

# Web 2.0 Basics

## A beginner's Handbook

By Sandesh Paudel

### Table of Contents:

#### 1. Javascript

- a. The Document-Object Model
- b. JavaScript and DOM
- c. Properties
- d. Methods

#### 2. Mobile Design

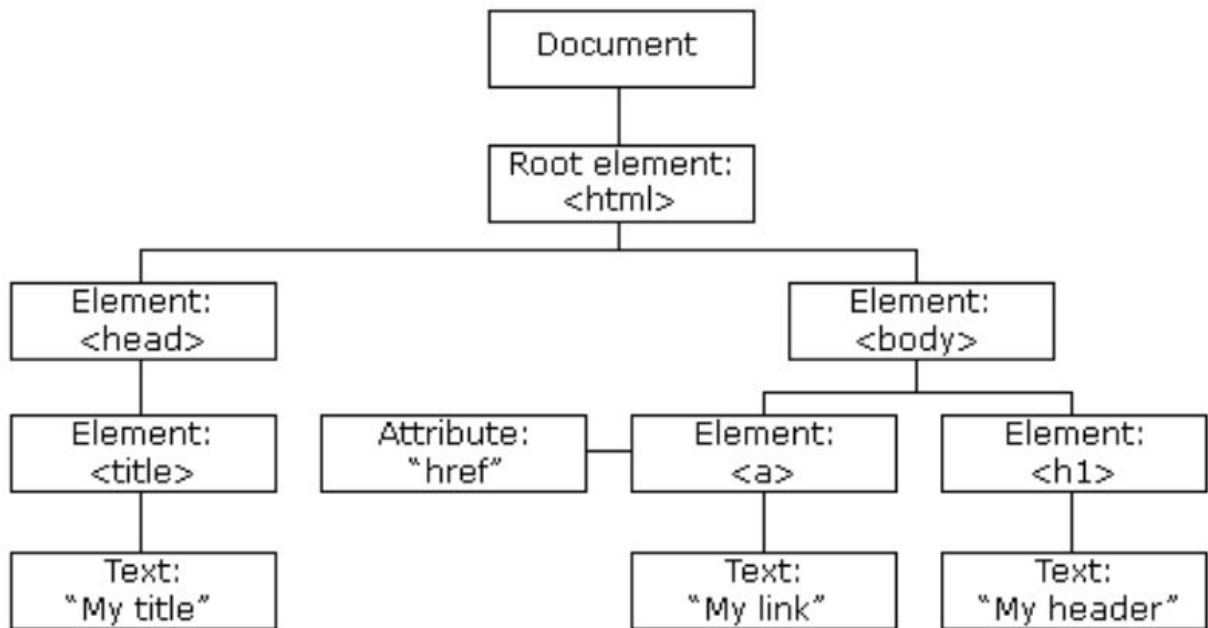
- a. Goals
- b. The ViewPort metatag
- c. Flexible images
- d. Media Queries
- e. Layout, Navigation & Text

#### 3. SEO

- a. Periodic table of SEO success
- b. On-page factors
- c. Off-page factors

## ❖ JavaScript

### THE DOM (DOCUMENT OBJECT MODEL)



> **Programming Interface:** DOM is a programming interface for HTML and XML. It represents the page in a tree-like structure so programs can change the document structure, style and content. It is an **object-oriented representation of a webpage**, which can be modified by a scripting language like JavaScript.

> **Object:** All the things in a document can be an Object and all of their "handles". This includes **Elements**, or any other user-defined objects.

> DOM objects have **properties** and **methods** associated with them.

> **Property:** a property is a **characteristic** or an **attribute** of an object/ element:

- The URL of a web page document  
`document.URL`
- The title of a web page document  
`document.title`
- The date the web page file was last modified  
`document.lastModified`
- The src file of an image object  
`mymage.src`

> Some common DOM object **properties** (x is a node object/HTML element):

- x.innerHTML - the inner text value of x (a HTML element)
- x.nodeName - the name of x
- x.nodeValue - the value of x
- x.parentNode - the parent node of x
- x.childNodes - the child nodes of x
- x.attributes - the attributes nodes of x

> **Methods:**

- A method is an action the object can do or respond to:
- Writing text to a web page document:

```
document.write()
```

.addEventListener() method

- Waits for a specific "event" on a particular element (.getElementById()), then runs a *function()*

> Some common DOM object **methods** (x is a node object/HTML element):

- x.getElementById(*id*) - get the element with a specified id
- x.getElementsByTagName(*name*) - get all elements with a specified tag name
- x.appendChild(*node*) - insert a child node to x
- x.removeChild(*node*) - remove a child node from x

> Accessing DOM object via Javascript:

```
/* Get a text from <p id="intro"> text </p> */
var content = document.getElementById("intro").
  childNodes[0].nodeValue;
```

> Example: Button toggle element display html

```
<button id="dynamicbutton">More </button>
```

```
<div id="hiddencontent">
```

**JS:**

```
// Step 1: Get a handle on the clickable element (the button) and create a variable
var toggle_button = document.getElementById("dynamicbutton");

// Step 2: Get a handle on the element to be toggled (the image) and create a variable
var toggle_element = document.getElementById("hiddencontent");

// Step 3: Set the default state of the element to be toggled; either "inline"
toggle_element.style.display = 'none';

// Step 4: write the function to do the toggling (won't execute 'til called)
function toggleElement(e) {

    // Step 4a: IF display:inline, it must be showing, so hide it
    if (e.style.display == 'inline') {
        e.style.display = 'none'
    }
    // Step 4b: ELSE it must be hidden, so show it
    else {
        e.style.display = 'inline'
    }
}

};

// Step 5: Apply a "click listener" to the clickable element,
toggle_button.addEventListener("click", function(){
    //toggleElement(toggle_element)
    toggleElement(toggle_element)
});
```

**> JS Libraries:**

- **Advanced JavaScript is difficult and time-consuming. Javascript helper libraries/ frameworks that are written by others can be used to make JS more convenient to use.**

- **To Use:**

1. Copy the library file to your file system
2. Link to the file from your HTML file
3. Learn how to use the library
4. Write the JavaScript code in your HTML file to "turn-on" and use the library

Advantages	Disadvantages
Speeds up the development process Helpful in collaborative work	Yet another learning curve
Clean and tidy code	Unused code leftover
Solutions to common problems <ul style="list-style-type: none"> <li>- Browser compatibility solutions</li> <li>- Typical patterns, ready to use/edit</li> </ul>	You inherit bugs & bad practices
Learn good practices	You don't learn to do it yourself

# jQuery

- The most popular JavaScript library on the Internet today
- It's JavaScript!
- Reuses what you know: CSS selectors to access and manipulate the DOM
- Unobtrusive by Design
  - Respects progressive enhancement - adds functionality without having to hack-up HTML documents
  - Follows the "Principle of least astonishment"
  - Built to be easy for Web Developers
 

```
$(".someSelector").someAction()
```
- Some guy from RIT: John Resig

The main jQuery function is the `$()` function ...the jQuery function  
jQuery allows you to select elements by CSS selectors.

## Plain JavaScript – stick some text in an HTML element

Step 1: Target the ID in the HTML (it must be an ID), get a "handle" on the element in the DOM, and assign the handle to a new variable

```
var e = document.getElementById("someID");
```

Step 2: Assign some HTML content to the "inside node" of the variable

```
e.innerHTML = "JavaScript says Hello World!";
```

## jQuery – stick some text in an HTML element

Step 1: install jQuery

Step 2: Send some HTML content to the `.html()` method that is associated with the targeted CSS selector

```
$(".any-element").html("jQuery says Hello World!");
```



# CDN - Content Delivery Networks

- You want your web pages to be as fast as possible
- You want the browser to **cache** as much as possible
- If many different web sites use the same JavaScript framework, it makes sense to host the framework library in a common location for every web page to share
- A CDN is a network of servers containing shared code libraries
- CDNs provide a number of popular JavaScript libraries, including:
  - jQuery
  - Prototype
  - MooTools
  - Dojo
  - Yahoo! YUI

Google: <http://code.google.com/apis/libraries/devguide.html>

Microsoft: <http://www.asp.net/ajax/cdn>

CDN JS: <https://cdnjs.com/>

# Where to get jQuery

- Option 1: download and install from [www.jquery.com](http://www.jquery.com)

- In your HTML...

```
<script src="js/jquery.js"></script>
```

- Option 2: use a Content Delivery Network (CDN)

- In your HTML...

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
```

- Or, for development (less typing)...

```
<script src="http://code.jquery.com/jquery.js"></script>
```

## Minification

- For faster Websites

- Faster downloads
- Fewer HTTP requests (when minifying multiple libraries together)

- Remove unnecessary characters (comments, line breaks, extra spaces)

- See:

<http://code.jquery.com/jquery/>

```
/*
 * jQuery JavaScript Library v2.1.3
 * http://jquery.com/
 *
 * Includes Sizzle.js
 * http://sizzlejs.com/
 *
 * Copyright 2005, 2014 jQuery Foundation, Inc. and other contributors
 * Released under the MIT license
 * http://jquery.org/license
 *
 * Date: 2014-12-18T15:11Z
 */
(function( global, factory ) {
    if ( typeof module === "object" && typeof module.exports === "object" ) {
        // For CommonJS and CommonJS-like environments where a proper "window"
        // is present, execute the factory and get jQuery.
        // For environments that do not have a "window" with a "document"
        // (such as Node.js), expose a factory as module.exports.
        // This accentuates the need for the creation of a real "window".
        // e.g. var jQuery = require("jquery")(window);
        // See ticket #14549 for more info.
        module.exports = global.document ?
            factory( global, true ) :
            function( w ) {
                if ( !w.document ) {
                    throw new Error( "jQuery requires a window with a document" );
                }
                return factory( w );
            };
    } else {
        factory( global );
    }
}( window, function( window, noGlobal ) {
    // ...
} )
```

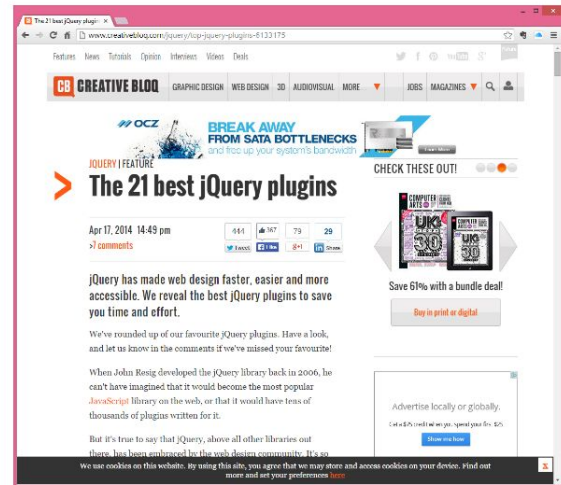
**Unminified (normal)**

```
/*! jQuery v2.1.3 | (c) 2005, 2014 jQuery Foundation, Inc. | jquery.org/license */
(function( window, noGlobal ) {
    // ...
} )
```

**Minified**

# Finding & Evaluating Plugins

- “Plugin” = an industry term
- The jQuery ecosystem
- Google it! It's been done
- Quality control
- jQuery plugin Programmers
  - Private use, or released into the wild
  - “Buyer” beware



- Author of the plugin decides
- 3 - 5 parts (depending on the plugin)
- “Open source”
- 5 Parts
  1. Library script
  2. Plugin script
  - (3) CSS file (from plugin website)
  - (4) Initialization and options script (instructions from plugin website)
  - (5) Content, arranged in a specific way (instructions from plugin website)



## 1) Dealing with Mobile Devices

### Goals

- **No horizontal scroll bars**
- **All elements fit their containers**
- Text is not too small to read comfortably, and there must be a lot of contrast between text color and its background
- Images must still be recognizable for what they are, i.e. not too small
- The navigation elements look like buttons and they are finger-sized (easily tap-able)

### Techniques:

1. Add viewport meta tag
2. Scale images as necessary
3. Use media queries to UNDO desktop layouts
  - Return side-by-side content to the natural document flow
  - Remove floats
4. Reconfigure wide elements like HTML Tables

## 2) The Viewport Meta Tag in <head>

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

-configures width and initial scale of browser viewport

## 3) Flexible images:



-**HTML**: remove height and width attributes. Wrap the image in a container that has a controlled width.

-**CSS**: target the image that needs to scale.

```
.scalable { width: 100%; }
```

## 4) Media Queries

Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

\*If the viewport is under 480 pixels wide or wider, the background will be light green.

```
@media screen and (min-width: 480px) {
  body {
    background-color: lightgreen;
  }
}
```

\*If the viewport is 480 pixels or wider, the menu will float to the left of the page.

```
@media screen and (min-width: 480px) {
  #leftsidebar {width: 200px; float: left;}
  #main {margin-left: 216px;}
}
```

> For mobile/tablet:

### • In CSS

```
@media (max-width: 768px) {
  //CSS here
}
@media (max-width: 480px) {
  //CSS here
}
```

> Reconfigure HTML tables, floats and anything else that doesn't fit

### • HTML Tables

- Change the *font-size*
- Move the table to a different area of the webpage
- Use *overflow-x:auto* on a container element
- Reconfigure the table into some other element (like an unordered list) using JavaScript
- Google: **html tables mobile**

### • Floats

- Remove floats – use *text-align: center* instead

# Layout

- Single column design
- Limit scrolling to one direction
- Use clear, short heading elements
- Use simple lists with short phrases instead of heavy textual content
- Avoid using HTML tables
- Avoid using pixel units in style sheets
- Avoid absolute positioning
- Hide content that is not essential for mobile use.



# Navigation

- Provide minimal navigation near the top or bottom of the page
- Toggle lesser navigation until needed
  - Use micro-interactions to show sub-menus as needed (sliding in and out when tapped)
  - hamburger menus?
- Avoid hyperlinks that open files in new windows or pop-up windows

# Images

- Avoid using floats
- Avoid displaying images that are wider than the screen width
- Configure alternate, small optimized background images or no background images at all
- Avoid using images that contain text
- Avoid the use of large graphic images (large file sizes)

# Text

*Remember: images can scale; text stays the same size*

- Configure good contrast between text and background colors
- Use common font typefaces
- Increase text sizes (sometimes)
- Use a short, descriptive headings





## SEO BEST PRACTICES

Search engine optimization (SEO) seems like alchemy to the uninitiated. But there's a science to it. Below are some important "ranking factors" and best practices that can lead to success with both search engines and searchers.

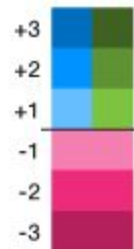
### On-The-Page SEO

### Off-The-Page SEO

Content	Architecture	HTML	Trust	Links	Personal	Social
<b>Cq</b> <sup>+3</sup> Quality	<b>Ac</b> <sup>+3</sup> Crawl	<b>Ht</b> <sup>+3</sup> Titles	<b>Ta</b> <sup>+3</sup> Authority	<b>Lq</b> <sup>+3</sup> Quality	<b>Pc</b> <sup>+3</sup> Country	<b>Sr</b> <sup>+2</sup> Reputation
<b>Cr</b> <sup>+3</sup> Research	<b>Am</b> <sup>+3</sup> Mobile	<b>Hd</b> <sup>+2</sup> Description	<b>Te</b> <sup>+2</sup> Engage	<b>Lt</b> <sup>+2</sup> Text	<b>Pl</b> <sup>+3</sup> Locality	<b>Ss</b> <sup>+1</sup> Shares
<b>Cw</b> <sup>+2</sup> Words	<b>Ad</b> <sup>+2</sup> Duplicate	<b>Hs</b> <sup>+2</sup> Structure	<b>Th</b> <sup>+1</sup> History	<b>Ln</b> <sup>+1</sup> Numbers	<b>Ph</b> <sup>+2</sup> History	
<b>Cf</b> <sup>+2</sup> Fresh	<b>As</b> <sup>+2</sup> Speed	<b>Hh</b> <sup>+1</sup> Headers	<b>Vd</b> <sup>-1</sup> Piracy	<b>Vp</b> <sup>-3</sup> Paid		
<b>Cv</b> <sup>+2</sup> Vertical	<b>Au</b> <sup>+1</sup> URLs	<b>Vs</b> <sup>-2</sup> Stuffing	<b>Va</b> <sup>-1</sup> Ads	<b>VI</b> <sup>-3</sup> Spam		
<b>Ca</b> <sup>+2</sup> Answers	<b>Ah</b> <sup>+1</sup> HTTPS	<b>Vh</b> <sup>-1</sup> Hidden				
<b>Vt</b> <sup>-2</sup> Thin	<b>Vc</b> <sup>-3</sup> Cloaking					

### Factors Work Together

All factors on the table are important, but those marked 3 carry more weight than 1 or 2. No single factor guarantees top rankings or success, but having several favorable ones increases the odds. Negative "violation" factors shown in red harm your chances.



## On-The-Page Factors

These elements are in the direct control of the publisher

### Content

<b>Cq</b>	QUALITY	Are pages well written & have substantial quality content?
<b>Cr</b>	RESEARCH	Have you researched the keywords people may use to find your content?
<b>Cw</b>	WORDS	Do pages use words & phrases you hope they'll be found for?
<b>Cf</b>	FRESH	Are pages fresh & about "hot" topics?
<b>Cv</b>	VERTICAL	Do you have image, local, news, video or other vertical content?
<b>Ca</b>	ANSWERS	Is your content turned into direct answers within search results?
<b>Vt</b>	THIN	Is content "thin" or "shallow" & lacking substance?

### Architecture

<b>Ac</b>	CRAWL	Can search engines easily "crawl" pages on site?
<b>Am</b>	MOBILE	Does your site work well for mobile devices?
<b>Ad</b>	DUPLICATE	Does site manage duplicate content issues well?
<b>As</b>	SPEED	Does site load quickly?
<b>Au</b>	URLS	Do URLs contain meaningful keywords to page topics?
<b>Ah</b>	HTTPS	Does site use HTTPS to provide secure connection for visitors?
<b>Vc</b>	CLOAKING	Do you show search engines different pages than humans?

### HTML

<b>Ht</b>	TITLES	Do HTML title tags contain keywords relevant to page topics?
<b>Hd</b>	DESCRIPTION	Do meta description tags describe what pages are about?
<b>Hs</b>	STRUCTURE	Do pages use structured data to enhance listings?
<b>Hh</b>	HEADERS	Do headlines & subheads use header tags with relevant keywords?
<b>Vs</b>	STUFFING	Do you excessively use words you want pages to be found for?
<b>Vh</b>	HIDDEN	Do colors or design "hide" words you want pages to be found for?

## Off-The-Page Factors

Elements influenced by readers, visitors & other publishers

### Trust

<b>Ta</b>	AUTHORITY	Do links, shares & other factors make pages trusted authorities?
<b>Te</b>	ENGAGE	Do visitors spend time reading or "bounce" away quickly?
<b>Th</b>	HISTORY	Has site or its domain been around a long time, operating in same way?
<b>Vd</b>	PIRACY	Has site been flagged for hosting pirated content?
<b>Va</b>	ADS	Is content ad-heavy? Do you make use of intrusive interstitials?

### Links

<b>Lq</b>	QUALITY	Are links from trusted, quality or respected web sites?
<b>Lt</b>	TEXT	Do links pointing at pages use words you hope they'll be found for?
<b>Ln</b>	NUMBER	Do many links point at your web pages?
<b>Vp</b>	PAID	Have you purchased links in hopes of better rankings?
<b>VI</b>	SPAM	Have you created links by spamming blogs, forums or other places?

### Personal

<b>Pc</b>	COUNTRY	What country is someone located in?
<b>Pl</b>	LOCALITY	What city or local area is someone located in?
<b>Ph</b>	HISTORY	Has someone regularly visited your site?

### Social

<b>Sr</b>	REPUTATION	Do those respected on social networks share your content?
<b>Ss</b>	SHARES	Do many share your content on social networks?