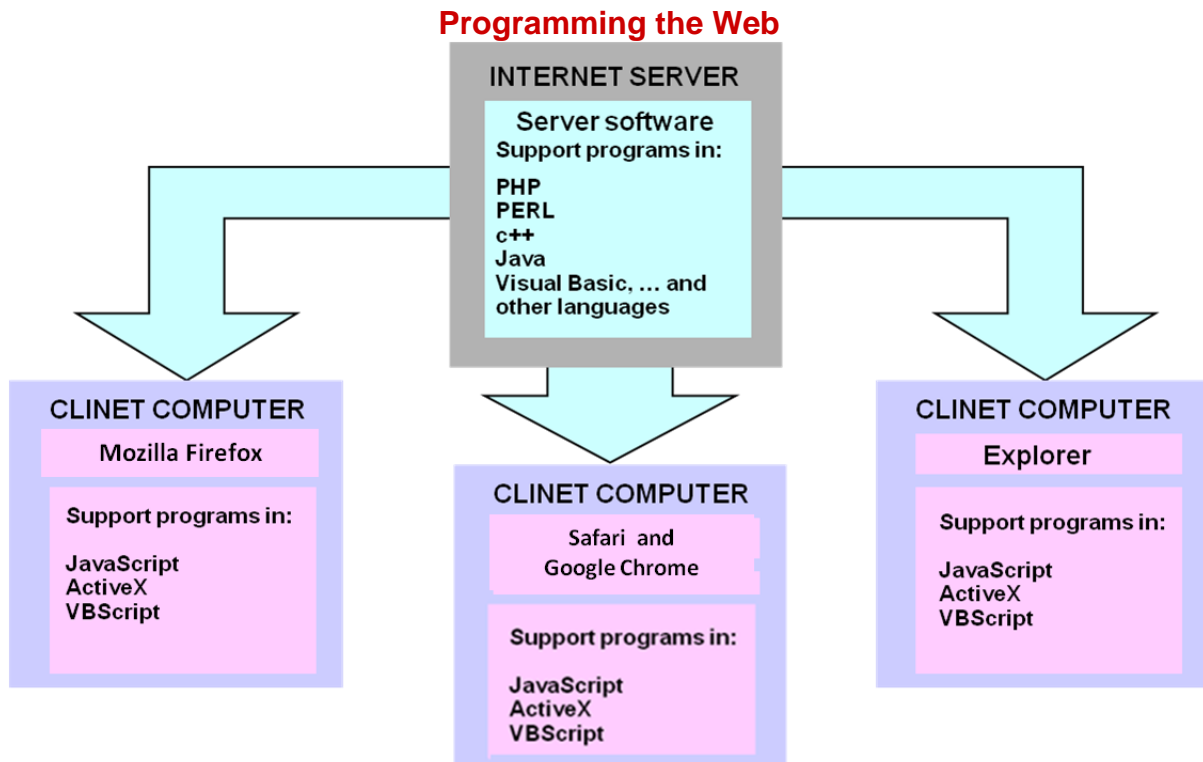


## Content:

 [Web Design Overview](#),  [HTML](#)  
[Tutorial](#),  [CSS Tutorial](#),  [PHP](#)  
[Tutorial](#),  [Bootstrap Tutorial](#),  [JavaScript](#)  
[Tutorial](#),  [Early Browsers](#),  [HCI](#)  
[Issues](#),  [Responsive Web Design](#)  
[\(RWD\)](#),  [Mobile-First Responsive Web](#)  
[Design](#),  [The Best Way Forward](#),  [Conclusions](#)  
[and Tips](#), and  [Useful Links](#).



## Web Design Overview



**Programs can be run either on the client machine by the (browser) or on the Web server.**

## **There are three types of Web Documents:**

1. Static document written once by the Web author and distributed to anyone that can access the URL for the document.

2. Dynamic documents created on the fly, based on information request by the user. For example sport scores, stock prices, . . . etc. This information may be obtained by a variety of services.

3. Active documents contains components that run on a user's browser. The components are downloaded to a user's browsers and remain available to the browser for future use. JavaScript and Active X are part of this technology.

**There are five languages / frameworks, web developers must acquire the knowledge, knowhow and the ability of using theme effectively in designing Web pages:**

↑ [Back to top](#)



### **Hypertext Markup Language**

1. **HTML** Hypertext Markup Language (HTML) is a standardised system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web (WWW) pages. It is widely used to define and position the content of static web pages.

***Do I need to learn HTML to be able to use It?:*** Most likely, no, there are many HTML editors which contains the required tools to allow you to design the Web Page. HTML editors such as [FrontPage](#), [Microsoft Expression](#), and [Macromedia Dreamweaver](#) consists of a split view option to allow the user to code in *Code* view and preview in *Design* view without the hassle of switching from the design and code view tabs for each review. By using such editors you will very quickly learn and familiarise your self with HTML tags and their attributes.

(1.1 [W3 School](#)) provides a comprehensive list of HTML tags and their attributes. (1.2. [W3 School](#)) provides examples of the usage of each tag starting with first HTML <!--...--> tag and ending with last HTML <wbr> tag in alphabetical order.

[> Learn HTML by examples](#)

[↑ Back to top](#)







## Cascading Style Sheets

2. **CSS**: Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language, and to specify the layout of web pages, particularly div class containers, and media queries are CSS3 module allowing content rendering to adapt to conditions such as screen resolution (e.g. Smartphone screen vs. computer screen).

**What is Media Query?** : The @media rule is used to define different style rules for different media types/devices. Media queries look at the capability of the device, and can be used to check many things, such as: width and height of the browser window, width and height of the device, orientation (is the tablet/phone in landscape or portrait mode?), resolution, and much more. It is widely used to redefine the attribute and features of div class containers for the given device screen size and resolutions. Media query allow web designers to design adaptive web pages responsive to the give device features and characteristics.

### **What Browsers Support Media**

**Queries?** :  IE9+,  Safari 4+,  Google Chrome 21+,  Opera 9+,  Firefox 3.5+, and others. The numbers above specifies the first browser version that fully supports the @media rule.

(2. [W3 School](#) ) provides good tutorials with examples on CSS style sheets language, and their use with media queries.

[> Learn CSS by examples](#)

[↑ Back to top](#)



## Personal Home Page

3. **PHP** Personal Home Page (PHP) is a server-side scripting language designed for web development, but also used as a general-purpose programming language. PHP (powered by the Zend Engine). After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page. PHP code can be simply mixed with HTML. It is widely used to manage Web database, community and E-commerce website projects.

**Do I Need to Learn PHP to Be Able to Use It?:** Most likely, no, providing that you are familiar with the syntax of a [structured programming language](#) such as [C](#). In the majority of web applications, server-side scripting languages, used to interrogate, and manage an existing web database, to get or put record/s in the web database. Such databases have their own fourth generation scripting language to allow you to implement the required user interface screens. [Expert grabber .com](#) is a good example of the application of (MySQL dBase). [MySQL](#) dBase is an open source relational database management system. Information in a MySQL database is stored in the form of related tables. MySQL databases are typically used for web application development (often accessed using PHP).

**Note that:** many existing languages have borrowed directly or indirectly from [C](#) including: [C++](#), [Java](#), [JavaScript](#), [Limbo](#), [LPC](#), [C#](#), [Objective-C](#), [Perl](#), [PHP](#), [Python](#), and [Verilog](#).

(3. [W3 School](#)) provides good tutorials with examples on the use and applications of PHP.



[Learn PHP by examples](#)

↑ [Back to top](#)



Twitter Bootstrap

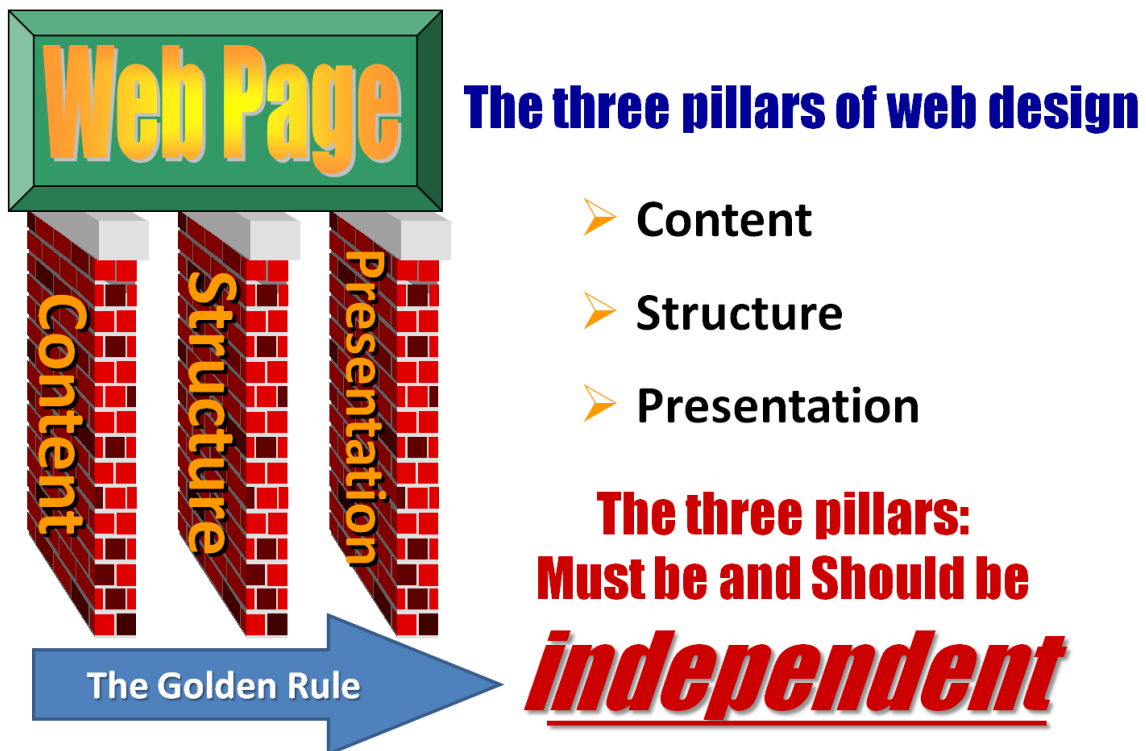
Bootstrap, from twitter

4. **Bootstrap** is a free and open-source collection of tools for creating websites and web applications. It contains HTML- and

CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. The bootstrap framework aims to ease web development. Bootstrap is a front end, that is an interface for the user, unlike the server-side code which resides on the "back end" or server. It is also a web application framework, that is a software framework which is designed to support the development of dynamic websites and web applications.

**Bootstrap vs. Foundation: Which Framework Should I Choose?** Bootstrap and Foundation are similar in many ways, and are both, belong to the family of Framework base layers of Objective-C classes. In addition both provides a set of useful primitive object classes, and they introduces several paradigms that define functionality not covered by the Objective-C language. Apparently, there are no major advantages of one framework to the other. The choice, however, is personal, depending on one experience and preference. Nick Pettit article on the subject ([Bootstrap vs. Foundation](#) ) is excellent and very informative.

## Responsive Web Design “good practice”



**When I should use Twitter Bootstrap?** if you have large project, and limited time, then, Bootstrap ability to produce rapid prototypes may be the solution for you. (Prototyping is often used as good practice in task analysis). Bootstrap is excellent front-end framework, take all the hassle of building grid system. Grid system is the core feature that enables designers and developers to rapidly prototype layouts structure for mobile first web design. However, front-end frameworks, such as Twitter

Bootstrap, are inherently exhibits dependency between the three domains: Content, Structure, and Presentation. Grid driven systems bound designer's creativity and their freedom in content driven environment, After all, web pages design is about creating content rich interactive environment to convey a given concepts, goals, and message to anticipated group/s of users. The interdependency between the three domains makes changes and modifications, time consuming, and costly challenge.

***Can I mix HTML/CSS with Twitter Bootstrap Code?*** Perhaps. However, you may get conflicts between the two styles and grid systems. Overall, you should proceed carefully and with caution!

(4. [W3 School](#)) provides good tutorials with examples on bootstrap templates and available tools.

[> Learn Bootstrap by examples](#)

[↑ Back to top](#)



**5. JavaScript** is an object-oriented computer programming language commonly used to create interactive effects within Web browsers, is often used to make dynamic and interactive web pages. It runs on your visitor's computer (client server) and doesn't require constant downloads from your website.

***Are JavaScript and Java the Same?:*** No, they are two completely different computer languages. Only their names are similar.

***What Do I Need to Run JavaScript?:*** JavaScript support is built right into all the major web browsers, including Internet Explorer, Firefox and Safari. Providing that the visitors to your site are using web browsers that support JavaScript (most do) and have JavaScript enabled (it is by default), then your JavaScript will run when they visit the page.

***Do I Need to Learn Javascript to Be Able to Use It?:*** Most likely, no, There are plenty of pre-written JavaScripts that people have made available for you to plug straight into your

web page. All you need to know to be able to use such scripts is how to paste the supplied code into the required places in your web page.

***What Do I Need to Write JavaScript?:*** JavaScript is an interpreted language, so no special program is required to create usable code. Any plain text editor such as Notepad is quite satisfactory for being able to write JavaScript. That said, an editor that colourises the code to make it easier to see what is what makes it easier to find your mistakes.

***Can I Use HTML Instead of JavaScript?:*** No, HTML and JavaScript are two completely different things. HTML is a markup language designed for defining static web page content. JavaScript is a programming language designed for performing dynamic tasks. Sometimes the distinction is confusing because JavaScript code can go in the same file as HTML

***Can I Use PHP or Some Other Server-Side Language Instead of JavaScript?:*** Perhaps. It depends where the code needs to run. If it can run before the page loads, you can use a server-side language. If it has to run after the page has loaded, then you must use JavaScript, as this is the only scripting.

***Does the Javascript Go in the Same File as the HTML?:*** It can, but your scripts will be more easily reused on multiple pages of your site if you place them in separate files. (Using a **.js** extension helps identify them as Javascript.) You then just link the JavaScript to your HTML by inserting a **<script>** tag. The same JavaScript can then be added to several pages just by adding the appropriate tag into each of the pages to set up the link.

**JavaScript Frameworks (Libraries)** There are extensive JavaScript libraries (framework) to provide the designer of ready tools to enhance the web pages dynamism and interactivities **jQuery** is the most popular JavaScript framework on the Internet today, **jQuery UI** is a collection of graphical user interface (GUI) widgets, animated visual effects, and themes implemented with **jQuery Prototype** is a JavaScript library that provides a simple API to perform common web tasks, **API** is short for Application Programming Interface. It is a library of properties and methods for manipulating the HTML DOM. **Prototype** enhances JavaScript by providing classes and inheritance, **MooTools** also a framework that offers an API to make common JavaScript programming

easier. **MooTools** also includes some lightweight effects and animation functions, **YUI** - The Yahoo! User Interface Framework is a large library that covers a lot of functions, from simple JavaScript utilities to complete internet widgets. **Ext JS** - Customizable widgets for building rich Internet applications. **Dojo**- A toolkit designed around packages for DOM manipulation, events, widgets, **Cript.aculo.us** - Open-source JavaScript framework for visual effects and interface behaviours, and **UIZE** - Widgets, AJAX, DOM, templates, and more.

**What is HTML DOM?:** The HTML DOM (Document Object Model). When a web page is loaded, the browser creates a Document Object Model of the page. The HTML DOM model is constructed as a tree of Objects. The DOM is a W3C (World Wide Web Consortium) standard. The DOM defines a standard for accessing documents: The DOM is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.

(5. [W3 School](#)) provides good tutorials with examples.



[Learn JavaScript by examples](#)

↑ [Back to top](#)



**Early Browsers**

Early browsers have very limited capabilities, "It's always expensive and complex to attempt to maintain different version of content for specific browsers" (Brian Behlendorf, Organic).

**Early Browser Capabilities**

		Netscape Navigator					Microsoft Internet Explorer			
		1.0	1.1	2.0	3.0	4.0	1.0	2.0	3.0	4.0
<b>HTML SUPPORT</b>										
<b>HTML 2.0</b>	Basic HTML and forms									
<b>HTML 3.2</b>	Style tags, tables, frames									
<b>HTML 4.0</b>	Style sheets, DHTML									
<b>TYPOGRAPHY CONTROLS</b>										
<b>&lt;FONT&gt; tag</b>	Type face, size colour									
<b>Style sheet</b>	Typographic control via style sheet									
<b>Downloadable font</b>	Automatic downloadable of nonstandard fonts									
<b>ANIMATION</b>										
<b>Animated GIF</b>	Sequenced playback of GFI graphics									
<b>DHTML</b>	Animation of HTML elements via Java Script									
<b>Java</b>	Animated interfaces programmed in Java									
<b>PROGRAMMING/INTERACTIVITY</b>										
<b>CGI</b>										
<b>JavaScript/JScript</b>										
<b>Java</b>										
<b>ActiveX</b>										
<b>ALTERNATIVE FORMAT</b>										
<b>Plug-in-support</b>										
<b>XML</b>										

It is always recommended to include the following Javascript functions to overcome the limitation of early browsers:

```
<!--[if lt IE 7]>
<script src = "http:// ie7-js.googlecode.com/svn/version /2.1 (beta4)/IE7.js"> </script>
<![endif-->
```

```
<!--[if lt IE 8]>
<script src = "http:// ie7-js.googlecode.com/svn/version / 2.1 \(beta4\)/IE8.js">
</script>
<![endif-->
```

```
<!--[if lt IE 9]>
<script src = "http:// ie7-js.googlecode.com/svn/version / 2.1 (beta4)/IE9.js">
</script>
<![endif-->
```

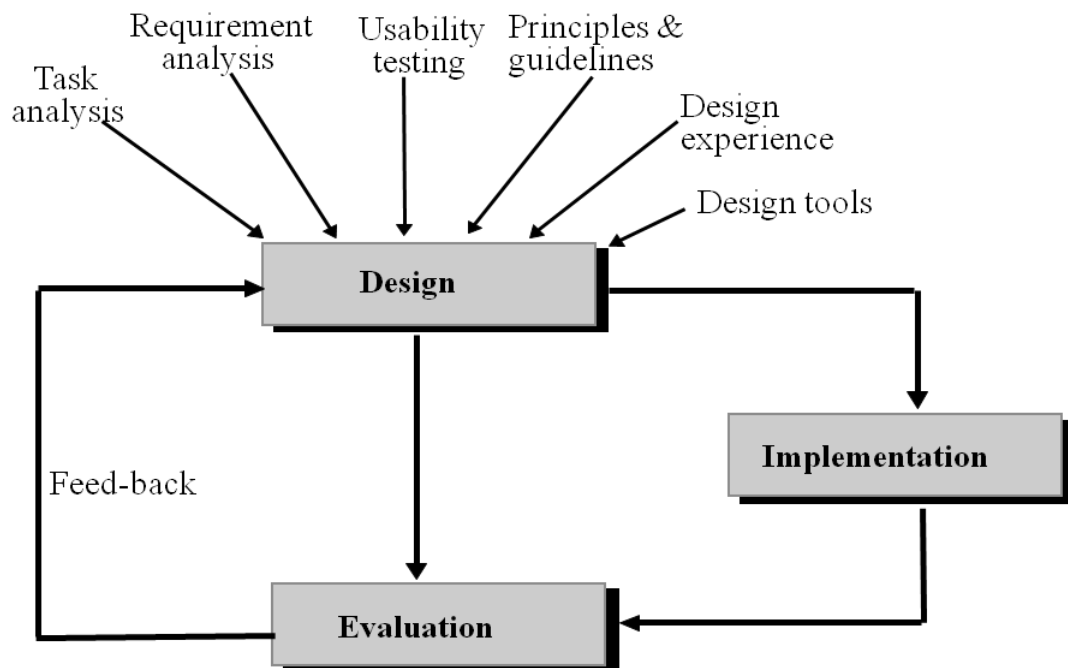
↑ [Back to top](#)



**HCI Issues**

## Human Computer Interaction (HCI): General Design Guidelines and Principles

1. the interface should use terms, concepts and metaphors which are **familiar** to the anticipated class of users,
2. the interface should be appropriately **consistent**, in all aspects of interaction, and in coding fonts, and all HTML content's attributes,
3. satisfying user **expectations**, the user should not be surprised or puzzled by any aspect of the interface,
4. the interface should include some mechanism which allows users to **recover** from their errors,
5. ease of **navigation**, users should feel in control at all times,
6. the interface should incorporate some form of **user guidance**, and
7. use formal **User Interface Design Cycle** to test your interface with different browsers, and group/s of anticipated class of users.



## User Interface Design Cycle

↑ [Back to top](#)



## Responsive Web Design

**Responsive Web Design (RWD):** Wikipedia gives a good exposure to the [history of the development of RWD](#). Ethan Marcotte in his [article](#) (2008/09), and his [book](#) (2011), the concept of responsive web design were further developed under the following three headings: **(i) The Flexible Grid, (ii) Media Queries, and (iii) Flexible images / contents.**

**1. Flexible Grid:** The flexible/fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points, the following diagram shows how the fluid grid behave as the size of the screen reduced to 75% and to 50% of a given screen size without losing any content.

### Fluid Grid

## 100% wide Container

Content Container

Sidebar

## 75% wide Container

Content Container

Sidebar

## 50% wide Container

Content Container

Sidebar

**2. Media queries** allow the page to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser.

#### **Media Queries Application**

**Note 1:** without any media queries the fluid grid containers, shrinks in size. As the size of device screen become smaller, the containers become too small to contain any real content.



**Note 2:** using media queries to remove the floats and change their widths, the content container and sidebar are now able to span the full width of the viewport, allowing breathing room for any existing content.



**3. Flexible images:** Images are also sized in relative units. You should read Ethan Marcotte [article on fluid images](#) techniques.

**How To Create Flexible Images And Media In CSS Layouts:** Typically when adding image and other media such as audio/video players to a web page we set a

width and height using absolute measurements like pixels. This fixes the media size and is completely inflexible.

At one specific point the absolutely fixed media size is probably the ***perfect size for the given context***. However in all other cases it will either be too large or too small for the layout. One way around this is to size media in relative units, rather than absolute pixel dimensions. Setting the width to 100%, and the height to auto, the image will take the width of the encompassing container, and the height take the appropriate value to keep the original aspect ratio. Further more, using media queries the container features may be redefined to be adaptive to the display device characteristics.

The following example is for an image ***"Palawan Island, Philippines"***, size **1600X1200 px**. The image placed in a container of width **40%** for Browser screen width **>= 960px**, and the container width changed to **60%**, **80%** and **100%** for Browser screen width **>= 640px** , **>= 480px** and **>= 380px** respectively.



***What are the resource available for me to help in RWD process:*** There are wide range of resources, each with its own characteristics, advantages and disadvantages. You should have reasonable understanding of the concept, techniques, and limitations of these popular, and widely used RWD resources: Fluid HTML/CSS/JS framework, with flexible grid [Bootstrap](#), and [Twitter Foundation](#),

minimal responsive framework [Skeleton](#), responsive website builder [Website Builder](#), and content management system with widgets, plug-in and customizable themes [WordPress](#), . . . and others. As a general rule, Frameworks and Website Builders allows you, to build responsive website prototypes quickly and effectively, obviously with wide range of limitation. Basic HTML tags, CSS style sheets, and script languages, are very flexible and gives the designer higher degree of freedom, to design Web pages reflecting the intended context, aims and objective of the website.

To ensure proper rendering and touch zooming, add the viewport **meta** tag to your **<head>**:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

You can disable zooming capabilities on mobile devices by adding **user-scalable=no** to the viewport **meta** tag. This disables zooming, meaning users are only able to scroll. Overall, you should consider this carefully and use it only when you need it for particular application, so use caution!

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

Define your container name, initial feature/s, and breakpoint/s in your **<head>** and between the tags **<styal></styal>**:

```
<style>
  .name {
    width: 50%;
  }

  @media screen and (max-width: 640px) {
    .name {
      width: 100%;
    }
  }
</style>
```

Note: at screen width of 640px or less the width of the container will be changed to 100%

Include the container in the appropriate required position in the **<body>**

```
<body>

  <div class="name">

    . . .

  </div>
```

</body>

Note: you may include a nest of tables, and their contents inside the container, or include nest of containers, and their contents inside table/cell. This is not applicable to Twitter Bootstrap and Foundation Frameworks.

## **A) Learn Responsive Design ([RWD](#)) by Examples: (1 Using HTML and CCS)**

**1 Containers Properties:** Note you should view HTML source to learn by examples, (Change the width of your browser to discover how the layout changes).

1.1. [Containers Code](#)

1.2 [Container Properties.](#)

1.3 [Content Flexibility](#)

**2 Page Structure:** (Change the width of your browser to discover how the layout changes).

2.1. [Fluid Page Structure](#) for five containers: Header , Content, Side Bar One, Side Bar Two, and Footer.

2.2. [Adaptive Page Structure](#)

**3 Responsive Menus:** (Change the width of your browser to discover how the menu changes)

3.1. [Fully Responsive Menu](#)

3.2. [Dreamweave Non-Pulldown Resposve Menu](#)

3.3. [Menu Using Slidesho](#)

## **B) Learn Responsive Design ([RWD](#)) by Examples: (2 Using JavaScript)**

4.1. [JQueryUI Menu](#)

4.2 [Pull Down Container](#)

4.3 [Horizontal Scrolling](#)

4.4 [Vertical Scrolling](#)

↑ [Back to top](#)



## Mobile First Responsive Web Design

***What Is Mobile-First Responsive Web Design?*** is a methodology, by which, the web designer starting point, is a page for a given mobile screen/s (portrait and landscape screens). That is to say, the **default** web page is that page designed for given mobile phone.

The main objective is that, the designer voluntarily constrain his/her design to mobile portrait and/or landscape screens. Decisions will be made about the default page structure, and content appearance based on the features and characteristics of the given mobile phone. Once the default page is designed, pages and contents may adapted downward for smaller devices, and upward for larger devices using media query and/or liquid grid systems. However, in some cases, designer may take alternative option/s to design different pages for different devices and programme the web device Browser to select the intended page/s.

Mobile phones, Tablets, PCs, and Desktops have a wide range of screen sizes and resolutions. Now days mobile phone have screens resolutions from 320 to 1242px.

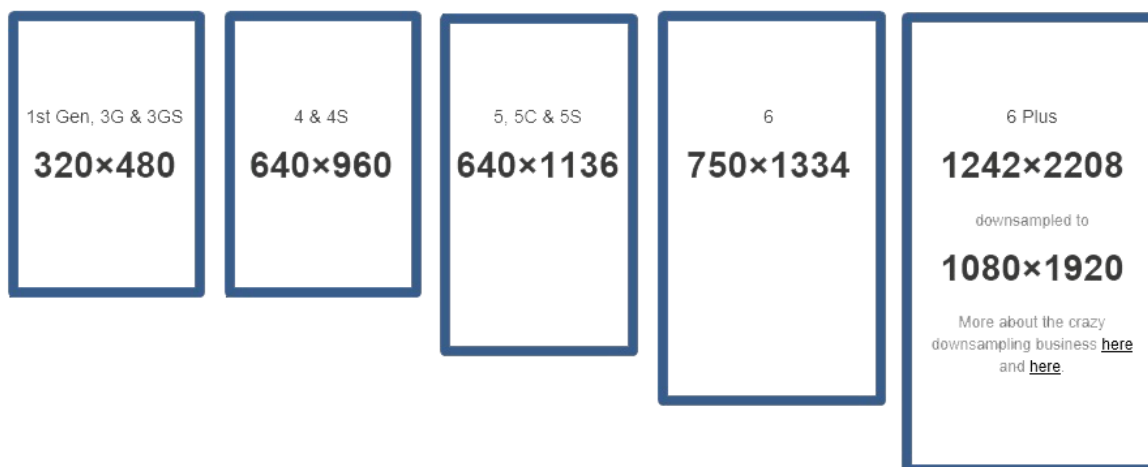


## Responsive Web Design

## Mobile First Web Design



### Smart Phones Screen Resolutions



↑ [Back to top](#)



**The best way forward**

**The best Way Forward:** There are millions, or rather billions of fixed size websites, well developed, and designed. The vast majority of these websites have been carefully designed in

these pages layout, contents features, and appearance to reflect the website goals and objectives.

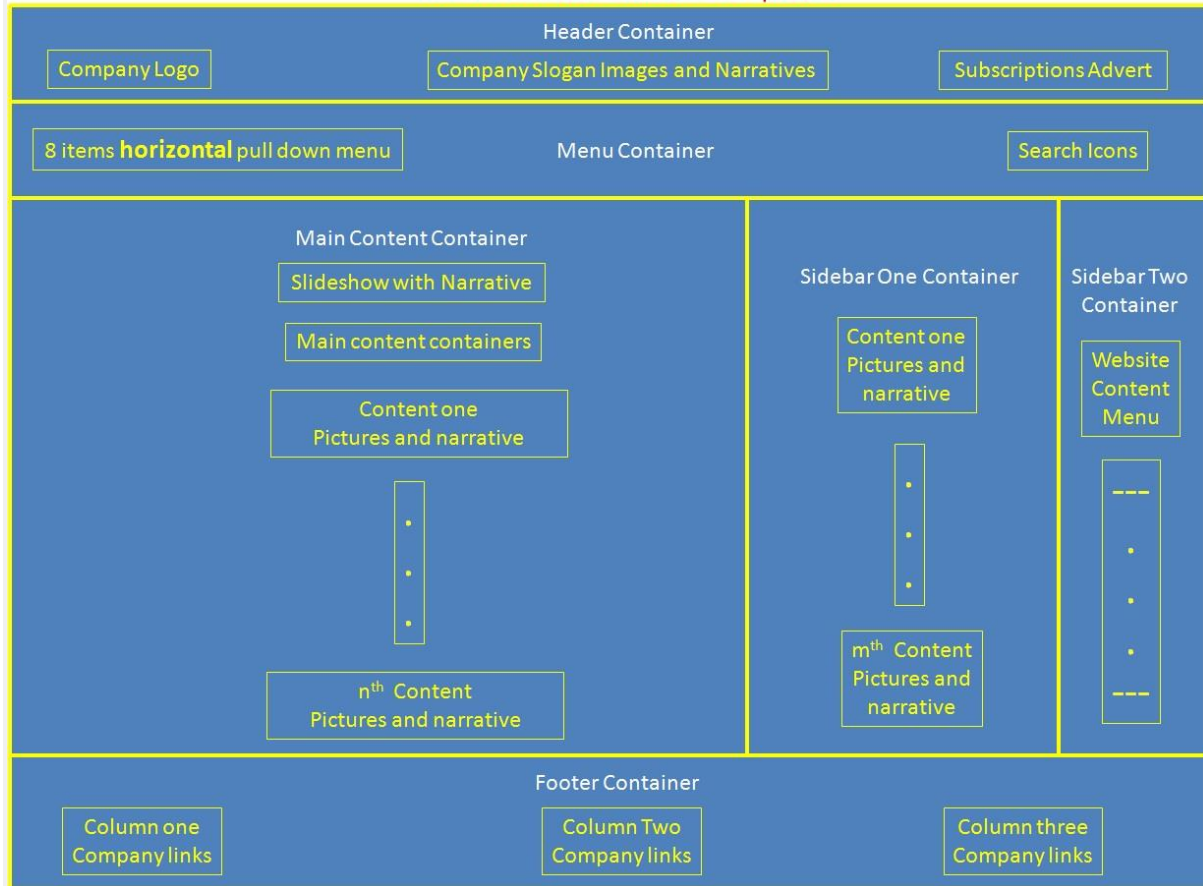
***The question is; Are we going to scrap these websites and go with the limitations of Twitter Framework, and it's apparent lack of designer degree of freedom?:***

No, there are two alternative to grid driven frameworks, **the first**, is using basic HTML tags, CSS style sheet, and script languages, **the second**, which is the best way forward; Design different pages for each category of devices, usually four pages are required, the first for extra small mobiles (320 X 480px), the second for normal size mobiles ( 640 X 960px), the third for large mobiles and tablets ( 750 X 1334px ), and the fourth for PCs and Desktops. You may keep your fixed size web page, originally designed for PCs and Desktop, and design the other three pages, either you use fixed size grid or fluid grid and/or adaptive RWD techniques. Using Media Quires, to allow the device Browser to select the appropriate portrait and landscape screen pages.

This proposal is very time consuming, however the resulting pages are fully optimized to the individual device category. The resulting optimized pages, in it's layout and content appearance, justifies the time spent.

The following example using typical company website page layout of fixed size grid for Browser size  $\geq 960\text{px}$ , newly designed pages for Browser size  $\geq 640\text{px}$ ,  $\geq 480\text{px}$ , and  $\geq 320\text{px}$  respectively. Adjust your screen size to view the four pages.

Page Layout for Browser Screen  $\geq 960$ px  
Maximum table width 960 pixels.



**Note:**

1. With **(Grid driven Framework)**, the four pages are **dependent** on each other. Individual pages layout and content appearance may be readjusted by the given framework to respond to the Browser width, designer degree of freedom is very limited.

2. With **(Basic HTML tags, CSS style sheet, and script languages)**, the four pages are **dependent** on each other. Individual pages layout and content appearance may be readjusted to respond to the Browser width, designer degree of freedom depending on the limitations of @media statement of CSS.

3. With **(The best way forward) proposal**, the four pages are **independent** of each other, designers have complete freedom to change the individual pages layout and content appearance as they may wishes.

**The trade off is between time, and designer degree of freedom.**

↑ [Back to top](#)

## Conclusions and Tips



### Conclusions and Tips

**The above languages allows the Web developer to create Web pages with following features:**

1. Structuring the page layout and style
2. Inserting Text, Graphics, Images, Sound and Video
3. Working with Tables and Containers/Wrappers
4. Creating Forms for user feedback
5. Using Frames
6. Managing and Enhance a Web interactivity
7. Working with databases, spreadsheets and interactive multimedia presentation.

### **Conclusions:**

1. Web page layout may be designed creatively by nests of tables, containers and/or grid system, web content may be inserted in the appropriate cells, Using HTML/CSS tags and/or scripts to define content's attributes and style of appearance,
2. Tables and containers size may be made fixed or fluid as required to achieve appropriate liquidity, and responsiveness to the display device features,
3. Using appropriate CSS media queries, and breakpoints, to make sure that, the web pages, and contents, are adaptive to the display device screen size, and orientations.
4. JavaScript and JavaScript Frameworks (Libraries), are a powerful tools to add high degree of dynamism and interactivity to static web pages. By using JavaScript you may define some or every HTML contents as objects in object-oriented environment,

5. If the web pages contents require the interaction with data files (such as guest books), and/or Web databases stored on the web server, a server-side scripting language such as PHP is needed to manage the process of interaction, and generating the appropriate user's interface screens.

### **Tips:**

1. Always try to use basic HTML tags, and CSS style sheets language. Avoid unnecessary animations and functions. Resist the temptation (were it is possible) to using web pages templates, and grid system frameworks. Templates, and grid driven systems, (sooner or later), loose you considerable time to re-customise or change your page content/s appearance.

[Custom Web Design vs. Website Templates, 5 reasons NOT to use Twitter Bootstrap, and is Twitter Bootstrap right for me? \[flowchart\]](#).

2. Through your navigation procedure, and content's appearance, users quickly develop a mental picture of the interaction process, and of the concept, aims and objectives of your website. Make sure to use intuitive navigating procedure, and be consistent in coding content's shapes, sizes and colours to reflect the given context of use and applications. Avoid were you can, clutter, and long pages (as this one). A simple clean page gives the user the time and the pleasure to understand and digest your message.

3. Adhere to the Golden Eighty Twenty Rule: spend 80% of the time to prepare and polish your contents, and 20% of the time on planning and coding your page layout and appearance. High quality logo, text, graphics, images, audio and video clips, what are make or break your sites.

4. Minimise the use of JavaScript, unless it is required to implement interactive tasks. Many JavaScripts functions are not supported by some laptop, tablets, and smart phones browsers. Some old version of popular browsers have limited support for many JavaScript functions.

5. Do not use PHP for the sake of using it, unless are needed for the particular website needs and requirements.

6. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. Learn Bootstrap, it is an asset, and probably, will be the framework of the future... !

7. Avoid using Flash Animations, and Videos. Flash is not supported by smart phone browsers.
8. Cut and paste content usually carry with it the formatting tags of the original document. Make sure to remove all formatting tags before pasting.
9. Good knowledge of using Microsoft Word, PowerPoint, Excel, Notepad, Paint and Adobe Photoshop will save you time to present your contents in elegant and professional manner.
10. Software such as (i) media convertors (ii) slideshow makers, (iii) menu software, and (v) HTML, and scripting languages editor/s are very useful. You should Invest time to learn how to use them effectively.

Finally, in my humble opinion, fixed size web site is a dying beast, move to responsive design today not tomorrow.

Applications & Performance	HTML / CSS	PHP	Bootstrap	Java Script
Static Page	Yes	Yes	Yes	Yes
Dynamic / Interactive page	No	Yes	limited	Yes
Execution	After page loaded	Before page loaded	After page loaded	After page loaded
Processing Location	Client side	Server side	Client side	Client side
Function Library	No	Yes	limited	Yes
Object-Oriented	No	Yes	limited set	Yes
Database Management	No	Yes	No	Very limited
Structured Language	No	Yes	No	Yes
Processing Speed	Fast	Slow	Medium	Medium
Error debugging	Easy	Medium	Hard	Medium
Page updating	Easy	Medium	Hard	Medium
Designer degree of freedom	Unlimited	limited	Highly limited	Limited

↑ [Back to top](#)

↑ [Back to top](#)



(1) [Responsive vs. adaptive](#), (2) [Responsive Web Design: Layouts and Media Queries](#), (3) [9 GIFs That Explain Responsive Design Brilliantly](#), (4) [Responsive Design in 3 Steps](#), (5) [HTML Responsive Web Design](#), (6) [Simple Responsive Images With CSS Background Images](#), (7) [Fixed, Fluid, Adaptive, and Responsive](#), (8) [Font sizing with Rems](#), (9) [Wiki color formatting help](#), (10) [Help:Using colours](#), (11) [iPhone Screen Resolutions](#), and (12) [Using media queries in JavaScript](#).

## Web Development Resources

### Coding Help

- [Home](#)
- [HTML](#)
- [CSS](#)
- [JavaScript](#)
- [PHP](#)
- [Apache Configuration](#)
- [Regular Expressions](#)
- [General Resources](#)
- [Mozilla Developer Network](#)
- [HTML Dog](#)
- [W3Schools](#)

### Development Tools

- [Recommended Tools](#)
- [WeBuilder code editor](#)
- [W3 HTML Validator](#)
- [W3 CSS Validator](#)

## Collection of some of the most useful web development resources

Here you will find carefully picked links to the **best tutorials, most useful references** and **official specifications** of the most important web programming languages.

### Browse

- [HTML](#)
- [CSS](#)

- [JavaScript](#)
- [PHP](#)
- [Bootstrap](#)
- [Foundation](#)
- [Apache Configuration](#)
- [Regular Expressions](#)

[↑ Back to top](#)

---

