**HTML:**

1)

<!DOCTYPE> must be the first thing in HTML document. It is is before the <html> tag. It is not the html tag. It is used to give instructions to web about what version of HTML the page is written in.

2)

<input type =”text”> defines one line text input field for text input.

<input type = “radio”> defines radio button

<input type =”checkbox”> defines checkbox.

It allows user to select zero or more options from limited number of choices.

3)

Table is represented with <table> tag. Following are table elements.

Table row

Table header

Table data.

4)

Iframe is used to display a webpage within a webpage.

Example:

<iframe src = <http://www.w3schools.com> width = 200px length = 250px style = “background: yellow”></iframe>

5)

**Block elements:** Block elements always start in a new line and take up the full width available.

Example: <div>

<h1>……<h6>

<form>

6)

Inline elements don’t start in a new line. These elements take required width.

7)

With local storage, web applications store data locally within users’ browser. Before HTML5 data had to be store in cookies local storage is very secure and larger amount of data can be store locally without effecting the website performance.

Local storage provided two objects for storing data on client.

Window.localstorage: Stores data with no expiration date.

Window.sessionstorage: stores data for only one session.

8)

HTML5 controls:

SVG: Scalable Vector Graphics.

SVG is used to define graphics for web.

Canvas elements is used to draw the graphics on fly via JS.

Canvas is just container for graphics but we must use JS to draw graphics.

Video element specifies a standard way to embed a video in webpage.

9)

Semantic element gives it’s exact meaning to both browser and developer.

Examples of semantic elements:

<Header> <footer> <aside> <section> <article> <nav> <fig> <figcaption> <main> <details> <mark>

10)

<Article> tag contains independent self-contained content.

<section> tag defines sections in a document such as chapters, headers,footers,or any other sections of a document.

**CSS:**

1)

In CSS, selectors are patterns used to select the elements that you want to style.

2)

A pseudo class is used to define special state of an element.

3)

CSS Z index specifies stack order of an element. The element with greater stack order is always in front with lower stack order.

**Javascript:**

3)

JS variables are containers for storing data values.

Var X = 5;

Var x stores value of 5.

2)

JS variables can hold many datatypes. Example: Number, string, null, object, Boolean, function.

1)

Creating function with no arguments:

Function myFunction(X ,Y) {  
if(Y===undefined) {  
Y = 0;

return X\*Y;  
}

}

4)

**JS properties** are values associated with JS object. JS object is collection of unordered properties.

JS properties are values associated with objects.

Person.name = Shruthi;

Here name is the property.

**5)**

**JS object:**

Var person = { firstname: ”Shruthi”,Lastname:”Mekarthi”,age:25};

6)

**JS methods** are actions that can be performed on objects.

7)

**Prototype:**

Every JS object has prototype. All JS objects inherit properties and methods from prototype.

8)

**Inheritance:** If one object acquires properties and methods of parent object is known as inheritance.

9)

**Self-invoking function**: A function that is called by itself.

10)

**XML HTTP request** object is used to request data from server. It is developers dream. We can request the data from a server, update a page without reloading the page.

11)

**Setintervel**: Setintervel is used to repeat particular task at the given time interval.

12)

**Hoisting** is JavaScript’s default behavior of moving all declarations to the top of current scope.

**JQUERY:**

**Callback function**: JS lines executed one after the other. Next line of code can be run even though previous effect is not finished. This creates errors. To avoid this we use call back functions. Callback function is executed after that current effect is 100% finished.

**Bind** method attaches one or more event handlers to element and specifies a function to run when the event occurs.

**DELEGATE** method attaches one or more event handlers to specified elements that are children of selected elements and specifies a function to run when the event occurs.

**LIVE** method is used to attach one or more event handlers to specified elements and specifies a function to run when the events occur.

**On** method is used to attach one or more event handlers to selected elements and child elements.

**Trigger method** triggers the specified event and default behavior of the event for selected elements. Trigger method is just like trigger-handler method except trigger-handler method doesn’t trigger the default behavior of event.

**Event. Target** property returns which DOM element triggers the event.

Bais anguler aplictaion