

CS5030: Web Technologies

Practical 2 (A2 Website)



University of
St Andrews

Matriculation Id: 220032472

Contents

1 Responsive Web	3
1.1 Introduction	3
1.2 Challenges faced while building responsive design	3
1.3 Bootstrap framework	4
1.4 Validation	4
2 Adaptive Mobile	5
2.1 Introduction	5
2.2 Challenges faced while building adaptive design	5
2.3 Validation	6
2.3.1 HTML Error	6
2.3.2 CSS Error	6
3 Adaptive Desktop	7
3.1 Introduction	7
3.2 Validation	7
Short Reflective session	7
References	8

Responsive web

1.1 Introduction

Current trends and improvements offered in terms of both software and hardware led us to develop websites in large number and to cater to vast screen sizes, which includes smart phones, tablets, dual screened mobile phones and large displays.

These comes with diverse characteristics with respect to their screen resolutions. When building the News website for assignment 2, I explored the possibility of building one webpage that suits all screen sizes, i.e., '**Responsive Web design**'.

This kind of webpage helped solve a lot of problems for the website owner as well as proved to be sustainable for the environment. Building responsive website made it mobile friendly as well as rendered as expected on larger screens. Website automatically scales its contents to match the screen size it is being viewed on. This prevented unnecessary zooming, panning or scrolling to view the contents.

The effectiveness of such a website correlated to the following benefits:

- One page fits all screens, therefore being **flexible**.
- Developer need not worry about the **screen-size**, the website is being rendered upon.
- **Single deployment**, improving the effective time to ship the product to market.
- **Ease of management** as the codebase was relatively smaller than 2 separate adaptive websites built to cater to different screen sizes.
- Improved **User experience**, as the transition from desktop to mobile and vice versa was seamless.

Google's page indexing and search engine result, ranking mobile friendly websites higher in the order and promotes responsive design as a standard when it comes to **SEO** (Search Engine Optimization) [1]. Because it fosters long-term, cost-effective business growth, SEO is economically sustainable. It is socially sustainable because it is about helping people find what they are looking for. Additionally, it is environmentally sustainable because it lessens the need for users to visit (many) other websites that might be less environmentally beneficial.

1.2 Challenges faced while building responsive design

There were intricacies when building a website that renders on all devices and validated using an emulator. The following difficulties was faced when designing the same:

- **Testing**: Each element added to the website needed to be tested on all possible screen resolutions to check for backlogs and breakages. Focus was to not lose the element when being rendered on different screens. Care needed to be taken after every addition, to see if there were no effect on other components of the page.
- **Scalability**: Each element in the webpage has its own styles of implementation with regards to scalability. Font, images, and videos (both from local and referenced websites) needed to scale as per the screen. This was rectified by using relative values such as **EM** (relative to parent), **%** (relative to parent), **VW** and **VH** (relative to viewports width and height), rather than using absolute value such as **PX** (pixels) in most cases.

1.3. Bootstrap Framework

The responsive website's components were built using the Bootstrap framework since it has a fluid grid structure that expands suitably up to 12 columns as the device or viewport size rises. It has strong mixins for creating more semantic layouts as well as preset classes for quick layout options. Grid layout offered by bootstrap was used in home screen to render cards within a row and 4 columns.

Bootstrap Framework via CDN : The following code for bootstrap compiled CSS was included in responsive websites to use components and styles provided by Bootstrap.

Version : Bootstrap 4.5.3

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-X8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkGIXeMed4M0jl1fIDPvg6uqKI2xXr2"
crossorigin="anonymous">
```

1.4. Validation

On validating the page after development, the following errors were found

1. **<h1>** tag was used inside lower ranked elements. h1 is a blocking element so it should not be inside a non-blocking element . This violates the html standards [2]. This was later replaced with **** tag.

1. **Warning** Consider using the `h1` element as a top-level heading only (all `h1` elements are treated as top-level headings by many screen readers and other tools).
From line 28, column 11; to line 28, column 33
`<h1 class="logo-title">` St. A

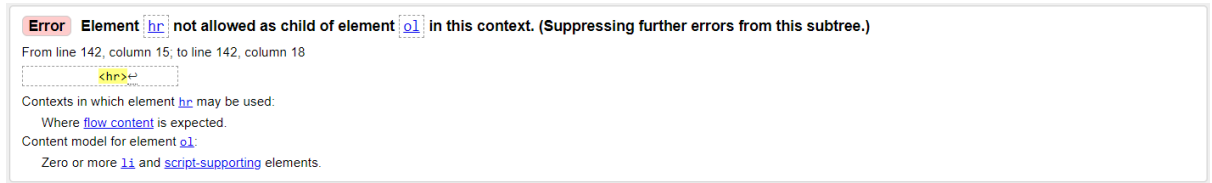
2. The following error was thrown like the one stated above. Rectification used for this error was to replace **p1** with **p** tag.

Error Element `p2` not allowed as child of element `div` in this context. (Suppressing further errors from this subtree.)
From line 42, column 7; to line 42, column 29
`<p2 class="lead my-3">Update`
Content model for element `div`:
If the element is a child of a `div` element: one or more `div` elements followed by one or more `div` elements, optionally intermixed with `script` supporting elements.
If the element is not a child of a `div` element: `flow content`.

3. As per the Html specification stated in **WHATWG** for '**a**' tag [3], there must be no interactive content as descendant, and **iframe** is an interactive element. The **source** used in the iframe element allows for navigation and interaction, thereby removing the dependency of wrapping around **<a>**

Error The element `iframe` must not appear as a descendant of the `a` element.
From line 137, column 31; to line 139, column 36
`href="#"><iframe src="https://www.youtube.com/embed/QSkKG5bthSA" title="YouTube video player" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>`

4. **<hr>** tag was used immediately after ordered list **** tag was used, therefor it had to be removed and added after the list item is added. Rectification used to resolve this was: **<hr>item** is not valid, but **item<hr >** is valid as the direct child of **** is ****. [4]



The Screenshot of the page rendered on a 1739 X 979-pixel dimension display is added as part of the source code, under the directory name 'screenshot'.

Adaptive Mobile

2.1 Introduction

The mobile version was built for 412px * 900px display and tested on Desktop browser's device emulator with 'fit to window' zoom option selected. This website is non-responsive and will only be rendered correctly on the above-mentioned resolutions. Any drastic change from the aforementioned resolution will lead to breakage and run overs in terms of the responsiveness.

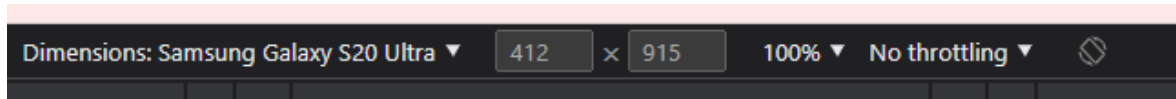


Figure illustrates the screen resolution considered to build adaptive mobile version of the website. The dimension is provided by google chrome web browser under its list of devices.

2.2 Challenges faced while building adaptive design

This approach towards building a native website came with a set of challenges:

- **Testing:** The websites are built and viewed on desktop browsers (large screen) and is heavily dependent on tools offered by the browsers such as 'Toggle device toolbar' menu in Google chrome's DevTools, 'Toggle device emulation' on Microsoft edge, etc. These offer a range a device list from which preferred screen dimension was chosen. The rendered page was inconsistent when switching between browsers, leading to inconvenience in adding adequate CSS styles for respective elements. Google chrome was chosen as a default browser when testing for adaptive screens.
- **Static elements and layout:** Mobile first approach when building a responsive website, gave good insight on understanding the scalability. But in case of adaptive web design, constraining to a specific fixed pixels and making a static layout at the cost of sacrificing user experience when scaled, seemed difficult to comprehend.

2.3 Validation

2.3.1 HTML Error

On validating the adaptive mobile after development, the following errors were found:

1. The relative path added for the images was using



Rectification:

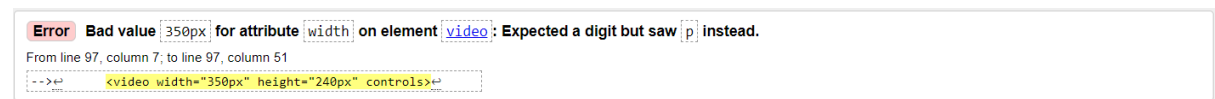
Bad input : `src="images\Carousel\category.png"`

Correct form of relative path was to use `'/'` : `src="images/Carousel/category.png"`

2. Unclose elements were caught by the validator and was added post validation.
Missed a div element closure before the body completion.

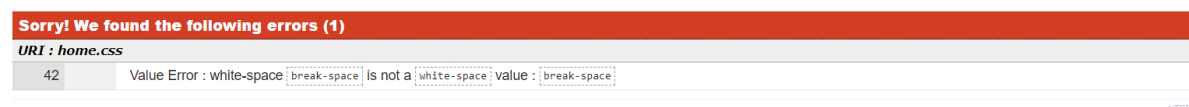


3. `` or `<video>` elements accept a width attribute to specify the size in pixels. This number should not contain units or % and can only be an integer. If you need to specify a percentage width, you need to do that with CSS. The browser truncates the value "420px" for key "width" to its numeric prefix and renders it without any difficulties.



2.3.2 CSS Error

W3C CSS Validator results for home.css (CSS level 3 + SVG)



The CSS validator [5] threw the above shown error, because of incorrect spelling used on the CSS property value.

The Screenshot of the page rendered on a 1739px * 979px dimension display is added as part of the source code, under the directory name `'screenshot'`. The screenshot shows how an adaptive website built for mobile screen renders on a larger display.

Adaptive Desktop

3.1 Introduction

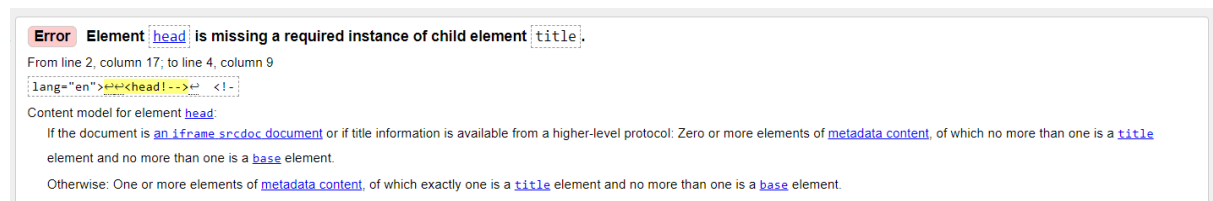
The Adaptive desktop version was built and tested on 1600px * 900px display with 'fit to window' zoom option selected. This website is non-responsive and will only be rendered correctly on the above-mentioned resolutions. Any drastic change from the aforementioned resolution will lead to breakage and run overs in terms of the responsiveness.

The challenges faced were similar to the ones faced when building an adaptive mobile website.

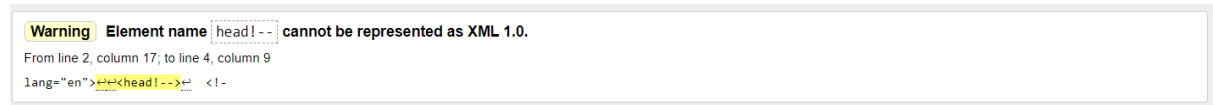
3.2 Validation

The following errors were identified on validating HTML and CSS.

1. This error was thrown when the head tag was written in the form <head!-->



This led to the following warning



Document type of HTML was rendered correctly on the browsers even when these validation errors existed indicating that the modern-day browser ignores the above-mentioned errors.

The Screenshot of the page rendered on a 1600 X 900-pixel dimension display is added as part of the source code, under the directory name 'screenshot'.

Short reflective session

- The complexities that arise due to improper understanding of the Responsive web designs vs Adaptive web design needs to be avoided by comparing multiple websites built for specific screens, for each design type before diving into the development process.
- Had to look up at multiple resources to conclude good practises and standards rather than referring to single source. There were discrepancies when referring to multiple online web pages. A reliable source of information needed to be analysed and followed throughout the coursework to maintain the consistency of design.

The references mentioned in the lecture slides (MDN Web docs) were of a good help when understanding the attributes supported by each of the tags used in the code, as well as the possible values for each CSS properties used.

REFERENCES

- [1] Google Search essentials , <https://developers.google.com/search/docs/essentials>, accessed on 10-11-2022
- [2] HTML5, <https://www.w3.org/TR/2010/WD-html5-20101019/sections.html#headings-and-sections/>, accessed 17-11-2022
- [3] HTML living standards, <https://html.spec.whatwg.org/multipage/text-level-semantics.html#the-a-element>, updated on 19-11-2022
- [4] HTML Checking for Large Sites, <https://rocketvalidator.com/html-validation/element-div-not-allowed-as-child-of-element-ul-in-this-context-suppressing-further-errors-from-this-subtree>, accessed 17-11-2022.
- [5] CSS validator, <https://jigsaw.w3.org/css-validator/>, accessed on 17-11-2002