call the jQuery DatePicker function.

```
<input type="text" id="dob">
```

In the script part of the code, call the datepicker() function to instantiate the datepicker widget.

```
<script>
```

```
$("#dob").datepicker();
```

```
</script>
```

28)How to apply theme for dialogs, header, footer or content?

Ans- <https://material.io/components/dialogs/android#theming-dialogs>

http://www-db.deis.unibo.it/courses/TW/DOCS/w3schools/jquerymobile/jquerymobile_themes.asp.html

29)What are scroll start and scroll stop events?

Ans- Scrollstart The scrollstart event is triggered when the user starts to scroll the page:

Example

```
$(document).on("scrollstart",function(){  
    alert("Started scrolling!");  
});
```

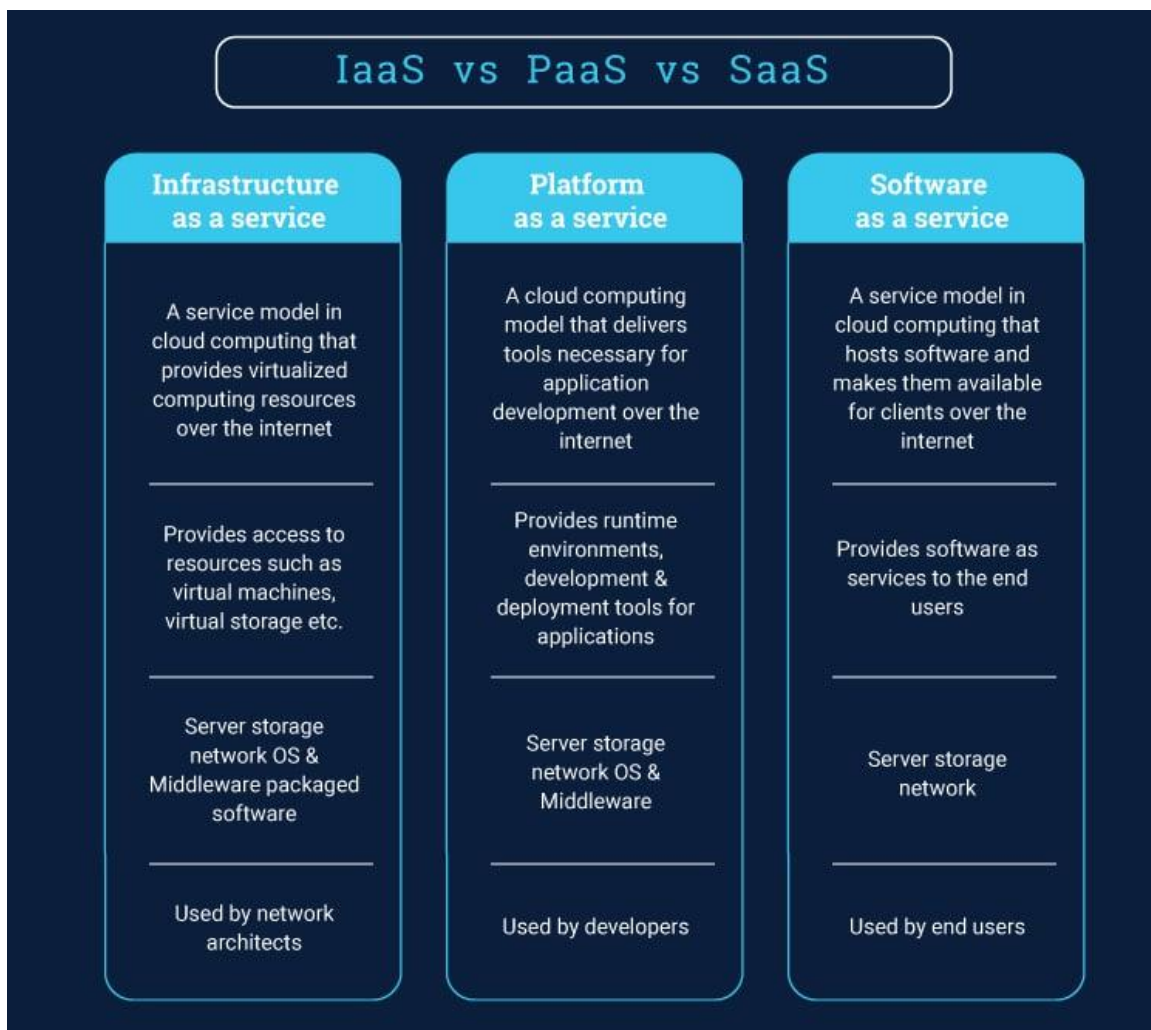
jQuery Mobile Scrollstop

The scrollstop event is triggered when the user stops to scroll the page:

Example

```
$(document).on("scrollstop",function(){  
    alert("Stopped scrolling!");  
});
```

30)List different types of Cloud Services.



31)What do you understand by stopping and terminating an EC2 Instance?

Ans-

When you stop an AWS EC2 instance, you do not get charged for that instance and can restart the instance with all the data and software on the EC2 instance being intact. However, you are charged for the EBS volumes attached with the instance.

When you terminate an AWS EC2 instance, you cannot restart the instance and all the data is permanently lost. You cannot use that same instance ever again. Once the state of the instance is changed to shutting-down or terminated you are no longer charged for that instance.

It is always a good practice to take backup of your EC2 instance as AMI and of your EBS volumes as snapshot before terminating the instance. A new EC2 instance with an EBS volume can be spawned from the AMI and the snapshot.

32)What are the different types of EC2 instances based on their costs?

Ans- Based on costs, Amazon Web Services (AWS) offers four different types of EC2 instances. These are:

- On-demand Instances;
- Reserved Instances;
- Spot Instances; and
- Dedicated Instances.

1. On-Demand instances

In simple language, you only pay what you consume with on-demand instances. With no upfront payment, you are charged based on the consumed computing capacity – by the day or hour – for the EC2 instance.

2. Reserved instances

Provides a discounted price – up to 75% of on-demand instance costs – in return for a one-time upfront payment. In the long run, reserved instances can be very cost-effective as compared to the on-demand pricing model.

3. Spot instances

This pricing model allows you to bid for spare or unused EC2 computing power for up to 90% of on-demand pricing. Spot instance pricing can depend majorly on the supply & demand for unused AWS EC2 cloud capacity.

4. Dedicated hosts

In this model, an AWS EC2 server is dedicated for your use. You are not billed on the basis of the number of instances that you have used. You can purchase a dedicated host either through an hourly on-demand price or through a reservation price that is 70% cheaper than on-demand pricing

33) Explain what S3 is?

Ans-

Amazon Simple Storage Service (Amazon S3) is a scalable, high-speed, web-based cloud storage service. The service is designed for online backup and archiving of data and applications on Amazon Web Services (AWS). Amazon S3 was designed with a minimal feature set and created to make web-scale computing easier for developers.

Amazon S3 is an object storage service, which differs from other types of cloud computing storage types, such as block and file storage. Each object is stored as a file with its metadata included. The object is also given an ID number. Applications use this ID number to access objects.

34) How does Amazon Route 53 provide high availability and low latency?

Ans-

Route 53 is built using Amazon Web Services highly available and reliable infrastructure. The distributed nature of our DNS servers helps ensure a consistent ability to route your end users to your application by circumventing any internet or network related issues. Route 53 is designed to provide the level of dependability required by important applications. Using a Network of DNS servers, Route 53 is designed to automatically answer queries from the optimal location depending on network conditions. As a result, the service offers low query latency for your end users.