



Certified Angular JS Developer
Sample Material
VS-1251

Vskills Certifications

Vskills Reading Material



1. WEB DEVELOPMENT BASICS

Web development involves developing a web site for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing the simplest static single page of plain text to the most complex web-based internet applications (or just 'web apps') electronic businesses, and social network services.

1.1. Web Application

A web application or web app is a client-server computer program in which the client (including the user interface and client-side logic) runs in a web browser. Common web applications include webmail, online retail sales, online auctions, wikis, instant messaging services and many other functions.

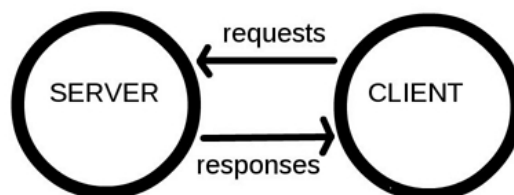
Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a mobile app. HTML5 introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.

Web applications can be considered as a specific variant of client-server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard procedures such as HTTP. Client web software updates may happen each time the web page is visited. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application.

Applications are usually broken into logical chunks called "tiers", where every tier is assigned a role. Web applications lend themselves to an n-tiered approach by nature with the most common structure is the three-tiered application. A web browser is the first tier (presentation), an engine using some dynamic Web content technology (such as ASP, CGI, ColdFusion, Dart, JSP/Java, Node.js, PHP, Python or Ruby on Rails) is the middle tier (application logic), and a database is the third tier (storage). The web browser sends requests to the middle tier, which services them by making queries and updates against the database and generates a user interface.

Web Client-Server

Computers connected to the web are called clients and servers. A simplified diagram of how they interact might look like this:



- ✓ Clients are the typical web user's internet-connected devices (for example, your computer connected to your Wi-Fi, or your phone connected to your mobile network) and web-accessing software available on those devices (usually a web browser like Firefox or Chrome).

- ✓ Servers are computers that store webpages, sites, or apps. When a client device wants to access a webpage, a copy of the webpage is downloaded from the server onto the client machine to be displayed in the user's web browser.

When you type a web address into your browser (for our analogy that's like walking to the shop)

- ✓ The browser goes to the DNS server, and finds the real address of the server that the website lives on (you find the address of the shop).
- ✓ The browser sends an HTTP request message to the server, asking it to send a copy of the website to the client (you go to the shop and order your goods). This message, and all other data sent between the client and the server, is sent across your internet connection using TCP/IP.
- ✓ Provided the server approves the client's request, the server sends the client a "200 OK" message, which means "Of course you can look at that website! Here it is", and then starts sending the website's files to the browser as a series of small chunks called data packets (the shop gives you your goods, and you bring them back to your house).
- ✓ The browser assembles the small chunks into a complete website and displays it to you (the goods arrive at your door – new shiny stuff, awesome!).

1.2. HTML

Web is a collection of documents that all link together, and bear a strong similarity to the printed documents around us. Web pages are written in HTML (Hypertext Markup Language) or XHTML (eXtensible Hypertext Markup Language). Both are document-layout and hyperlink-specification language. They define how to display the contents of the document, including text, images, and other support media. The language also tells how to make hypertext links, which connect document with other documents.

In keeping with the principle of separation of concerns, the function of HTML is primarily to add structural and semantic information to the raw text of a document. Presentation and behavior are separate functions, which can be added as desired, ideally through links to external documents such as style sheets, graphics files, and scripts.

HTML Versions

There have been several versions of HTML and is overseen by an organization called the W3C (World Wide Web Consortium). The last major version of HTML was HTML 4.01 in December 1999. In January 2000, some stricter rules were added to HTML 4.01, called as XHTML (Extensible Hypertext Markup Language). HTML 5 is the latest revision of the HTML standard and currently remains under development.

Elements, Tags and Attributes

HTML is an embedded language, the language's directions or tags are inserted into the HTML document that browser loads for viewing. The web browser uses the information inside the HTML tags to decide how to display or otherwise treat the subsequent contents of a HTML document.

Similar to a word processor like MS-Word in which styles are applied to the text; in a HTML document markups or tags are applied to stylize text as bold or italicize. Specific tags are applied on the text for specific styling.

HTML documents are composed of a tree of HTML elements. Each element can have attributes specified. Elements can also have content, including other elements and text. HTML elements represent semantics, or meaning. For example, the title element represents the title of the document. In the HTML syntax, most elements are written with a start tag and an end tag, with the content in between like

`<p>In the HTML syntax, most elements are written ...</p>`

However, not all of these elements require the end tag, or even the start tag, to be present like the `br` element.

HTML element - It represents structure or semantics and generally consists of a start tag, content, and an end tag for example, following is a bold element

` This is in bold or more dark.`

HTML tags - They are used to mark the start and end of an HTML element.

Start tag has opening angle bracket (`<`) followed by element name, zero or more space separated attribute/value pairs, and a closing angle bracket (`>`).

A start tag with no attributes: `<p>`

A start tag with an attribute `<p class="info">`

End tag has opening angle bracket followed by forward slash, the element name, and a closing angle bracket

`</p>`

This is the opening tag that tells the browser to begin treating this text as a level-1 header.

`<h1>This is an example of a level 1 header. </h1>`

This closing tag tells the browser to stop treating the text as a level-1 header and to return to the default text formatting.

Empty tag There are also some elements that are empty, meaning that they only consist of a single tag and do not have any content and look like opening tags

`
`

In XHTML, empty elements must have an end tag or the start tag must end with `'/>'` so it is used as

`
`

Difference - A tag consists of a left- and right-angle bracket and letters and numbers between those brackets, but an element is the opening and closing tags plus anything between the two tags.

HTML Attributes - It defines property for an element, consists of an name and value pair appearing within the element's start tag. Multiple name and value pairs are space separated.

All are illustrated in the figure



Few HTML tags with description, are listed

Tags	Description
<a>	ANCHOR tag creates link to other internet location, or file.
<applet>	APPLET element tags are used to embed and invoke a Java application within an HTML page.
<body>	<pre><html> <head> </head> <body> Body tags identify the content of a web page. </body> </html></pre>
 	Line Break tag is specifying a new line
<button>	BUTTON tag is used to create a <button type="button">Push Button</button>
<center>	<center>CENTER tags center text, images, etc.</center>
<code>	CODE tags are used for example, to indicate a code of the current <code>htmltags.html</code> page.
<div>	DIV tag is a logical section of a web document. <pre><div> <h1>Home Pets</h1> <p>Cats</p> <p>Dogs</p> </div></pre>

All major tags are listed

- ✓ The Root Element - html
- ✓ Metadata - head, title, base, link, meta, style
- ✓ Sections - body, article, section, nav, aside, h1, -, h6, header, footer, address
- ✓ Grouping - p, pre, blockquote, ol, ul, li, dl, dt, dd, figure, figcaption, div, main, hr

- ✓ Text - a, em, strong, cite, q, dfn, abbr, data, time, code, var, samp, kbd, mark, ruby, rb, rt, rp, rtc, bdi, bdo, span, br, wbr, small, i, b, u, s, sub, sup,
- ✓ Edits - ins, del
- ✓ Embedded Content - img, embed, object, param, video, audio, source, track, map, area, iframe
- ✓ Tables - table, tr, td, th, caption, tbody, thead, tfoot, colgroup, col
- ✓ Forms - form, input, textarea, select, option, optgroup, datalist, label, fieldset, legend, button, output, progress, meter, keygen
- ✓ Scripting - script, noscript, template, canvas

1.3. CSS

Cascading Style Sheets (CSS) is a language specifically made for illustrating the appearance of various parts of a web page. CSS gives option to control the text color, fonts style, inter-paragraph spacing, columns sizing and layout, background images or colors and many other visual effects. CSS is used to style web pages, developed in HTML or XHTML.

The major advantage of CSS is to customize the style of an entire website without changing each page individually. CSS-driven web design makes writing HTML easier. The HTML-only approach requires a lot more code to achieve the same visual effects compared with CSS version.

Including CSS

CSS can be included in a web page in following four ways

Embedded Style Sheets - Embedded style sheets are included between `<style>` and `</style>` tags in an web page's head section as

```
<style type="text/css">
body { font: 12px sans-serif; }
</ style >
```

External Style Sheets - In it CSS is in a separate document which is linked to an web page by the `<link>` element with "rel" attribute having "stylesheet" as its value, "type" attribute having "text/css" as its value and "href" attribute having the path to CSS file. It is used as

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

- ✓ rel="stylesheet" tells the link type—in this case, a link to a style sheet.
- ✓ type="text/css" lets web browser know the type of data to get—a text file, having CSS.
- ✓ href points to the location of the external CSS file and has the value of a URL similar to the src attribute in an `<image>` tag.

Import Rule - It is similar to external style sheet but instead of `<link>` element, it uses "@import" with "url" attribute storing the path to the CSS file. It is used as

```
<style type="text/css">
@import url(style.css);
</ style >
```

It can attach external style sheets to an external style sheet but the <link> tag can't.

Direct Style - Inline styles with the style attribute are used to apply CSS rules directly to an element in an web as

```
<body style="font: 12px sans-serif;">
```

CSS Components

A CSS style is basically a rule which specifies to a web browser about how to format a HTML element on a web page like a rule to make a headline blue, draw a border around a photo, or create a 250-pixel-wide sidebar box to hold a list of links. A style consist of two elements

- ✓ The web page element that the browser formats also called as the selector.
- ✓ The actual formatting instructions also called as the declaration block or rules block.

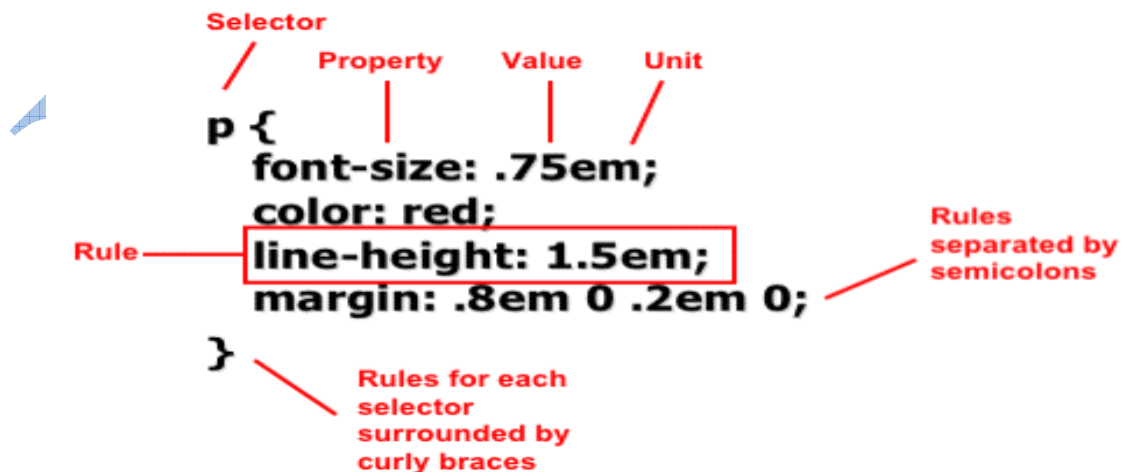
CSS Rules

A selector can be a <h1> or a <h2> tag, a <p> tag having a paragraph of text and declaration block can turn that text blue, add a red border around a paragraph, position the photo in the center of the page as required by the user.

The declaration block also called as rules block consists of property and value pairs separated by ';' and curly braces to surround the block and colons to separate the property and value. The following rule shows the parts of a style sheet and the special characters that separate them.

```
h2 {  
font-size: 16px;  
margin-top: 0;  
}
```

Also illustrated in the figure



Rules can be given in any sequence and spacing or line breaks or white space are given for better organization and readability as given in HTML. Thus giving increased maintainability and productivity.

CSS Selector Basics

Selectors are used to select the HTML elements on the web page on which declarations or rules given in the declaration block or rule block will be applied. Selectors can be divided into the following categories

- ✓ Simple selectors: Match one or more elements based on element type, class, or id.
- ✓ Attribute selectors: Match one or more elements based on their attributes/attribute values.
- ✓ Pseudo-classes: Match one or more elements that exist in a certain state, such as an element that is being hovered over by the mouse pointer, or a checkbox that is currently disabled or checked, or an element that is the first child of its parent in the DOM tree.
- ✓ Pseudo-elements: Match one or more parts of content that are in a certain position in relation to an element, for example the first word of each paragraph, or generated content appearing just before an element.
- ✓ Combinators: These are not exactly selectors themselves, but ways of combining two or more selectors in useful ways for very specific selections. So for example, you could select only paragraphs that are direct descendants of divs, or paragraphs that come directly after headings.
- ✓ Multiple selectors: Again, these are not separate selectors; the idea is that you can put multiple selectors on the same CSS rule, separated by commas, to apply a single set of declarations to all the elements selected by those selectors.

1.4. JavaScript

JavaScript is a programming language that allows you to implement complex things on web pages – every time a web page does more than just sit there and display static information for you to look at – displaying timely content updates, or interactive maps, or animated 2D/3D graphics, or scrolling video jukeboxes, etc. – you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS).

Why use JavaScript

- ✓ all browsers process JavaScript
- ✓ many web services rely on JavaScript in browser
- ✓ can use it in your own web pages
- ✓ can understand what other web pages are doing (and steal from them)
- ✓ easy to start with
- ✓ easy to do useful things with it
- ✓ programming ideas carry over into other languages

Using Variables, Objects and Arrays

```
var str = "Hello";           // local variable, when inside a function
str2 = "Hello World";       // global variable in default context (window.str2)
str3 = 'My quote char: " '; // single or double quote
str4 = "My really really really \
really long string broken into \
```


multiple lines";

```
str = 19;           // change to int
str = 0xfe + 2.343 + 2.5e3; // hex, floats, exp
```

```
var newObj = new Object(); // constructor
newObj = {}; // shorthand for same
newObj.name = "bob" // dynamic attributes
newObj.name = null // it's there (null item)
delete newObj.name // it's gone (undefined)
newObj["real age"] = 33; // array notation/hash table
```

```
var obj = { // create object using JSON
  name: "Bob", // aka JavaScript Object Notation
  details: {
    age: 33,
    "favorite color": "green"
  }
}
obj.name
obj.details["favorite color"]
```

```
var newArray = []; // no size
newArray[3] = "hi"; // grows dynamically
newArray[2] = 13; // any type
newArray.push(newObj); // add new item
newArray.pop(); // remove it
```

Comparisons and Manipulations

JavaScript has some funky types and comparisons.

```
/* javascript types */
typeof("string") == "string"
typeof(3) == typeof(3.4) == typeof(0x34) == "number"
typeof(myObject) == typeof(myArray) == "object" // arrays are objects
typeof(true) == typeof(1 == 2) == "boolean"
typeof(Math.sin) == "function"
typeof(notthere) == "undefined"
```

```
/* comparisons */
123 == "123" // true => converts type
123 === "123" // false => checks type
typeof(x) == "undefined" // x isn't there
x == null // x is defined, but null
```

```
/* Numbers */
parseInt("123") // base 10 => 123
```

```
parseInt("123", 16);    // base 16 => 291
parseFloat("123.43");   // 123.43
```

```
isNaN(0/0) == true     // illegal number
3/0 == Infinity       // legal...
-3/0 == -Infinity     //
isFinite(3/0) == false // ... but not finite
```

```
/* regular expression (regex) string comparisons */
matches = "hello".match(/h./) // returns array ["hel"] or null
```

```
re = new RegExp("h.", "ig"); // construct regexp -- no slashes
matches = "hello".match(re); // use it
```

```
"hello".replace(/h/, "b") // => "bello"
```

Conditionals and Loops

```
if (str == "Hello"){ // if-else
  alert("Hi");       // popup dialog
}
else{
  alert("something is wrong!");
}
```

```
a = 3, b = 4; // multi-assignment
c = a > b ? a : b; // c gets bigger item (b)
```

```
switch (name){ // switch statement
  case "Bob":
    alert("Hi Bob!");
    break
  case "Joe":
    alert("Hey Joe.");
    break
  default: alert("Do I know you?")
}
```

```
while (i < n){ // the basics
  // do something
  i++;
}
```

```
for (var i=0; i<n; i++){
  // do something else
}
```

```
for (var key in obj){
```

```
// do something with obj[key]
}
```

Defining Functions

```
function foo(a,b){ // global function
  return a + b;
}
```

```
var fn = function(a,b){ // save function as variable...
  return foo(a,b);
}
```

```
obj.fn = function(a,b){ // ...or as part of object
  return a + b;
}
```

```
function bar(a,b){
  var n = 1; // local var

  function helper(x) { // inner function...
    return 1/Math.sqrt(x + n); // .. can use local vars
  }
  return helper(input); // avoid need for global function
}
```

```
foo(1,2) == fn(1,2) == 3; // true
```

Browser Document Object Model (DOM), and GUI

Find and change HTML elements.

```
alert("message"); // messagebox with "OK"
var choice = confirm("message"); // OK/CANCEL true or false
var input = prompt("message", "default value"); // enter a value; null if cancelled
```

```
x = document.getElementById("foo"); // finds <div id="foo"></div>
```

```
x.style.background = "#333"; // set CSS style
x.style.borderLeft = "1px solid #ccc"; // border-left => borderLeft (camelCase)
```

```
x.className = "myclass"; // set CSS class
x.innerHTML = "Hello"; // set html inside div
```

```
y = document.getElementById("myinput"); // input area/textarea
y.value = "Hi"; // get or set text
```

Certifications

- ▶ **Accounting, Banking & Finance**
 - Certified GST Professional
 - Certified AML-KYC Compliance Officer
 - Certified Business Accountant
 - Certified BASEL III Professional
 - Certified GAAP Accounting Standards Professional
 - Certified Treasury Markets Professional
- ▶ **Big Data**
 - Certified Hadoop and Mapreduce Professional
- ▶ **Cloud Computing**
 - Certified Cloud Computing Professional
- ▶ **Design**
 - Certified Interior Designer
- ▶ **Digital Media**
 - Certified Social Media Marketing Professional
 - Certified Inbound Marketing Professional
 - Certified Digital Marketing Professional
- ▶ **Foreign Trade**
 - Certified Export Import (Foreign Trade) Professional
- ▶ **Health, Nutrition and Well Being**
 - Certified Fitness Instructor
- ▶ **Hospitality**
 - Certified Restaurant Team Member (Hospitality)
- ▶ **Human Resources**
 - Certified HR Compensation Manager
 - Certified HR Staffing Manager
 - Certified Human Resources Manager
 - Certified Performance Appraisal Manager
- ▶ **Office Skills**
 - Certified Data Entry Operator
 - Certified Office Administrator
- ▶ **Project Management**
 - Certified Master in Project Management
 - Certified Scrum Specialist
- ▶ **Real Estate**
 - Certified Real Estate Consultant
- ▶ **Marketing**
 - Certified Marketing Manager
- ▶ **Quality**
 - Certified Six Sigma Green Belt Professional
 - Certified Six Sigma Black Belt Professional
 - Certified TQM Professional
- ▶ **Logistics & Supply Chain Management**
 - Certified International Logistics Professional
 - Certified Logistics & SCM Professional
 - Certified Supply Chain Management Professional
- ▶ **Legal**
 - Certified IPR & Legal Manager
 - Certified Labour Law Analyst
 - Certified Business Law Analyst
 - Certified Corporate Law Analyst
- ▶ **Information Technology**
 - Certified Angular JS Professional
 - Certified Basic Network Support Professional
 - Certified Business Intelligence Professional
 - Certified Core Java Developer
 - Certified E-commerce Professional
 - Certified IT Support Professional
 - Certified PHP Professional
 - Certified Selenium Professional
- ▶ **Mobile Application Development**
 - Certified Android Apps Developer
 - Certified iPhone Apps Developer
- ▶ **Security**
 - Certified Ethical Hacking and Security Professional
 - Certified Network Security Professional
- ▶ **Management**
 - Certified Corporate Governance Professional
 - Certified Corporate Social Responsibility Professional
 - Certified Leadership Skills Professional
- ▶ **Life Skills**
 - Certified Business Communication Specialist
 - Certified Public Relations Officer
- ▶ **Media**
 - Certified Advertising Manager
 - Certified Advertising Sales Professional
- ▶ **Sales, BPO**
 - Certified Sales Manager
 - Certified Telesales Executive

& many more job related certifications

Contact us at:
V-Skills
011-473 44 723 or info@vskills.in
www.vskills.in