International Journal of Advanced Technology in Engineering and Science www.ijates.com
Volume No.02, Issue No. 11, November 2014 ISSN (online): 2348 – 7550

RESPONSIVENESS

Shivangi Chaudhary

Computer Science & Engineering, Raj Kumar Goel Institute of Technology for Women, Ghaziabad, (India)

ABSTRACT

I am looking forward to develop a responsive website (named as Mosby's Dictionary of Engineering - Computer Science) for majorly computer science students and practitioners to carry with them when studying. This website will provide help to the students and practitioners in learning the computer terms. It will help the students by easy search and as well as they can add the notes which will be helpful to them. Provide option for users to easily access the content of Engineering textbook. Users can easily search and reference the content and terms/definition. It will be based on some latest frontend technologies like HTML5, CSS3, BOOTSTRAP, JQUERY, JAVASCRIPT for increasing the future scope. It focuses to be a responsive website.

Keywords: Responsiveness, HTMl5, CSS3, BOOTSTRAP, JQUERY, JAVASCRIPT, RWD

I. SECTION

1.1 Responsiveness Vs Performance

Software which lacks a decent process management can have poor responsiveness even on a fast machine. On the other hand, even slow hardware can run responsive software.

It is much more important that a system actually spend the available resources in the best way possible. For instance, it makes sense to let the mouse driver run at a very high priority to provide fluid mouse interactions. For long-term operations, such as copying, downloading or transforming big files the most important factor is to provide good user-feedback and not the performance of the operation since it can quite well run in the background, using only spare processor time.

1.2 Influential Factors and Testing

There are many factors that can influence the responsiveness of an interaction system, such as poor design, improper input from users, problems with the operation system or the network. It is generally a good practice to have the designer(s) of the system play the role of the user and run diagnostics to determine if it causes any unreasonably long delays. This will allow them to affect any changes that need to be made before the system is introduced to the users worldwide, thus avoiding such problems earlier on in the systems life-cycle.

1.3 Delays

Long delays can be a major cause of user frustration, or can lead the user to believe the system is not functioning, or that a command or input gesture has been ignored. Responsiveness is therefore considered an

essential usability issue for human-computer-interaction (HCI). The rationale behind the responsiveness principle is that the system should deliver results of an operation to users in a timely and organized manner.

The frustration threshold can be quite different, depending on the situation.

The three steps are 0.1s, 1s, and 10s.

1.4 Solutions to Improve Responsiveness

Although numerous other options may exist, the most frequently used and recommended answers to responsiveness issues are:

Optimizing the process that delivers the output by eliminating wasteful, unproductive output from the algorithm or method by which the result is produced.

A decent process management system, giving highest priority to operations that would otherwise interrupt the user's work flow, such as typing, onscreen buttons, or moving the mouse pointer. Usually there is enough "idle time" in between, for the other operations.

Using idle time to prepare for the operations a user might do next.

Let the user do something productive while the system is busy - for instance, writing information in a form, reading a manual, etc. For instance, in a tabbed browser, the user can read one page while loading another.

Deliver intermediate results, before the operation is finished. For instance, a web page can already be operated before all images are loaded, which will take up the idle time which would otherwise be spent needlessly.

If some waiting is inevitable, a progress indicator can significantly reduce frustration. For short delays, an animated icon might be sufficient. Longer delays are better covered with a progress bar, or, if possible, the system should provide an approximation of the time that an operation is going to take before starting it.

1.5 Responsive Web Design

Responsive web design (RWD) is a web design approach aimed at crafting sites to provide an optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices (from mobile phones to desktop computer monitors).

A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries, an extension of the @media rule.

The fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points.

Flexible images are also sized in relative units, so as to prevent them from displaying outside their containing element.

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.

Mobile first, unobtrusive JavaScript, and progressive enhancement

"Mobile first", unobtrusive JavaScript, and progressive enhancement (strategies for when a new site design is being considered) are related concepts that predated RWD: browsers of basic mobile phones do not understand JavaScript or media queries, so the recommended practice is to create a basic web site, and enhance it for smart phones and PCs—rather than try graceful degradation to make a complex, image-heavy site work on the most basic mobile phones.

Progressive enhancement based on browser-, device-, or feature-detection

Where a web site must support basic mobile devices that lack JavaScript, browser ("user agent") detection (also called "browser sniffing"), and mobile device detection are two ways of deducing if certain HTML and CSS features are supported (as a basis for progressive enhancement)—however, these methods are not completely reliable unless used in conjunction with a device capabilities database.

For more capable mobile phones and PCs, JavaScript frameworks like Modernizer, jQuery, and jQuery Mobile that can directly test browser support for HTML/CSS features (or identify the device or user agent) are popular. Polyfills can be used to add support for features—e.g. to support media queries (required for RWD), and enhance HTML5 support, on Internet Explorer. Feature detection also might not be completely reliable: some may report that a feature is available, when it is either missing or so poorly implemented that it is effectively nonfunctional.

1.6 Challenges, And Other Approaches

Luke Wroblewski has summarized some of the RWD and mobile design challenges, and created a catalog of multi-device layout patterns. He suggests that, compared with a simple RWD approach, device experience or RESS (responsive web design with server-side components) approaches can provide a user experience that is better optimized for mobile devices. Server-side "dynamic CSS" implementation of stylesheet languages like Sass or Incentivated's MML can be part of such an approach by accessing a server based API which handles the device (typically mobile handset) differences in conjunction with a device capabilities database in order to improve usability. RESS is more expensive to develop, requiring more than just client-side logic, and so tends to be reserved for organizations with larger budgets. Google recommends responsive design for smartphone websites over other approaches.

Although many publishers are starting to implement responsive designs, one ongoing challenge for RWD is that some banner advertisements and videos are not fluid. However, search advertising and (banner) display advertising support specific device platform targeting and different advertisement size formats for desktop, smartphone, and basic mobile devices. Different landing page URLs can be used for different platforms, or Ajax can be used to display different advertisement variants on a page.

An alternative to RWD is the method of Adaptive Web Delivery or AWD that is adopted by consumer brands worldwide. Although it is very similar to Responsive Web Design, with adaptive delivery the most significant difference is that the server hosting the website detects the devices making requests to it, and uses this information to deliver different batches of HTML and CSS code based on the characteristics of the device that have been detected.

There are now many ways of validating and testing RWD designs, ranging from mobile site validators and mobile emulators to simultaneous testing tools like Adobe Edge Inspect. The Firefox browser and the Chrome console offer responsive design viewport resizing tools, as do third parties.

II. SECTION

2.1 History

The technique of adapting a site's layout to a device's display was first written about by Cameron Adams in 2004. Ethan Marcotte coined the term responsive web design (RWD) in a May 2010 article in A List Apart. He described the theory and practice of responsive web design in his brief 2011 book titled Responsive Web

Design. Responsive design was listed as #2 in Top Web Design Trends for 2012 by .net magazine after progressive enhancement at #1.

Mashable called 2013 the Year of Responsive Web Design. Many other sources have recommended responsive design as a cost-effective alternative to mobile applications.

Forbes featured a piece, 'Why You Need To Prioritize Responsive Design Now' where the importance was made clear that having a mobile version of your website isn't enough anymore. Jody Resnick, President of Trighton Interactive stated in his interview with Forbes, "Responsive websites simplify internet marketing and SEO. Instead of having to develop and manage content for multiple websites, businesses with responsive sites can take a unified approach to content management because they have only the one responsive site to manage.

Resnick predicts, "As the internet transforms further into a platform of services and user interfaces that tie those services together, leveraging this technology in the future will allow companies to integrate a plethora of back-end services, such as Facebook, Twitter, Salesforce.com, and Amazon Web Services, and then present the integrated data back out the front-end iad layer on a responsive design so the application looks great on all devices without custom coding needed for each device or screen size."

Some believe that responsive design will be more prevalent than native apps simply because of the browser compatibility and the cost associated with programming the apps.

CSS frameworks are pre-prepared software frameworks that are meant to allow for easier, more standards-compliant web design using the Cascading Style Sheets language. Most of these frameworks contain at least a grid. More functional frameworks also come with more features and additional JavaScript based functions, but mostly design oriented and unobtrusive. This differentiates these from functional and full JavaScript frameworks.

Some notable and widely used examples are Bootstrap or Foundation .

CSS frameworks offer different modules and tools:

Reset-Stylesheet

grid especially for responsive web design

web typography

set of icons in sprites or iconfonts

styling for tooltips, buttons, elements of forms

parts of graphical user interfaces like Accordion, tabs, slideshow or Modal windows (Lightbox)

Equalizer to create equal height content

often used css helper classes (left, hide)

Bigger frameworks use CSS interpreter like LESS or SASS.

Bootstrap is a free 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. In June 2014 it was the No.1 project on GitHub with 71,000+ stars and 26,000+ forks, with a user base including MSNBC and NASA.

2.2 Origin

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to

inconsistencies and a high maintenance burden. According to Twitter developer Mark Otto:

"A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company."

The first deployment under real conditions happened during Twitter's first Hackweek. Mark Otto showed some colleagues how to accelerate their project's development with the help of the toolkit. As a result, dozens of teams have moved to the framework.

In August 2011, Twitter released Bootstrap as open source. In February 2012, it was the most starred GitHub development project, a position it still holds in June 2014.

2.3 Features

Bootstrap is compatible with the latest versions of all major browsers. It gracefully degrades when used on older browsers such as Internet Explorer 8.

Since version 2.0 it also supports responsive web design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with version 3.0, Bootstrap adopted a mobile first design philosophy, emphasizing responsive design by default.

Bootstrap is open source and available on GitHub. Developers are encouraged to participate in the project and make their own contributions to the platform.

Recently, community members have translated Bootstrap's documentation into various languages, including Chinese, Spanish and Russian.

2.4 Structure and function

Bootstrap is modular and consists essentially of a series of LESS stylesheets that implement the various components of the toolkit. A stylesheet called bootstrap. less includes the components stylesheets. Developers can adapt the Bootstrap file itself, selecting the components they wish to use in their project.

Adjustments are possible to a limited extent through a central configuration stylesheet. More profound changes are possible by the LESS declarations.

The use of LESS stylesheet language allows the use of variables, functions and operators, nested selectors, as well as so-called mixins.

Since version 2.0, the configuration of Bootstrap also has a special "Customize" option in the documentation. Moreover, the developer chooses on a form the desired components and adjusts, if necessary, the values of various options to their needs. The subsequently generated package already includes the pre-built CSS style sheet.

Grid system and responsive design comes standard with a 1170 pixel wide, grid layout. Alternatively, the developer can use a variable-width layout. For both cases, the toolkit has four variations to make use of different resolutions and types of devices: mobile phones, portrait and landscape, tablets and PCs with low and high resolution. Each variation adjusts the width of the columns.

The CSS stylesheet

Bootstrap provides a set of stylesheets that provide basic style definitions for all key HTML components. These

provide a uniform, modern appearance for formatting text, tables and form elements.

Re-usable components

In addition to the regular HTML elements, Bootstrap contains other commonly used interface elements. These include buttons with advanced features (e.g. grouping of buttons or buttons with drop-down option, make and navigation lists, horizontal and vertical tabs, navigation, breadcrumb navigation, pagination, etc.), labels, advanced typographic capabilities, thumbnails, warning messages and a progress bar.

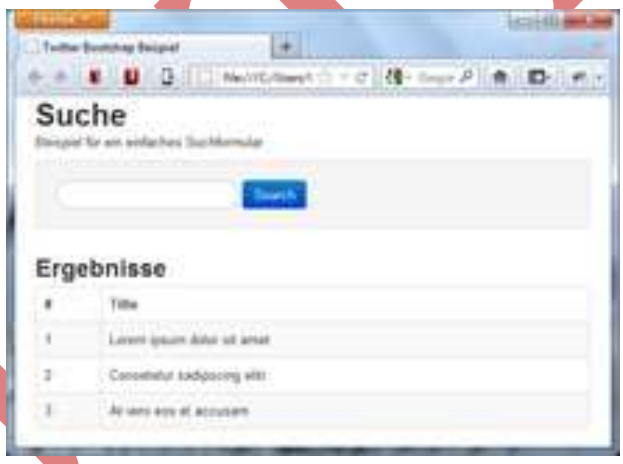
JavaScript components

Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields. In version 2.0, the following JavaScript plugins are supported: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Typeahead.

An implementation of Twitter Bootstrap using the Dojo Toolkit is also available. It is called Dojo Bootstrap and is a port of the Twitter Bootstrap plugins. It uses 100% Dojo code and has support for AMD (Asynchronous Module Definition).

Likewise, Bootstrap controls for AngularJS are also available; they are called UI Bootstrap. This port reuses some Bootstrap markup.

2.5 Use



A Bootstrap-designed document shown in Mozilla Firefox 10

To use Bootstrap in an HTML page, the developer downloads the Bootstrap CSS stylesheet and includes a link in the HTML file.

(The developer can also compile the CSS file from the downloaded LESS or SASS stylesheets, with a special compiler.)

If the developer wants to use the JavaScript components, they must be referenced along with the jQuery library in the HTML document.

The following example illustrates how this works. The HTML code defines a simple search form and a list of results in tabular form. The page consists of HTML 5 elements and CSS information according to the Bootstrap documentation. The figure shows the representation of the document in Mozilla Firefox 10.

```
<!DOCTYPE html>
<html>
 <head>
  <title>Example of Twitter Bootstrap</title>
  <!-- Include the bootstrap stylesheets -->
  <link href="http://netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css"
rel="stylesheet">
 </head>

 <body>
  <div class="container">
   <h1>Search</h1>
   <label>Example for a simple search form.</label>

   <!-- Search form with input field and button -->
   <form class="well form-search">
    <input type="text" class="input-medium search-query">
    <button type="submit" class="btn btn-primary">Search</button>
   </form>

   <h2>Results</h2>

   <!-- Table with alternating cell background color and outer sframe -->
   <table class="table table-striped table-bordered">
    <thead>
     <tr>
      <th>#</th>
      <th>Title</th>
     </tr>
    </thead>
    <tbody>
     <tr>
      <td>1</td>
      <td>Lorem ipsum dolor ...</td>
```

```
      </tr>
      <tr>
        <td>2</td>
        <td>Ut enim ad minim veniam, ...</td>
      </tr>
      <tr>
        <td>3</td>
        <td>Duis aute irure dolor ...</td>
      </tr>
    </tbody>
  </table>
</div>
<!-- JavaScript placed at the end of the document so the pages load faster -->
<!-- Optional: Include the jQuery library -->
<script src="http:/code.jquery.com/jquery-1.7.2.min.js"></script>

<!-- Optional: Incorporate the Bootstrap JavaScript plugins -->
<script
src="http:/netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>
 </body>
</html>
```

Creating a simple layout grid (Fluid)

```
<div class="row">
  <div class="col-md-4">hi</div>
  <div class="col-md-4">...</div>
  <div class="col-md-4">...</div>
  <div class="col-md-4">...</div>
  <div class="col-md-4">...</div>
</div>
```

REFERENCES

[1] Mr. Himanshu Kumar, Program Manager, Infopro,Noida,U.P.,India,2014, himanshu.kumar@infoprolearning.com

[2] Mr. Pradeep Kapoor, Project Manager,Infopro,Noida,U.P.,India,2014, pardeep.kapoor@infoprolearning.com

[3] https://www.google.com

[4] http://getbootstrap.com/getting-started/