Bachelor Thesis Project

Quality Assessment of Web Pages
- is it worth the trouble?

Author: Imad Collin
Supervisor: Dr. Rüdiger Lincke
Semester: VT 2016
Subject: Computer Science
Abstract

The number of new websites is increasing every day, especially within several domains such as educations, business and government. According to the most recent statistics, high-quality web pages are strategically needed for many of these domains and organizations. As for software in general, web pages need to be tested to assure proper function and compliance with specifications. In particular, the HTML and CSS that is used should be valid. Even more important is good performance (e.g., load times), SEO, adherence to best practices, accessibility, etc. to assure a good user experience (on most browser platforms). However, performing and acting on these tests costs valuable time and resources. Therefore, it would be interesting to know for real-world web pages, which test scores are acceptable, which errors should/must be fixed, what recommendations should/must be followed, what is the benefit, and to what extent does it matter?

With the help of common tools used for an automatic quality assessment of web pages, we attempt to determine the common types of validation errors in web based projects. We automatically run these tools on a large number of web pages to collect the required data, also we did some manual tests for analyzing the actual impact of the identified errors/warnings on the website appearance.

Keywords: Quality evaluation software, web diagnostic online tools cross-platform, web apps, API-Getaway, Web-Services.
Preface

I would like to say that I am grateful to my supervisor, Dr. Rüdiger Lincke for all supports, courage, and advice for achieving this work and also all support and mentoring along the way until finishing this work. I am grateful to have been taught and guided by such a wonderful person.
# Contents

## 1 Introduction

1.1 Background  
1.2 Previous research  
1.3 Problem formulation  
1.4 Motivation  
1.5 Research Question  
1.6 Scope/Limitation  
1.7 Target group  
1.8 Outline  

## 2 Method

2.1 Scientific Approach  
2.2 Method Description  
2.3 Reliability and Validity  
2.4 Ethical Considerations  

## 3 Implementation  

## 4 Results and Analysis  

## 5 Discussion  

## 6 Conclusion

6.1 Future Research  

## References  

## A Appendix 1
1 Introduction
This is a degree project in computer science investigating the impact of quality related issues on web pages and their appearance.

1.1 Background
Web browsers are rather forgiving when it comes to errors caused by HTML, CSS or JavaScript. For example, missing/wrong attributes, missing tags/closing tags, faulty element structure, use of bad practices, etc. may remain undetected during development, since the browser attempts to correct them silently. Also, the operation of a web page is highly dependent on the web server configuration. Providing an optimal configuration requires a complete different skillset from that required to develop a web page. A badly configured server might take the fun out of the best web page [2].

Many researchers have proven that the quality of web pages impacts the user/consumer trust [3] and that higher quality leads to better results. Today, more than 4.76 billion webpages [4] are accessible on the Internet, but the quality of these sites differ greatly. Moreover, with the rapid development of web techniques, there are several differing points of view, standards and metrics for determining website quality today. There are also some tools used for measuring website quality.

This study focuses on a few common tools used for automatic quality assessment of web pages: W3C validator, Sitespeed.io, CSE validator and Xenu’s Link Sleuth. These tools are used to determine common quality errors in web pages, and also they provide useful information for avoiding the detected errors.

In addition to these tools, there will be some manual testing for checking the impact that the detected errors have on web page appearance.

1.2 Previous research
A significant amount of research has been done regarding website quality assessment. Often researchers were trying to indicate, examine or measure the website quality from different perspectives, such as user perspective, usability or even the quality of the content [5].

Several researchers were trying to investigate in the same domain, validating the web pages, regarding design quality and infrastructure characteristics such as design parameters (e.g. HTML errors, links validity, load time and the browser compatibility). In these researchers, the results are based on some online diagnostic tools for
evaluating purpose. Similarly, we notice that most of the research was about how the web page quality can impact the users and their interactions, based on human testing and investigating in the literature. A good of such research is “a quality evaluation model for online shopping website” [6].

1.3 Problem Formulation
Measuring web page quality was and still the substantial task for most of the recent researches and studies [7]. Today a lot of tools and techniques exist for automatic quality assessment of web pages. These tools were developed and are used for validating the correctness of HTML and CSS, check web pages and servers for (load) performance, SEO, accessibility, security related issues, etc.

These tools usually run a series of automated tests and provide a comprehensive list/report of found issues, recommendations, and even scores. However, fulfilling most/all of these recommendations can be quite time-consuming, in particular for low-skilled hobbyists or IT professionals. Therefore, to make the quality assessment process more effective, it is important to define what attributes, and metrics (existing tools can measure that) are most important for web page quality. The tools must also be examined to see whether or not they can reliably measure these attributes and metrics.

Therefore, it would be interesting to know for real-world web pages: which test scores are acceptable, which errors should/must be fixed, what recommendations should/must be followed, what is the benefit, and to what extent does it matter? This might make it possible to find a good balance between quality and cost.

1.4 Motivation
Websites play a very important role in the delivery of information [8]. Assessing the quality of these websites was and still is a challenging area of most researchers today [9]. Also, there are some published tools and techniques that measure various aspects of website quality including usability, accessibility and validity. Talking about the quality assessment criteria, many of the questions addressed.

The goal of this study is to identify the most common validation errors that appear during the automated test, and what test scores can be considered acceptable. The study will also examine how reliable the tools used to check HTML or CSS are at detecting errors, and if the detected errors impact the appearance of the web page. This will help us examine how to reliably assess the quality of web pages while reducing the time and resources needed to do
1.5 Research Questions

Answering the following research questions should help us to solve the problems mentioned above.

<table>
<thead>
<tr>
<th>RQ1.</th>
<th>Do validation errors (and warnings) have an impact on the web page? If yes, what is an acceptable upper threshold?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2.</td>
<td>What are the most common validation errors?</td>
</tr>
<tr>
<td>RQ3.</td>
<td>Which are web quality tool recommendations worth following?</td>
</tr>
<tr>
<td>RQ4.</td>
<td>Which are the most common recommendations/scores for improving a web page?</td>
</tr>
<tr>
<td>RQ5.</td>
<td>What is an acceptable score for a certain tool/advice?</td>
</tr>
</tbody>
</table>

The expected results of these questions are to define the most common types of validation errors in web pages and represent the most common tools used for web pages evaluating such as **W3C validator**, **Sitespeed.io**, **CSE validator** and **Xenu's Link Sleuth** checker.

This will be by using a big number of web pages and then, sorting the result to show the most common validation errors. As well as defining the reliability and correctness of these tools.

The second part of the result will show the impact of these errors on the quality of a web page, this will be by implementing and testing a website then running one of these tools against this web page to see the behavior of the tool.

1.6 Scope/Limitation

There are some limitations to the preparation workflow. The first limitation is the way we can show how the validation errors impact web pages.

Since we cannot trust the machine’s result by using automate tools for the testing, therefore, the testing will be based on **human** assessment and experience. In other words, this will be by building a web page with widely used attributes, then testing these webpages elements/attributes by one of validation tools. After that, we will correct produced error **manually** and see the differences in term of appearance.
The second limitation is the number of tools commonly used for evaluating web pages, the limitation is the limited number of free tools available to use, for that reason the work will be limited to some free tools and libraries.

The third limitation is the testing and implementing environment this work, and its related results will be based on Mac OS. Finally, the fourth limitation is the time for achieving this work. For this study, more time would yield better results.

1.7 Target group
The target groups of this study are the web developers and research communities that work with website quality assurance, as well as programmers who have concerns about optimizing their work. In general, all people who want to achieve a high-quality of the website also user experience evolution.

1.8 Outline
The remainder of this report is structured as follows:

Chapter 2: defines the methodology used to complete the work.
Chapter 3: defines the work approach for setting the environment and implementing the required tools.
Chapter 4: shows the results which were produced by running tools and represents the outputs.
Chapter 5: compares the result tables showed in chapter 4.
Chapter 6: contains the discussion and some observations during the workflow.
Chapter 7: concludes our report and presents ideas for future work.
2 Method

This chapter contains a description of the method which will be used to answer the research questions. Furthermore, we discuss the reliability and validity of our study as well as ethical considerations.

2.1 Scientific Approach

The scientific approach for this thesis is an inductive approach using an empirical method for quantitative data collected to answer the research questions which have been covered in Section 1.5.

To answer some research questions we might require qualitative data which we will collect through manual assessment of limited samples.

2.2 Method Description

To answer our research questions, we conduct some experiments as following:

We analyze a selection of websites (see Section 2.2.1 regarding selection criteria) with selected tools (see Section 2.2.2 regarding selection criteria) to extract relevant data. In particular, we extract data from:

- 200 websites with SiteSpeed.io tool.
- 200 website links with Xenu\'s Link Sleuth\(^2\) (link checker).
- 200 websites with validator.nu.
- An appropriate number of tests applied to single HTML/CSS files.

For each of these experiments, the data are collected by implementing and modifying a set of frameworks and libraries with different patterns.

Since the work consists of some manual tests, the results are limited to user experience, and the impact of any validation errors on web pages’ appearance also will depend on the user experience.

In the following subsections, we provide details on how we selected websites and tools.

2.2.1 Selection of Websites

To get representative research result, we want to answer our research question based on current, reachable and popular web pages. There are several lists of popular web pages. We choose our web pages from the Alexa top 500 sites on
the web index [10]. The index is calculated as follows:

“The sites in the top sites lists are ordered by their 1-month Alexa traffic. The 1-month rank is calculated using a combination of average daily visitors and page views over the past month. The site with the highest combination of visitors and page views is ranked 1 in the first order in the list.” [10]

In our study, we include the 200 top ranked websites from the Global Alexa top 500 list. This should represent the state-of-the-art regarding the quality of web pages and user experiences and lead to significant results [11].

2.2.2 Web Page Assessment Tools Selection

Validating and testing web page’s contents, structures and configurations are an important aspect of creating a successful and accessible website [12] [13]. Instead of implementing our validating tools, we choose to use existing tools and we have selected some tools which must meet the following criteria:

- It must be modern and knows the state-of-the-art in web page assessment.
- Can be used in an automated way, i.e., be scripted to assess our selected web pages in an automated way.
- Is open source, i.e., we can use it for free and have free access to it.
- Provides a selection of metrics we can use to assess the quality of a web page in a quantitative way.
- Allows to assess website performance (different aspects), the validity of CSS and HTML structure, and checking links.

After some research we selected the following tools:

2.2.2.1 Sitespeed.io

This tool helps to analyze and optimize web pages regarding speed, performance and accessibility. This tool is based on performance best practices rules, and also is collecting browser metrics by the use of User Timing and Navigation Timing API [13].

The reasons for choosing this tool is because it is open source and covers different criteria for testing web pages and reporting the results. The tool has the following features:
- It allows testing web pages against Web Performance best practice
- It allows collecting timing metrics such as Navigation Timing API
- It can test multiple pages to save time and get more accurate results and metrics
- It allows modification so we can write our plugins and test what we need.
- Finally, Graphite which is a create advantage.

There were a few versions of these tools where version 4.0 supports HTTP/2. It uses Phantom JS and Y-Slow for analyzing purposes (for more details see [12]).

2.2.2.2 W3C Nu HTML Checker

This is an online tool for checking HTML it is an ongoing experiment in better HTML checking. This tool was used for detecting unintended mistakes.

The tool attempts to catch these mistakes and provide appropriate messages for certain kinds of potential problems. The tool has the following features:

- The tool is available online and free to use.
- There are some alternatives such as Grunt plugin, Gulp plugin and more.
- The reason for choosing this tool is because it counted as a baseline in HTML checking and one of the most reliable tools. (For more information, please check [14]).

2.2.2.3 Xenu's Link Sleuth

This tool will be used for testing links within the web pages even though some tools such as Site speed.io or CSE allow us to check the broken links within the current page and give a warning message for broken links, but it will be interesting to use another like Xenu's Link Sleuth tool for this specific task.

For this purpose, the used tool is called Xenu’s Link Sleuth. This tool is used to check the web pages for broken/dead links. The tool can display a continuous updated list of those URLs, sorted in multiple ways depending on
criteria with an ability for generating the corresponding report along with the test. The tool has the following features:

- Free to use and no licenses/limitation needed
- Re-check broken link, in case of temporary network errors
- Easy way to represent the report and output
- It supports SSL web pages “https://
- Detecting, handling and reporting redirected URLs
- Finally, Site Map

2.2.2.4 CSE Validator

This tool offers various options for checking pages including:

- HTML validating.
- CSS validating.
- Link checking.
- Spell checking.

Furthermore, it provides some useful options, for instance, accessibility, PHP syntax and search engine messages.

The tool will check the pages from different aspects like one-click SEO, spelling, accessibility and many other problems that appear within the tested page. One advantage of using this tool is that we do not need to send our web page over the internet and this will protect our privacy.

This tool will be used for tracking the detected errors/problems within an entire web page and see if these errors/problems impact the web page appearance or not and if these detected errors can help to improve the quality of these pages or not.

2.3 Reliability and Validity

In the following, we discuss the reliability and validity of our research approach.

**Reliability**: The used method for measuring the data depends on many factors
such as the installing environment where these tools will run (CPU, memory, OS, etc.) also, it depends on the connection quality with its traffic metrics since the link checker will require a network connection.

Since the tested webpage’s content changes from time to time, and tools being used could be updated continuously, therefore the produced result cannot be valid to use for other research projects as exact output. However, it can be used as a starting point for some studies as a valid results for some purposes.

Internal validity risks: The internal validity risks of this work are existing, and one of the highest risks is the standards which is used as an architecture for building these tools where some tools are using W3C standards as a base for evaluating pages but might be not all standards. The second risk is whether some unknown variables affect the result, such as testing environment and any programs running in background that affect the performance or even the traffic speed since these tools are working with online websites, meaning that these tools will send requests to check these websites and the site behavior could be changed comparing to the load and server responses.

Construct validity risks: Website standards and checklist used in Section 1.1 and other Sections both are widely used terms within web world, and we could find a different explanation from one resource to another.

2.4 Ethical Considerations

This work is based on some tools which implemented and approved by different organizations and most of these tools/applications are under free license and using some standards which are provided by W3C.

3 Implementation

In this chapter, the tools/applications, environment, and implementation will be explained.

The first step is to get a list of URLs for the web pages which will be used for our evaluation, i.e., as input for the evaluation tools.

We found that Alexa [10] offers a good overview of relevant web pages allowing us to extract the needed URLs as input for our assessment tools. Also, it offers to download an extensive list of websites according to their reputation and usage.

This makes it very easy to select a subset of URLs as input for our tools.
One available download is the top-1m.csv.

The file ¹ which includes the 1 million top web pages according to Alexa (rank and URL). Since we within the limitations of this project do not have the resources to assess all 1 million web pages, we extracted the first 200 URL's (rank 1 to 200) as input for our tools (see the appendix for more details).

3.1 Sitespeed.io

Sitespeed.io is a grunt project, based on task runner, making it easier to run and use locally.

Requirements for installing Sitespeed.io are:

- **Npm** – for running site-speed.io tool we need to set up the package manager npm which will help to install the package modules.

- **Node** – Node.js® is “a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js’ package ecosystem, npm, is the largest ecosystem of open source libraries in the world.” [15]

To install and use Site-speed.io, these requirements have to be installed.

Since Sitespeed.io allows us to pass a list of URLs this is the best way of running the tool, so the 200 URLs extracted from Alexa will be sent as an array to the tool once as a list, as shown in (Listing 1). After the tool finishes, the output will be saved automatically to the folder which has been declared in the JS file for later analysis.

¹ http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
module.exports=function(grunt){
  grunt.initConfig({
    sitespeedio: {
      default: {
        options: {
          urls: ['http://www.google.com', 'http://www.youtube.com', // Here all URLs should be written
          deep: 1,
          resultBaseDir: 'Results/Project/'
        }
      }
    }
  });
  grunt.registerTask('default', ['sitespeedio']);
  grunt.loadNpmTasks('grunt-sitespeedio');
}

Listing 3.1: Running Sitespeed.io on 200 URL’s

3.2 W3C Html Checker
W3C HTML Checker is a web service API available online for checking web pages and sending the result as a response to the request. The produced data will be in JSON format, and it includes all outputs which we want to see for each of the URL we send.

Requirements for using W3C HTML Checker are:
- Any IDE (Integrated Development Environment)
- Some external library for JSON (Json-sample-1.1 or java-json)

Since the used tool is an online API, the implementation will directly interact with the tool by sending requests and getting the data as a response from the tool.

We use the following functions of the API:
public void testSite()

This function will take each link and send to the API for validating, and the response will be some JSON data produced by the tool. It consists of a buffered reader for reading from the opened stream. It includes the parser for parsing the read stream.

public void printMessages()

This function will produce the whole required output sorted according to the intended implementation showing all common errors and info messages and printing in an appropriate way.

public static sortingBySite()

This function will show for each website the total number of validation errors and info messages.

private static Map<String, Integer> sortByValue(Map<String, Integer> unsortMap)

This function will sort the produced map, which consists of a pair of key and value, to sorted map.

public static <K, V> void printMap(Map<K, V> map)

This function will print the sorted map as a result, and the output is each type of error with the corresponding message likewise for info.

For running W3C HTML Checker on our selection of 200 websites, we wrote the following small program which sends each URL to the API and retrieves the result for later analysis.
public static void main(String[] args) {
    //Objects
    W3C w3c=new W3C();

    /**<
     * Testing the tools with any number of sites
     * @param site = site index from the file top.csv =200 web sites
     */
    for (int site = 0; site < 200; site++) {
        String url = SitesUrl.getSpecificUrl(site);
    }

    /**Printing messages*/
    w3c.printMessages();
    w3c.sortingBySites();

Listing 3.2: Running W3C Html Checker on 200 URL’s

(For the complete implementation please check the appendix)

3.3 Link Checker

This tool is an executable program with a user interface which can easily be used, and for our work, we pass the total number of URLs to the tool as a list. One of the requirements for running the tool is an internet connection, so the tool can check the link online and detect any broken or dead link.

The picture below shows the tool interface:

Figure 3.1: Xenu’s Link Sleuth Link Checker
### 3.4 Manual Testing

To see if there is any qualitative impact of the most common validation error/warning identified by the tool on web pages and checking whether the used tool is reliable for detecting errors/warning, we need to run the tool with an appropriate number of tests.

On the other hand, for achieving that, the testing environment will consist of single HTML pages which consist of several components including tags, elements, and their attributes.

The page is a normal HTML file with a common structure for building the web pages[16]. The general structure looks as follows:

```html
<!DOCTYPE html>
<html>
<head>
<title>This is a title</title>
</head>
<body>
<p>Hello world!</p>
</body>
</html>
```

**Element example:**

```html
<head>
<title>The Title</title>
</head>
```

**Attribute example:**

```html
<tag attribute1="value1" attribute2="value2">'content'</tag>
```

The tests will run against each element with some error and here the goal is to see if the tool will detect the error or not and then checking the web page
to see (manually) if any defects can be found. i.e., if it is worthwhile to fix all errors identified by the validation tool or if the browser corrects all errors anyway, this means there is no visible difference.

Example:

\( \text{<h1> this is heading 1 <h1> } \)

The expected result should be:

Closing tag is missing or something similar this

The page which will be used for testing looks like the screenshot below, which I believe is including most common attributes and elements and fulfill the requirement to test.

![HTML & CSS Example Page](image)

*Figure 3.2: HTML & CSS Example Page*
4 Results and Analysis

In this chapter, we present the results of the different experiments, and we discuss for each tool the analysis of the collected data based on the results.

Figure 4.1 shows the largest pages by size during the experiment, and this figure will help to extract a conclusion for best practice and some recommendations in the next chapter.
Figure 4.2: pages with worst score (sitespeed.io)

Figure 4.2 represent the pages with worst scores in descending order.

This analysis based on site-speed.io where the bad score is between 49 - 77

While 49 mean very bad and 77 mean bad.

<table>
<thead>
<tr>
<th>Site</th>
<th>Site Language</th>
<th>Site Link</th>
<th>Site</th>
<th>Site language</th>
<th>Site-Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Unknown</td>
<td><a href="http://www.t.co/">http://www.t.co/</a></td>
<td>16</td>
<td>Auto redirecting</td>
<td><a href="http://se.indeed.com/?r=us">http://se.indeed.com/?r=us</a></td>
</tr>
<tr>
<td>7</td>
<td>Not English</td>
<td><a href="http://www.youth.cn/">http://www.youth.cn/</a></td>
<td>19</td>
<td>English</td>
<td><a href="https://www.cnet.com/">https://www.cnet.com/</a></td>
</tr>
<tr>
<td>8</td>
<td>Not English</td>
<td><a href="https://www.pixnet.net/">https://www.pixnet.net/</a></td>
<td>20</td>
<td>Not English</td>
<td><a href="http://www.glob.com/">http://www.glob.com/</a></td>
</tr>
<tr>
<td>12</td>
<td>Not English</td>
<td><a href="http://www.csdn.net/">http://www.csdn.net/</a></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1 Sites not checked by sitespeed.io
Looking at this Table (4.1), we can see that most of the sites that were not checked were because of the site language not being or not supporting English.

From 23 sites which were not checked, 14 were not English sites.

<table>
<thead>
<tr>
<th>Color</th>
<th>From</th>
<th>To</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>49</td>
<td>77</td>
<td>78</td>
</tr>
<tr>
<td>Yellow</td>
<td>81</td>
<td>90</td>
<td>46</td>
</tr>
<tr>
<td>Green</td>
<td>91</td>
<td>99</td>
<td>53</td>
</tr>
</tbody>
</table>

Table 4.2: Total webpages by scores (Sitespeed.io)

Looking at the output generated by Site-speed.io we can see that the number of sites with red color (with bad scores) is much larger than the yellow (web pages with accepted scores) and green, but since we are using the best-ranked sites available today online, this result might not be true, or at least it cannot be trusted to some degree using only basic calculation as in this project.

The total tested websites were 200 URLs.

Number of sites Not checked by Sitespeed.io tool: **23** (Table 4.1)

Number of sites checked by Sitespeed.io tool: **177**

**Equation:** \( Y = P\% \times X \)
Where this equation calculates the percentage of \( x \)
That mean:

\[ Y \text{ is } P\% \text{ of } X \]

Where: \( Y \) is what Present of \( X \)

We have \( Y=78 \) site with red color (sites with bad scores)
And the total checked sites: \( X=177 \)
So, 78 is \( P\% \) of 177 where \( P = \frac{Y}{X} \times 100 \)
Hence, the result is 44.07%

We see that when the tools based on current best practices are used, the top 200 pages in the world get low scores.

<table>
<thead>
<tr>
<th>State</th>
<th>URLs</th>
<th>Percentage</th>
<th>State</th>
<th>URLs</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>21247 URLs</td>
<td>94.95%</td>
<td>Server Error</td>
<td>2 URLs</td>
<td>0.01%</td>
</tr>
<tr>
<td>Not Found</td>
<td>232 URLs</td>
<td>1.04%</td>
<td>Mail Host OK</td>
<td>10 URLs</td>
<td>0.04%</td>
</tr>
<tr>
<td>Timeout</td>
<td>551 URLs</td>
<td>2.46%</td>
<td>Error 521</td>
<td>2 URLs</td>
<td>0.01%</td>
</tr>
<tr>
<td>File Not Found</td>
<td>1 URLs</td>
<td>0.00%</td>
<td>Auth Required</td>
<td>5 URLs</td>
<td>0.02%</td>
</tr>
<tr>
<td>No Info To Return</td>
<td>13 URLs</td>
<td>0.06%</td>
<td>Connection Reset</td>
<td>3 URLs</td>
<td>0.01%</td>
</tr>
<tr>
<td>No Such Host</td>
<td>64 URLs</td>
<td>0.29%</td>
<td>No Object Data</td>
<td>5 URLs</td>
<td>0.02%</td>
</tr>
<tr>
<td>Cancelled /Timeout</td>
<td>10 URLs</td>
<td>0.04%</td>
<td>Temporarily Overloaded</td>
<td>7 URLs</td>
<td>0.03%</td>
</tr>
<tr>
<td>Forbidden Request</td>
<td>59 URLs</td>
<td>0.26%</td>
<td>The resource is no longer available</td>
<td>2 URLs</td>
<td>0.01%</td>
</tr>
<tr>
<td>Timed out waiting for gateway</td>
<td>15 URLs</td>
<td>0.07%</td>
<td>Method is not allowed</td>
<td>6 URLs</td>
<td>0.03%</td>
</tr>
<tr>
<td>Skip type</td>
<td>15 URLs</td>
<td>0.07%</td>
<td>Error response received from gateway</td>
<td>4 URLs</td>
<td>0.02%</td>
</tr>
<tr>
<td>Error 999</td>
<td>77 URL</td>
<td>0.34%</td>
<td>SSL certificate common name incorrect</td>
<td>1 URLs</td>
<td>0.00%</td>
</tr>
<tr>
<td>No Connection</td>
<td>44 URLs</td>
<td>0.20%</td>
<td>Unsupported media type</td>
<td>1 URLs</td>
<td>0.00%</td>
</tr>
<tr>
<td>Total</td>
<td>22376 URLs</td>
<td>Percentage</td>
<td></td>
<td></td>
<td>100.00%</td>
</tr>
</tbody>
</table>

Table 4.3: All Pages, by result type produced by Xenu Link
The result seen in this Table 4.3 showed the total number of web pages checked with Xenu Link Checker.

The total number of links was 22,376 URLs as we notice the number of health URLs was very good based on the total links with 94.95% as a percentage. On the other hand, the number of URLs with timeout was also noticeable with 551 URLs and 2.46% as well as the number of URLs with Not found was 232 URLs comparing to the total number of URLs.

<table>
<thead>
<tr>
<th>Type</th>
<th>No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of Errors</td>
<td>956</td>
</tr>
<tr>
<td>Total number of Info</td>
<td>177</td>
</tr>
<tr>
<td>Sum</td>
<td>1133</td>
</tr>
</tbody>
</table>

Table 4.4: Total number of errors and info by (Validator.nu)

The result from validator.nu experiments (Table 4.4) show the total number of detected errors and warning/info message in 200 Webpages where the tool examines those sites and demonstrate for each site the specific message and the type of these messages (see the Appendix 10.3 for the complete Tables). An observation is that the number of errors was very big compared to the total number of warning/info.

<table>
<thead>
<tr>
<th>Common Info messages</th>
<th>No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The “name” attribute is obsolete. Consider putting an “id” attribute on the nearest container instead</td>
<td>637</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Common Error messages</th>
<th>No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute “_sp” not allowed on element “a” at this point.</td>
<td>854</td>
</tr>
</tbody>
</table>

Table 4.5: Common errors and info messages by(Validator.nu)

Looking at the Table 4.5, where the common errors and warning messages represented, we notice that the most common error was “Attribute “_sp” not allowed on element “a” at this point.”, whereas the common info message was “The “name” attribute is obsolete. Consider putting an “id” attribute on the nearest container instead”
### Table 4.6 CSE HTML manual testing

<table>
<thead>
<tr>
<th>no</th>
<th>Test case</th>
<th>Error/warning</th>
<th>Detected</th>
<th>Appearance</th>
<th>type</th>
<th>behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;html lang=&quot;en&quot;&gt;</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Error</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>&lt;head &gt;</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Error</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>title2&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Both</td>
<td>Both</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>&lt;header&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Error</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>&lt;h1&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Error</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>&lt;div&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Error</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>&lt;tr&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Missing</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>&lt;form action=&quot;action_page.php&quot;&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Error</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>&lt;img src=&quot;/images&quot;</td>
<td>Y</td>
<td>Y</td>
<td>Error</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>&lt;h1&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Error</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>#eee; padding-bottom: 10010px;</td>
<td>Y</td>
<td>Y</td>
<td>Error</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

By looking at the results from the CSE tool (both Table 4.6 and Table 4.7).
We observe some unexpected behaviors:

1. In the case of error, some character could be used in a section or a paragraph, and for this characters, the tool will show that they are errors:

   \[ \text{Example} \]
   <h1 style="text-align:center; padding-bottom:20px ; color:black">HTML & CSS</h1>

   \[ \text{Fixed by} \]
   <h1 style="text-align:center; padding-bottom:20px ; color:black">HTML & CSS</h1>

2. An error detected when there was no space between two attributes:

   \[ \text{Example} \]
   <img src="images/linnaeus_university.jpg" alt="HTML5 Icon" class="imagediv">

   \[ \text{Fixed by} \]
   <img src="images/linnaeus_university.jpg" alt="HTML5 Icon" class="imagediv">

3. Checking some warnings and the tools behaviors:

   The number of warnings was bigger than errors but I would like to represent some of these warnings just to see how the tool check and work

   \[ \text{Example} \]
   \[ \text{html} \]

   This is the first line in any HTML document will throw a warning and the reason is because of the language.

   \[ \text{Fixed by} \]
4. Checking some href tags

The tool throws many errors in the following case

Example

```html
<li><a href="#">Link 1</a></li>
<li><a href="#">Link 2</a></li>
<li><a href="#">Link 3</a></li>
<li><a href="#">Link 4</a></li>
<li><a href="#">Link 5</a></li>
```

This will produce a lot of warning messages, and all these messages are because of the numbers 2,3,4, etc.

Fixed by

```html
<li><a href="#">Link one</a></li>
<li><a href="#">Link two</a></li>
<li><a href="#">Link three</a></li>
<li><a href="#">Link four</a></li>
<li><a href="#">Link five</a></li>
```
6 Discussion

By looking at the results from the previous chapter, we find that Sitespeed.io is checking the page size. Hence this a good notifies to say that the tool is showing some notification about the influenced of the page size. Similarly, we come back to the Table 4.1 we see that most web pages that were not checked were non-English web pages and this will address some question regarding this behavior, is there some issues with the tool? Can the website language affect the tool functionality?

On the other hand, when we are looking carefully at the result produced by Xenu Link Checker at least the number of Not found links, we discover they are 232 and this also leading for addressing some questions, where do the results will change in case of deep analyzing otherwise will be the same? Furthermore, checking the result produced by the Validator. Now we see a big number of errors and warnings (appendix 10.2), But the most common error was:

*Attribute “_sp” not allowed on element “a” at this point.* Whereas the common info message was:

*The “name” attribute is obsolete. Consider putting an “id” attribute on the nearest container instead.* Where the tool was not showing the exact reason for these two results.

In the end, the aim of using these tools is to indicate some errors or warnings within the tested page/website, which will enhance the quality of testing pages and show at least some concern about the weak parts detected during the test.

Of course, it is good to check some previous studies and compare this work to what others did but, unfortunately, studies within the same topic has been hard to find. One of the previous studies is WebMate, where the researcher was representing a tool which used for testing Web 2.0 applications. But, the reader can’t decide if this tool is the right choice because the tool is a pure black-box testing tool with no feedback from the server side. Also, the user needs to see the difference comparing to some other tools used for the same goal.
7 Conclusion

In this section we are representing our conclusions based on the results and outputs achieved so far from this work.

Our goal is to define the extracted results from evaluating a few common tools used for evaluating web pages and gives a brief summary of validation errors and warning with theirs affect and impact on website appearance. Also, we are describing the research questions and their answers.

RQ1. Do validation errors (and warnings) have an impact on the web page? If yes, what is an acceptable upper threshold?

As represented in Section (3. Implementation) the manual testing results answer this question.

As we can see the results from both Table (4.6 and 4.7) Checking the validation errors/warning impact on the web pages show how the validation errors/warnings can affect the web page’s appearance.

Of course, at the same time, the accepted upper threshold is:

Looking at the results in Figure 4.3 and Table 4.4 we see the scores for all web pages are as follow

- All web pages with worst scores are between 49-77 these web pages are counted as the lowest score.
- All web pages with scores between 81 and 99 are acceptable

For example, www.instragram.com has 97, and Google also has over 90.

RQ2. What are the most common validation errors?

1- URLs with time out represented 2.46% of the total test URLs.
2- Web pages not being checked because of the language 56.5%.
3- A missing space character and a quoted string.
4- Missing closing tag (i.e. > or <).
5- The following message:
   
   The “name” attribute is obsolete. Consider putting an “id” attribute on the nearest container instead.

6- This warning:
   
   Attribute “_sp” not allowed on element “a” at this point.
RQ3. Which are web quality tool recommendations worthwhile following?

Depending on the results which we find during this work, we think this question is hard to answer, and we should take the question in details since there are many commonalities between the tools we used for achieving this work, for instance, both Sitespeed.io and Valditor.nu, are checking web pages but each of them showing different outputs according to their implementation.

These tools attempting to check if a page is built by web standards Hence, we think that we can't define which tool is more reliable.

Therefore, we think it is difficult to answer this question, and it needs more advance work for testing and comparing these tools with various aspects such as the implementation, functionality and quality attributes.

RQ4. Which are the most common recommendations/scores for improving a web page?

*According to Sitespeed.io here are some recommendations which we can find:*

1) Doing as little as possible within the HEAD tags

   By following this recommendation, the browser can start rendering a page faster and easier (for instance in case of a lot of files in head tags the DNS will look for all those tags to load CSS/JS files)

2) Some JS files per page:

   The number of JS files affect the page speed

   Hence using a few large files is good practice in case of using HTTP 1,1

   And in the case of using HTTP 2,0, it is better with many small files from fewer domains.

3) Some numbers of CSS files per page:

   As JS file case it is better to use few number of big pages in case of using HTTP 1,1 and the opposite in case of using HTTP 2,0

4) Document weight:

   By keeping the document at an appropriate size, the browser will render it faster.

5) A number of domains:
Using a big number of domains mean a big number of DNS which will make the page slower

6) redirect per page:

When redirecting from one view to another, the page will be slower

*Based on the results from Figure 4.3 and Table 4.4*

The accepted scores are divided into the following categories

- **Green color:** where the scores from 49 or less up to 77 means the website score is accepted and nothing more needs to be done.
- **Yellow color:** where the scores from 81-90 means the website still at an acceptable level, but there is something that needs to be improved.
- **Red color:** where the scores from 91-99 means there are some serious issues that need to be fixed.
- **Blue color:** not related to the website score but, means some suggestions and general information for improving the site in general.

### 7.1 Future Work

Future research in the same area may be done at a larger scale for instance. In this work, the numbers of tools used for testing purposes were just a few. The focus was on general key points such as testing HTML or Links. Therefore, it may be necessary to investigate areas not covered by this work, such as checking the JavaScript code or file within specific web-pages or even the performance.

On the other hand, some studies could be done with different tools for evaluating web pages.

By comparing these tool behaviors to some developed methods and evaluating the results beside, with new research, it is useful to explore and investigate some questions which appeared during this work.

An example of such question is evaluating the different tool outputs of two tools for the same goal, making it clear during this work I found that some HTML errors detected by one tool but not with another in spite of both tools are for the same purpose.

So, checking the standards or checklist of these tools working according and
comparing will be interesting.

Similarly, the current work could be improved by implementing a tool for checking and comparing the output produced by two different tools for the same goal, testing and evaluating the website in term of architecture, followed standards, used frameworks and Languages, and then evaluating the results from the different perspectives.

9 References


10 Appendix

10.1 Validdator.nu implementation

package com;
/**
 * This class is implemented for fetching all popular sites
 * The class will read a big file of type CSV
 * It will fetch all websites addresses
 * @author imad Collin
 * @version 0
 */
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;

public class SitesUrl {

    /**ArrayList of sites
     * From CSV file and as element in AraayList
     * @return ArrayList of Sites
     */
    public static ArrayList<String> TopSites(){

        String csvFile = "~/Users/Imad Collin/Desktop/200.csv";
        BufferedReader br = null;
        String line = "";
        String str = "";
        String append = "www.";
        
        return;
    }
}
ArrayList<String> siteList = new ArrayList<String>();

try {

    br = new BufferedReader(new FileReader(csvFile));

    while ((line = br.readLine()) != null) {
        int index = line.indexOf(",");
        if (index >= 0)
            str = line.substring(index + 1, line.length());
        str = append.concat(str);
        siteList.add(str);
    }

} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

finally {
    if (br != null) {
        try {
            br.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

return siteList;
/** Get specific URL
 * This method will return any URL by passing the
 * index in ArrayList (0-100000)
 * @param urlIndex this equivalent to get(i)
 * @return the specific URL
 */

public static String getSpecificUrl(int urlIndex) {
    ArrayList<String> urlList = TopSites();
    String url = "";
    for (int i = 0; i <= urlList.size(); i++) {
        url = urlList.get(urlIndex);
    }
    return url;
}

10.2 Validator.nu result against two websites

************************** Info **************************
Using “windows-1252” instead of the declared encoding “iso-8859-1”. : 1
Legacy encoding “windows-1252” used. Documents should use UTF-8. : 1
The “button” role is unnecessary for element “button”. : 120

************************** Errors **************************
The “bgcolor” attribute on the “body” element is obsolete. Use CSS instead. : 1
Element “h2” is missing required attribute “role”. : 1
Internal encoding declaration “utf-8” disagrees with the actual encoding of the document (“windows-1252”). : 1
Bad value “handheld” for attribute “media” on element “link”. Deprecated media type “handheld”. For guidance, see the Media Types section in the current Media Queries specification. : 1
Element “a” is missing one or more of the following attributes: “aria-checked”, “role”. : 1
Element “div” not allowed as child of element “span” in this context. (Suppressing further errors from this subtree.) : 1

The “valign” attribute on the “tr” element is obsolete. Use CSS instead. : 1

The “clear” attribute on the “br” element is obsolete. Use CSS instead. : 1

Element “p” not allowed as child of element “ol” in this context. (Suppressing further errors from this subtree.) : 1

Attribute “width” not allowed on element “div” at this point. : 1

Attribute “href” not allowed on element “button” at this point. : 1

The “cellspacing” attribute on the “table” element is obsolete. Use CSS instead. : 1

Element “p” not allowed as child of element “span” in this context. (Suppressing further errors from this subtree.) : 1

The “cellpadding” attribute on the “table” element is obsolete. Use CSS instead. : 1

Attribute “name” not allowed on element “style” at this point. : 1

Bad value “” for attribute “id” on element “div”: An ID must not be the empty string. : 1

The “center” element is obsolete. Use CSS instead. : 1

The “width” attribute on the “td” element is obsolete. Use CSS instead. : 2

Element “div” not allowed as child of element “label” in this context. (Suppressing further errors from this subtree.) : 2

The “align” attribute on the “td” element is obsolete. Use CSS instead. : 2

The “nowrap” attribute on the “td” element is obsolete. Use CSS instead. : 2

Attribute “name” not allowed on element “script” at this point. : 4

Attribute “name” not allowed on element “link” at this point. : 4

Element “div” not allowed as child of element “a” in this context. (Suppressing further errors from this subtree.) : 2

Attribute “name” not allowed on element “script” at this point. : 4

Attribute “name” not allowed on element “link” at this point. : 4

Element “div” not allowed as child of element “a” in this context. (Suppressing further errors from this subtree.) : 7

Element “ul” not allowed as child of element “button” in this context. (Suppressing further errors from this subtree.) : 121

An element with “role=menuitem” must be contained in, or owned by, an element with “role=menubar” or “role=menu”. : 240

******************************************************************************

Common_Info_messages and high appearance is:The “button” role is unnecessary for element “button”. : 120

Common_Error_messages and high appearance is:An element with “role=menuitem” must be contained in, or owned by, an element with “role=menubar” or “role=menu”. : 240

******************************************************************************

Total number of Errors type:27

Total number of Info type:3
10.3 Table showing the total errors and info messages for each website

<table>
<thead>
<tr>
<th>SITE</th>
<th>MESSAGE TYPE AND NUMBER</th>
<th>MESSAGE TYPE AND NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://WWW.GOOGLE.COM">WWW.GOOGLE.COM</a></td>
<td>Info messages:2</td>
<td>Errors messages:18</td>
</tr>
<tr>
<td><a href="http://WWW.YOUTUBE.COM">WWW.YOUTUBE.COM</a></td>
<td>Info messages:122</td>
<td>Errors messages:401</td>
</tr>
<tr>
<td>WWW_FACEBOOK.COM</td>
<td>Info messages:126</td>
<td>Errors messages:430</td>
</tr>
<tr>
<td><a href="http://WWW.BAIDU.COM">WWW.BAIDU.COM</a></td>
<td>Info messages:126</td>
<td>Errors messages:436</td>
</tr>
<tr>
<td><a href="http://WWW.YAHOO.COM">WWW.YAHOO.COM</a></td>
<td>Info messages:126</td>
<td>Errors messages:544</td>
</tr>
<tr>
<td><a href="http://WWW.AMAZON.COM">WWW.AMAZON.COM</a></td>
<td>Info messages:126</td>
<td>Errors messages:551</td>
</tr>
<tr>
<td><a href="http://WWW.WIKIPEDIA.ORG">WWW.WIKIPEDIA.ORG</a></td>
<td>Info messages:148</td>
<td>Errors messages:566</td>
</tr>
<tr>
<td><a href="http://WWW.QQ.COM">WWW.QQ.COM</a></td>
<td>Info messages:148</td>
<td>Errors messages:567</td>
</tr>
<tr>
<td><a href="http://WWW.TWITTER.COM">WWW.TWITTER.COM</a></td>
<td>Info messages:150</td>
<td>Errors messages:574</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.CO.IN">WWW.GOOGLE.CO.IN</a></td>
<td>Info messages:152</td>
<td>Errors messages:595</td>
</tr>
<tr>
<td><a href="http://WWW.LIVE.COM">WWW.LIVE.COM</a></td>
<td>Info messages:152</td>
<td>Errors messages:619</td>
</tr>
<tr>
<td><a href="http://WWW.TAOBAO.COM">WWW.TAOBAO.COM</a></td>
<td>Info messages:154</td>
<td>Errors messages:662</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.CO.JP">WWW.GOOGLE.CO.JP</a></td>
<td>Info messages:155</td>
<td>Errors messages:682</td>
</tr>
<tr>
<td><a href="http://WWW.BING.COM">WWW.BING.COM</a></td>
<td>Info messages:161</td>
<td>Errors messages:757</td>
</tr>
<tr>
<td><a href="http://WWW.SINA.COM.CN">WWW.SINA.COM.CN</a></td>
<td>Info messages:161</td>
<td>Errors messages:758</td>
</tr>
<tr>
<td><a href="http://WWW.INSTAGRAM.COM">WWW.INSTAGRAM.COM</a></td>
<td>Info messages:161</td>
<td>Errors messages:759</td>
</tr>
<tr>
<td><a href="http://WWW.LINKEDIN.COM">WWW.LINKEDIN.COM</a></td>
<td>Info messages:161</td>
<td>Errors messages:763</td>
</tr>
<tr>
<td><a href="http://WWW.WEIBO.COM">WWW.WEIBO.COM</a></td>
<td>Info messages:164</td>
<td>Errors messages:764</td>
</tr>
<tr>
<td><a href="http://WWW.YAHOO.CO.JP">WWW.YAHOO.CO.JP</a></td>
<td>Info messages:164</td>
<td>Errors messages:764</td>
</tr>
<tr>
<td><a href="http://WWW.MSN.COM">WWW.MSN.COM</a></td>
<td>Info messages:165</td>
<td>Errors messages:777</td>
</tr>
<tr>
<td><a href="http://WWW.VK.COM">WWW.VK.COM</a></td>
<td>Info messages:166</td>
<td>Errors messages:788</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.DE">WWW.GOOGLE.DE</a></td>
<td>Info messages:168</td>
<td>Errors messages:808</td>
</tr>
<tr>
<td><a href="http://WWW.YANDEX.RU">WWW.YANDEX.RU</a></td>
<td>Info messages:172</td>
<td>Errors messages:826</td>
</tr>
<tr>
<td><a href="http://WWW.HAO123.COM">WWW.HAO123.COM</a></td>
<td>Info messages:173</td>
<td>Errors messages:907</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.CO.UK">WWW.GOOGLE.CO.UK</a></td>
<td>Info messages:175</td>
<td>Errors messages:929</td>
</tr>
<tr>
<td>Website</td>
<td>Info messages</td>
<td>Errors messages</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><a href="http://WWW.REDDIT.COM">WWW.REDDIT.COM</a></td>
<td>176</td>
<td>933</td>
</tr>
<tr>
<td><a href="http://WWW.EBAY.COM">WWW.EBAY.COM</a></td>
<td>182</td>
<td>1613</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.FR">WWW.GOOGLE.FR</a></td>
<td>184</td>
<td>1633</td>
</tr>
<tr>
<td><a href="http://WWW.T.CO">WWW.T.CO</a></td>
<td>184</td>
<td>1634</td>
</tr>
<tr>
<td><a href="http://WWW.TMALL.COM">WWW.TMALL.COM</a></td>
<td>186</td>
<td>1650</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.COM.BR">WWW.GOOGLE.COM.BR</a></td>
<td>188</td>
<td>1670</td>
</tr>
<tr>
<td><a href="http://WWW.PINTEREST.COM">WWW.PINTEREST.COM</a></td>
<td>189</td>
<td>1677</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.RU">WWW.GOOGLE.RU</a></td>
<td>190</td>
<td>1697</td>
</tr>
<tr>
<td><a href="http://WWW.AMAZON.CO.JP">WWW.AMAZON.CO.JP</a></td>
<td>194</td>
<td>1713</td>
</tr>
<tr>
<td><a href="http://WWW.MAIL.RU">WWW.MAIL.RU</a></td>
<td>357</td>
<td>1730</td>
</tr>
<tr>
<td><a href="http://WWW.SOHU.COM">WWW.SOHU.COM</a></td>
<td>358</td>
<td>1733</td>
</tr>
<tr>
<td><a href="http://WWW.560.CN">WWW.560.CN</a></td>
<td>358</td>
<td>1734</td>
</tr>
<tr>
<td><a href="http://WWW.NETFLIX.COM">WWW.NETFLIX.COM</a></td>
<td>361</td>
<td>1750</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.IT">WWW.GOOGLE.IT</a></td>
<td>363</td>
<td>1770</td>
</tr>
<tr>
<td><a href="http://WWW.ONCLICKADS.NET">WWW.ONCLICKADS.NET</a></td>
<td>363</td>
<td>1771</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.ES">WWW.GOOGLE.ES</a></td>
<td>365</td>
<td>1792</td>
</tr>
<tr>
<td><a href="http://WWW.MICROSOFT.COM">WWW.MICROSOFT.COM</a></td>
<td>453</td>
<td>2249</td>
</tr>
<tr>
<td><a href="http://WWW.GMW.CN">WWW.GMW.CN</a></td>
<td>453</td>
<td>2250</td>
</tr>
<tr>
<td><a href="http://WWW.PAYPAL.CN">WWW.PAYPAL.CN</a></td>
<td>462</td>
<td>2262</td>
</tr>
<tr>
<td><a href="http://WWW.WORDPRESS.COM">WWW.WORDPRESS.COM</a></td>
<td>465</td>
<td>2266</td>
</tr>
<tr>
<td><a href="http://WWW.TUMBLR.COM">WWW.TUMBLR.COM</a></td>
<td>465</td>
<td>2273</td>
</tr>
<tr>
<td><a href="http://WWW.BLOGSPOT.COM">WWW.BLOGSPOT.COM</a></td>
<td>474</td>
<td>2288</td>
</tr>
<tr>
<td><a href="http://WWW.IMGUR.COM">WWW.IMGUR.COM</a></td>
<td>476</td>
<td>2556</td>
</tr>
<tr>
<td><a href="http://WWW.CHINADAILY.COM.CN">WWW.CHINADAILY.COM.CN</a></td>
<td>476</td>
<td>2557</td>
</tr>
<tr>
<td><a href="http://WWW.NAVER.COM">WWW.NAVER.COM</a></td>
<td>476</td>
<td>2558</td>
</tr>
<tr>
<td><a href="http://WWW.STACKOVERFLOW.COM">WWW.STACKOVERFLOW.COM</a></td>
<td>477</td>
<td>2569</td>
</tr>
<tr>
<td><a href="http://WWW.ALIEXPRESS.COM">WWW.ALIEXPRESS.COM</a></td>
<td>480</td>
<td>2595</td>
</tr>
<tr>
<td><a href="http://WWW.GITHUB.COM">WWW.GITHUB.COM</a></td>
<td>482</td>
<td>2615</td>
</tr>
<tr>
<td><a href="http://WWW.APPLE.COM">WWW.APPLE.COM</a></td>
<td>488</td>
<td>2629</td>
</tr>
<tr>
<td><a href="http://WWW.XVIDEOS.COM">WWW.XVIDEOS.COM</a></td>
<td>488</td>
<td>2629</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.COM.MX">WWW.GOOGLE.COM.MX</a></td>
<td>490</td>
<td>2649</td>
</tr>
<tr>
<td><a href="http://WWW.OK.RU">WWW.OK.RU</a></td>
<td>543</td>
<td>2857</td>
</tr>
<tr>
<td><a href="http://WWW.163.COM">WWW.163.COM</a></td>
<td>543</td>
<td>2858</td>
</tr>
<tr>
<td><a href="http://WWW.IMDB.COM">WWW.IMDB.COM</a></td>
<td>574</td>
<td>3004</td>
</tr>
<tr>
<td>Website</td>
<td>Info messages</td>
<td>Errors messages</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><a href="http://WWW.PORNHUB.COM">WWW.PORNHUB.COM</a></td>
<td>574</td>
<td>3045</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.CO.KR">WWW.GOOGLE.CO.KR</a></td>
<td>575</td>
<td>3065</td>
</tr>
<tr>
<td><a href="http://WWW.JD.COM">WWW.JD.COM</a></td>
<td>576</td>
<td>3167</td>
</tr>
<tr>
<td><a href="http://WWW.FC2.COM">WWW.FC2.COM</a></td>
<td>576</td>
<td>3178</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.COM.HK">WWW.GOOGLE.COM.HK</a></td>
<td>577</td>
<td>3199</td>
</tr>
<tr>
<td><a href="http://WWW.WHATSAPP.COM">WWW.WHATSAPP.COM</a></td>
<td>577</td>
<td>3209</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.CA">WWW.GOOGLE.CA</a></td>
<td>579</td>
<td>3230</td>
</tr>
<tr>
<td><a href="http://WWW.BLOGGER.COM">WWW.BLOGGER.COM</a></td>
<td>588</td>
<td>3245</td>
</tr>
<tr>
<td><a href="http://WWW.XHAMSTER.COM">WWW.XHAMSTER.COM</a></td>
<td>588</td>
<td>3412</td>
</tr>
<tr>
<td><a href="http://WWW.AMAZON.IN">WWW.AMAZON.IN</a></td>
<td>593</td>
<td>3429</td>
</tr>
<tr>
<td><a href="http://WWW.YOUTH.CN">WWW.YOUTH.CN</a></td>
<td>619</td>
<td>3492</td>
</tr>
<tr>
<td><a href="http://WWW.OFFICE.COM">WWW.OFFICE.COM</a></td>
<td>621</td>
<td>3587</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.COM.TR">WWW.GOOGLE.COM.TR</a></td>
<td>623</td>
<td>3607</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.CO.ID">WWW.GOOGLE.CO.ID</a></td>
<td>625</td>
<td>3628</td>
</tr>
<tr>
<td><a href="http://WWW.RAKUTEN.CO.JP">WWW.RAKUTEN.CO.JP</a></td>
<td>626</td>
<td>3631</td>
</tr>
<tr>
<td><a href="http://WWW.CRAIGSLIST.ORG">WWW.CRAIGSLIST.ORG</a></td>
<td>627</td>
<td>3634</td>
</tr>
<tr>
<td><a href="http://WWW.AMAZON.DE">WWW.AMAZON.DE</a></td>
<td>631</td>
<td>3650</td>
</tr>
<tr>
<td><a href="http://WWW.NICVIDEO.JP">WWW.NICVIDEO.JP</a></td>
<td>631</td>
<td>3651</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.PL">WWW.GOOGLE.PL</a></td>
<td>632</td>
<td>3671</td>
</tr>
<tr>
<td><a href="http://WWW.YOUKU.COM">WWW.YOUKU.COM</a></td>
<td>632</td>
<td>3672</td>
</tr>
<tr>
<td><a href="http://WWW.BILIBILI.COM">WWW.BILIBILI.COM</a></td>
<td>632</td>
<td>3691</td>
</tr>
<tr>
<td><a href="http://WWW.BONGACAMS.COM">WWW.BONGACAMS.COM</a></td>
<td>638</td>
<td>4239</td>
</tr>
<tr>
<td><a href="http://WWW.PIXNET.NET">WWW.PIXNET.NET</a></td>
<td>642</td>
<td>4341</td>
</tr>
<tr>
<td><a href="http://WWW.BOOKING.COM">WWW.BOOKING.COM</a></td>
<td>646</td>
<td>4424</td>
</tr>
<tr>
<td><a href="http://WWW.ALIBABA.COM">WWW.ALIBABA.COM</a></td>
<td>647</td>
<td>4451</td>
</tr>
<tr>
<td><a href="http://WWW.DROPBOX.COM">WWW.DROPBOX.COM</a></td>
<td>652</td>
<td>4458</td>
</tr>
<tr>
<td><a href="http://WWW.SOSO.COM">WWW.SOSO.COM</a></td>
<td>652</td>
<td>4487</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.COM.AU">WWW.GOOGLE.COM.AU</a></td>
<td>655</td>
<td>4505</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLE.COM.TW">WWW.GOOGLE.COM.TW</a></td>
<td>656</td>
<td>4525</td>
</tr>
<tr>
<td><a href="http://WWW.ALIPAY.COM">WWW.ALIPAY.COM</a></td>
<td>657</td>
<td>4538</td>
</tr>
<tr>
<td><a href="http://WWW.OUTBRAND.COM">WWW.OUTBRAND.COM</a></td>
<td>657</td>
<td>4574</td>
</tr>
<tr>
<td><a href="http://WWW.GOOGLEUSERCONTENT.COM">WWW.GOOGLEUSERCONTENT.COM</a></td>
<td>657</td>
<td>4575</td>
</tr>
<tr>
<td><a href="http://WWW.POPADS.NET">WWW.POPADS.NET</a></td>
<td>658</td>
<td>4577</td>
</tr>
<tr>
<td>URL</td>
<td>Info messages</td>
<td>Errors messages</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><a href="http://www.amazon.co.uk">www.amazon.co.uk</a></td>
<td>662</td>
<td>4593</td>
</tr>
<tr>
<td><a href="http://www.diply.com">www.diply.com</a></td>
<td>667</td>
<td>4596</td>
</tr>
<tr>
<td><a href="http://www.microsoftonline.com">www.microsoftonline.com</a></td>
<td>667</td>
<td>4597</td>
</tr>
<tr>
<td><a href="http://www.zhihu.com">www.zhihu.com</a></td>
<td>668</td>
<td>4606</td>
</tr>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>678</td>
<td>4629</td>
</tr>
<tr>
<td><a href="http://www.xinhuanet.com">www.xinhuanet.com</a></td>
<td>678</td>
<td>4630</td>
</tr>
<tr>
<td><a href="http://www.cococ.com">www.cococ.com</a></td>
<td>678</td>
<td>4645</td>
</tr>
<tr>
<td><a href="http://www.chase.com">www.chase.com</a></td>
<td>682</td>
<td>4655</td>
</tr>
<tr>
<td><a href="http://www.quora.com">www.quora.com</a></td>
<td>685</td>
<td>4693</td>
</tr>
<tr>
<td><a href="http://www.amazon.cn">www.amazon.cn</a></td>
<td>698</td>
<td>4852</td>
</tr>
<tr>
<td><a href="http://www.myway.com">www.myway.com</a></td>
<td>698</td>
<td>4853</td>
</tr>
<tr>
<td><a href="http://www.twitch.tv">www.twitch.tv</a></td>
<td>700</td>
<td>4860</td>
</tr>
<tr>
<td><a href="http://www.ask.com">www.ask.com</a></td>
<td>702</td>
<td>4882</td>
</tr>
<tr>
<td><a href="http://www.isanalyze.com">www.isanalyze.com</a></td>
<td>702</td>
<td>4883</td>
</tr>
<tr>
<td><a href="http://www.bbc.co.uk">www.bbc.co.uk</a></td>
<td>743</td>
<td>4970</td>
</tr>
<tr>
<td><a href="http://www.wikia.com">www.wikia.com</a></td>
<td>748</td>
<td>5137</td>
</tr>
<tr>
<td><a href="http://www.adobe.com">www.adobe.com</a></td>
<td>759</td>
<td>5200</td>
</tr>
<tr>
<td><a href="http://www.cntv.cn">www.cntv.cn</a></td>
<td>759</td>
<td>5201</td>
</tr>
<tr>
<td><a href="http://www.google.com.eg">www.google.com.eg</a></td>
<td>760</td>
<td>5222</td>
</tr>
<tr>
<td><a href="http://www.google.com.pk">www.google.com.pk</a></td>
<td>762</td>
<td>5243</td>
</tr>
<tr>
<td><a href="http://www.sogou.com">www.sogou.com</a></td>
<td>762</td>
<td>5272</td>
</tr>
<tr>
<td><a href="http://www.google.co.th">www.google.co.th</a></td>
<td>764</td>
<td>5294</td>
</tr>
<tr>
<td><a href="http://www.detail.tmall.com">www.detail.tmall.com</a></td>
<td>764</td>
<td>5295</td>
</tr>
<tr>
<td><a href="http://www.google.com.sa">www.google.com.sa</a></td>
<td>765</td>
<td>5316</td>
</tr>
<tr>
<td><a href="http://www.nytimes.com">www.nytimes.com</a></td>
<td>785</td>
<td>5347</td>
</tr>
<tr>
<td><a href="http://www.bbc.com">www.bbc.com</a></td>
<td>794</td>
<td>5349</td>
</tr>
<tr>
<td><a href="http://www.adf.ly">www.adf.ly</a></td>
<td>806</td>
<td>5365</td>
</tr>
<tr>
<td><a href="http://www.google.com.ar">www.google.com.ar</a></td>
<td>808</td>
<td>5385</td>
</tr>
<tr>
<td><a href="http://www.txxx.com">www.txxx.com</a></td>
<td>1054</td>
<td>6032</td>
</tr>
<tr>
<td><a href="http://www.ettoday.net">www.ettoday.net</a></td>
<td>1099</td>
<td>6110</td>
</tr>
<tr>
<td><a href="http://www.flipkart.com">www.flipkart.com</a></td>
<td>1099</td>
<td>6112</td>
</tr>
<tr>
<td><a href="http://www.google.nl">www.google.nl</a></td>
<td>1101</td>
<td>6132</td>
</tr>
<tr>
<td><a href="http://www.360.com">www.360.com</a></td>
<td>1101</td>
<td>6133</td>
</tr>
<tr>
<td><a href="http://www.ebay.co.uk">www.ebay.co.uk</a></td>
<td>1105</td>
<td>6747</td>
</tr>
<tr>
<td>Website</td>
<td>Info messages</td>
<td>Errors messages</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><a href="http://www.daum.net">www.daum.net</a></td>
<td>1105</td>
<td>6748</td>
</tr>
<tr>
<td><a href="http://www.livedoor.jp">www.livedoor.jp</a></td>
<td>1105</td>
<td>6749</td>
</tr>
<tr>
<td><a href="http://www.salesforce.com">www.salesforce.com</a></td>
<td>1316</td>
<td>6847</td>
</tr>
<tr>
<td><a href="http://www.china.com">www.china.com</a></td>
<td>1316</td>
<td>6848</td>
</tr>
<tr>
<td><a href="http://www.haosou.com">www.haosou.com</a></td>
<td>1317</td>
<td>6859</td>
</tr>
<tr>
<td><a href="http://www.dailymail.co.uk">www.dailymail.co.uk</a></td>
<td>1349</td>
<td>7005</td>
</tr>
<tr>
<td><a href="http://www.ifeng.com">www.ifeng.com</a></td>
<td>1349</td>
<td>7006</td>
</tr>
<tr>
<td><a href="http://www.ebay.de">www.ebay.de</a></td>
<td>1353</td>
<td>7618</td>
</tr>
<tr>
<td><a href="http://www.xnxx.com">www.xnxx.com</a></td>
<td>1354</td>
<td>7627</td>
</tr>
<tr>
<td><a href="http://www.dailymotion.com">www.dailymotion.com</a></td>
<td>1358</td>
<td>7802</td>
</tr>
<tr>
<td><a href="http://www.bankofamerica.com">www.bankofamerica.com</a></td>
<td>1462</td>
<td>7844</td>
</tr>
<tr>
<td><a href="http://www.indiatimes.com">www.indiatimes.com</a></td>
<td>1569</td>
<td>7952</td>
</tr>
<tr>
<td><a href="http://www.walmart.com">www.walmart.com</a></td>
<td>1571</td>
<td>8380</td>
</tr>
<tr>
<td><a href="http://www.godaddy.com">www.godaddy.com</a></td>
<td>1574</td>
<td>8869</td>
</tr>
<tr>
<td><a href="http://www.amazonaws.com">www.amazonaws.com</a></td>
<td>1574</td>
<td>8870</td>
</tr>
<tr>
<td><a href="http://www.soundcloud.com">www.soundcloud.com</a></td>
<td>1574</td>
<td>8875</td>
</tr>
<tr>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td>1575</td>
<td>9208</td>
</tr>
<tr>
<td><a href="http://www.buzzfeed.com">www.buzzfeed.com</a></td>
<td>1615</td>
<td>9276</td>
</tr>
<tr>
<td><a href="http://www.terrACLicks.com">www.terrACLicks.com</a></td>
<td>1615</td>
<td>9278</td>
</tr>
<tr>
<td><a href="http://www.etsy.com">www.etsy.com</a></td>
<td>1634</td>
<td>9441</td>
</tr>
<tr>
<td><a href="http://www.wellsfargo.com">www.wellsfargo.com</a></td>
<td>1646</td>
<td>9471</td>
</tr>
<tr>
<td><a href="http://www.iqiyi.com">www.iqiyi.com</a></td>
<td>1646</td>
<td>9472</td>
</tr>
<tr>
<td><a href="http://www.indeed.com">www.indeed.com</a></td>
<td>1646</td>
<td>9496</td>
</tr>
<tr>
<td><a href="http://www.zillow.com">www.zillow.com</a></td>
<td>1646</td>
<td>9500</td>
</tr>
<tr>
<td><a href="http://www.china.com.cn">www.china.com.cn</a></td>
<td>1646</td>
<td>9501</td>
</tr>
<tr>
<td><a href="http://www.vimeo.com">www.vimeo.com</a></td>
<td>1651</td>
<td>9526</td>
</tr>
<tr>
<td><a href="http://www.cnet.com">www.cnet.com</a></td>
<td>1655</td>
<td>9749</td>
</tr>
<tr>
<td><a href="http://www.uptodown.com">www.uptodown.com</a></td>
<td>1793</td>
<td>9804</td>
</tr>
<tr>
<td><a href="http://www.so.com">www.so.com</a></td>
<td>1794</td>
<td>9815</td>
</tr>
<tr>
<td><a href="http://www.thepiratebay.org">www.thepiratebay.org</a></td>
<td>1794</td>
<td>9816</td>
</tr>
<tr>
<td><a href="http://www.google.co.ve">www.google.co.ve</a></td>
<td>1796</td>
<td>9836</td>
</tr>
<tr>
<td><a href="http://www.google.co.za">www.google.co.za</a></td>
<td>1798</td>
<td>9857</td>
</tr>
<tr>
<td><a href="http://www.globo.com">www.globo.com</a></td>
<td>1798</td>
<td>9858</td>
</tr>
<tr>
<td><a href="http://www.stackexchange.com">www.stackexchange.com</a></td>
<td>1799</td>
<td>9858</td>
</tr>
<tr>
<td>Website</td>
<td>Info messages</td>
<td>Errors messages</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
<td>1806</td>
<td>10105</td>
</tr>
<tr>
<td><a href="http://www.washingtonpost.com">www.washingtonpost.com</a></td>
<td>1813</td>
<td>10290</td>
</tr>
<tr>
<td><a href="http://www.tripadvisor.com">www.tripadvisor.com</a></td>
<td>1817</td>
<td>10515</td>
</tr>
<tr>
<td><a href="http://www.tudou.com">www.tudou.com</a></td>
<td>1817</td>
<td>10516</td>
</tr>
<tr>
<td><a href="http://www.espn.com">www.espn.com</a></td>
<td>2091</td>
<td>10890</td>
</tr>
<tr>
<td><a href="http://www.douyu.com">www.douyu.com</a></td>
<td>2092</td>
<td>10925</td>
</tr>
<tr>
<td><a href="http://www.aol.com">www.aol.com</a></td>
<td>2092</td>
<td>10933</td>
</tr>
<tr>
<td><a href="http://www.directrev.com">www.directrev.com</a></td>
<td>2092</td>
<td>10934</td>
</tr>
<tr>
<td><a href="http://www.ameblo.jp">www.ameblo.jp</a></td>
<td>2092</td>
<td>10935</td>
</tr>
<tr>
<td><a href="http://www.avito.ru">www.avito.ru</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.google.com.ua">www.google.com.ua</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.force.com">www.force.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.uol.com.br">www.uol.com.br</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.google.com.br">www.google.com.br</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.yelp.com">www.yelp.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.feedly.com">www.feedly.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.dmm.co.jp">www.dmm.co.jp</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.taboola.com">www.taboola.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.onlinesbi.com">www.onlinesbi.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.pixiv.net">www.pixiv.net</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.mediafire.com">www.mediafire.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.ikea.com">www.ikea.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.gfycat.com">www.gfycat.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.google.com.vn">www.google.com.vn</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.google.com.co">www.google.com.co</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.redtube.com">www.redtube.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.hclips.com">www.hclips.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.naver.jp">www.naver.jp</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.weather.com">www.weather.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.goo.ne.jp">www.goo.ne.jp</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.steampowered.com">www.steampowered.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.google.com.sg">www.google.com.sg</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.google.com.ng">www.google.com.ng</a></td>
<td>2092</td>
<td>10938</td>
</tr>
<tr>
<td><a href="http://www.target.com">www.target.com</a></td>
<td>2092</td>
<td>10938</td>
</tr>
</tbody>
</table>
### 10.4 Correct internal URLs, by MIME type (Xenu Link Sleuth)

<table>
<thead>
<tr>
<th>MIME TYPE</th>
<th>COUNT URLs</th>
<th>% COUNT</th>
<th>Σ SIZE</th>
<th>% SIZE</th>
<th>MIN SIZE</th>
<th>MAX SIZE</th>
<th>Ø SIZE</th>
<th>Ø SIZE</th>
<th>Ø TIME</th>
<th>MIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXT/PLAIN</td>
<td>23</td>
<td>0.32%</td>
<td>5097</td>
<td>0.00%</td>
<td>0 Bytes</td>
<td>4740</td>
<td>221</td>
<td>(0)</td>
<td>(0)</td>
<td></td>
</tr>
<tr>
<td>TEXT/HTML</td>
<td>6364</td>
<td>89.58%</td>
<td>82322</td>
<td>98.49%</td>
<td>0 Bytes</td>
<td>48682</td>
<td>1293</td>
<td>9 Bytes</td>
<td>5 Bytes</td>
<td></td>
</tr>
<tr>
<td>IMAGE/X-ICON</td>
<td>40</td>
<td>0.56%</td>
<td>50808</td>
<td>0.01%</td>
<td>0 Bytes</td>
<td>25214</td>
<td>1270</td>
<td>1 Bytes</td>
<td>1 Bytes</td>
<td></td>
</tr>
<tr>
<td>IMAGE/PNG</td>
<td>126</td>
<td>1.77%</td>
<td>26387</td>
<td>0.32%</td>
<td>251 Bytes</td>
<td>47507</td>
<td>2094</td>
<td>2 Bytes</td>
<td>20 Bytes</td>
<td></td>
</tr>
<tr>
<td>TEXT/CSS</td>
<td>31</td>
<td>0.44%</td>
<td>20493</td>
<td>0.25%</td>
<td>935 Bytes</td>
<td>56348</td>
<td>6610</td>
<td>8 Bytes</td>
<td>64 Bytes</td>
<td></td>
</tr>
<tr>
<td>IMAGE/SVG+XML</td>
<td>6</td>
<td>0.08%</td>
<td>16882</td>
<td>0.00%</td>
<td>1443 Bytes</td>
<td>6420</td>
<td>2813</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td></td>
</tr>
<tr>
<td>APPLICATION/X-JAVASCRIPT</td>
<td>30</td>
<td>0.42%</td>
<td>89217</td>
<td>0.11%</td>
<td>0 Bytes</td>
<td>28765</td>
<td>2973</td>
<td>9 Bytes</td>
<td>9 Bytes</td>
<td></td>
</tr>
<tr>
<td>APPLICATION/JSON</td>
<td>2</td>
<td>0.03%</td>
<td>0 Bytes</td>
<td>0.00%</td>
<td>0 Bytes</td>
<td>0</td>
<td>0 Bytes</td>
<td>0 Bytes</td>
<td>0 Bytes</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>URLs</td>
<td>Bytes</td>
<td>%</td>
<td>Bytes</td>
<td>%</td>
<td>Bytes</td>
<td>%</td>
<td>Bytes</td>
<td>%</td>
<td>Bytes</td>
</tr>
<tr>
<td>---------------------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>TEXT/XML</td>
<td>6</td>
<td>1280</td>
<td>0.08</td>
<td>0</td>
<td>0</td>
<td>213</td>
<td>0</td>
<td>696</td>
<td>0</td>
<td>34145</td>
</tr>
<tr>
<td>APPLICATION/JAVASCRIPT</td>
<td>27</td>
<td>10292</td>
<td>0.38</td>
<td>0</td>
<td>0</td>
<td>3812</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>3184</td>
</tr>
<tr>
<td>TEXT/JAVASCRIPT</td>
<td>2</td>
<td>0</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>APPLICATION/XML</td>
<td>6</td>
<td>1064</td>
<td>0.08</td>
<td>0</td>
<td>0</td>
<td>177</td>
<td>0</td>
<td>559</td>
<td>0</td>
<td>29521</td>
</tr>
<tr>
<td>IMAGE/GIF</td>
<td>8</td>
<td>30330</td>
<td>0.11</td>
<td>0</td>
<td>0</td>
<td>3791</td>
<td>0</td>
<td>29521</td>
<td>0</td>
<td>3791</td>
</tr>
<tr>
<td>APPLICATION/BINARY</td>
<td>1</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IMAGE/VND.MICROSOFT.ICON</td>
<td>3</td>
<td>35698</td>
<td>0.04</td>
<td>0</td>
<td>0</td>
<td>1189</td>
<td>0</td>
<td>4838</td>
<td>0</td>
<td>1189</td>
</tr>
<tr>
<td>APPLICATION/HTML+XML</td>
<td>1</td>
<td>12554</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>1255</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1255</td>
</tr>
<tr>
<td>IMAGE/JPEG</td>
<td>61</td>
<td>18344</td>
<td>0.86</td>
<td>0</td>
<td>0</td>
<td>3007</td>
<td>0</td>
<td>3864</td>
<td>0</td>
<td>3864</td>
</tr>
<tr>
<td>APPLICATION/PDF</td>
<td>5</td>
<td>37425</td>
<td>0.07</td>
<td>0</td>
<td>0</td>
<td>7485</td>
<td>0</td>
<td>1056</td>
<td>0</td>
<td>1056</td>
</tr>
<tr>
<td>APPLICATION/RESOURCE+XML</td>
<td>1</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>APPLICATION/READ+XML</td>
<td>1</td>
<td>1803</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>1803</td>
<td>0</td>
<td>1803</td>
<td>0</td>
<td>1803</td>
</tr>
<tr>
<td>TOTAL</td>
<td>7104</td>
<td>83584</td>
<td>100.0</td>
<td>816</td>
<td>100.0</td>
<td>251</td>
<td>0.0%</td>
<td>83584</td>
<td>100.0</td>
<td>83584</td>
</tr>
</tbody>
</table>

10.5 Total Error Detected by CSE Tool

1. `<h1 style="text-align:center; padding-bottom:20px ; color:black">HTML&CSS</h1>`

"&CSS" is an invalid character reference. This may be because the '& character was not escaped as "&amp;". To use a literal ampersand it must be encoded as "&amp;" (even in URLs) because '& is an escape character in HTML/XHTML.

2. `<img src="images/linnaeus_university.jpg" alt="HTML5 Icon class="imagDiv">`
Is there a missing space character? A quoted string (like an attribute value) cannot be immediately followed by text. It must be followed by a space or by the end of the tag.

fixed:

1. `<h1 style="text-align:center; padding-bottom:20px ; color:black">HTML&CSS</h1>`

   Solved: & removed.

2. `<img src="images/linnaeus_university.jpg" alt="HTML5 Icon " class="imagediv"`

   Solved: Space provided.

10.6 Warning Detected by CSE Tool

1. `<img src="images/linnaeus_university.jpg" alt="HTML5 Icon " class="imagediv"`  
   Possibly misspelled words (79, 60 unique): adipisci (1x), aliquam (1x), aliquid (1x), acrie(1x),AnotherParagraph (1x),asperiores (1x), beetae (1x), consequuntur (1x), deserunt (1x), distinctio (1x), Dolore (1x), doloremque (1x), dolores (1x), dolorum (1x), cum (1x), excepturi (1x), explicabo (1x), Halk (1x), Habil (1x), illo (1x), Ipsa (1x), iuquae (1x), laborum (1x), laudantium (1x), magni (1x), modi (1x), nisi (1x), Odit (1x), officiiis (1x), parietur (1x), quia (1x), qui (1x), repellant (1x), repudiandae (1x), Rerum (1x), Saepe (1x), sapiente (1x), sed (1x), sequi (1x), Souiri (1x), tempora (1x), temporiibus (1x), tenetur (1x), totam (1x), vel (1x), vero (1x), voluptatum (1x), esse (2x), facere (2x), inventore (2x). List limited to first 50 unique words. 161 total words checked (0 in comments)

2. `<html>`

   The natural primary language is not specified. It is highly recommended that the "lang" and/or "xml:lang" (for XHTML) attributes be used with the "html" element to specify the primary language. For example, add the attribute `lang="en"` for English or `lang="fr"` for French. Specifying the language assists braille translation software, speech synthesizers, translation software, and has other benefits.

3. `input[type=submit]:hover {`  
   Many mobile devices use touch screens making ":hover" problematic, cumbersome, or impossible to use. Depending on the browser, ":hover" might never match, or it might match for brief moments, or until the user touches another element. Ensure that content is never only accessible or only visible by hovering. This message is displayed only once.

5. `Links`  
   `<li><a href="#">Link 1</a></li>`  
   `<li><a href="#">Link 2</a></li>`  
   `<li><a href="#">Link 3</a></li>`  
   `<li><a href="#">Link 4</a></li>`  
   `<li><a href="#">Link 5</a></li>`

   Link text is important for SEO and should probably be more descriptive than simply "Link 1". Furthermore, poor link text may also cause accessibility issues.
The Main implementation using Java

```java
package com;
/**
 * This class is implemented for fetching all popular sites
 * The class will read a big file of type CSV
 * It will fetch all websites addresses
 * @author imad Collin
 * @version 0
 */
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
public class SitesUrl {
    /**
     * ArrayList of sites
     * From CSV file and as element in ArrayList
     * @return ArrayList of Sites
     */
    public static ArrayList<String> TopSites() {
        String csvFile = "//Users/Imad Collin/Desktop/Praktik/200.csv";
        BufferedReader br = null;
        String line = "";
        String str = "";
        String append = "www.";
        ArrayList<String> siteList = new ArrayList<String>();
        try {
            br = new BufferedReader(new FileReader(csvFile));
            while ((line = br.readLine()) != null) {
                int index = line.indexOf(",");
                if (index >= 0)
                    str = line.substring(index + 1, line.length());
                siteList.add(str);
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (br != null) {
                try {
                    // Code
```
br.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

return siteList;
}

/**
 * This method will return any URL by passing the index in ArrayList (0-100000)
 * @param urlIndex this equivalent to get(i)
 * @return the specific URL
 */
public static String getSpecificUrl(int urlIndex){
    ArrayList<String>urlList=TopSites();
    String url="";
    for (int i=0;i<=urlList.size();i++){
        url=urlList.get(urlIndex);
    }
    return url;
}

package com;
/**
 * This class will be used for testing purpose
 * The class includes the JsonMessages Objects
 * @author Imad Collin
 * @version 1.0
 */
public class JsonMessage {
    private String message;
    private String subType;
    private String type;

    public String getMessage ()
    {
        return message;
    }

    public void setMessage (String message)
    {
        this.message = message;
    }

    public String getSubType ()
    {
        return subType;
    }
}
public void setSubType (String subType)
{
    this.subType = subType;
}

public String getType ()
{
    return type;
}

public void setType (String type)
{
    this.type = type;
}

@Override
public String toString()
{
    return "ClassPojo [message = "+message+", subType = "+subType+", type = "+type+"]";
}

package com;
/**
 * This class is implemented for W3C API
 * The class will read a big file and fetch all websites addresses
 * By using the tool a number of messages will produced
 * The implementation include Message sorting as well
 * @author imad Collin
 * @version 0.1
 */
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.ParseException;


public class W3C {
    public static int total;
    public static int Errors = 0;
    public static int info = 0;

    public static JSONArray sites = new JSONArray();
    public static ArrayList<String> listOfErrors = new ArrayList<String>();
    public static ArrayList<String> listOfInfo = new ArrayList<String>();
    public static Map<String, Integer> MapOfErrors = new HashMap<String, Integer>();
    public static Map<String, Integer> MapOfInfo = new HashMap<String, Integer>();
    public static ArrayList<String> counterforurl = new ArrayList<String>();

    public void testSites(String siteurl) {

        try {
            //Send the request to the API
            URL url = new URL("https://validator.w3.org/nu/?doc=https%3A%2F%2F" + siteurl + "&%2F%2Fout=json");

            BufferedReader br = new BufferedReader(new InputStreamReader(url.openStream(), "utf-8"));

            String strTemp = "";
            strTemp = br.readLine();

            JSONParser parser = new JSONParser();

            try {
                Object obj = parser.parse(strTemp);
                JSONObject jsonObject = (JSONObject) obj;

                // Getting Messages from Json Data
                JSONArray msg = (JSONArray) jsonObject.get("messages");

                for (int i = 0; i < msg.size(); i++) {
                    JSONObject specific_msg = (JSONObject) msg.get(i);
                }
        }
    }
}
String mesg = (String) specific_msg.get("message");

//Counting Errors and Info messages
String type = (String) specific_msg.get("type");
if (type.contains("error")) {
    // Errors++;
    listOfErrors.add(mesg);
}
if (type.contains("info")) {
    // info++;
    listOfInfo.add(mesg);
}
System.out.println("Type of the message:" + type + "\tMessage:" + mesg);

} catch (IOException e) {
    e.printStackTrace();
}
} catch (ParseException e) {
    e.printStackTrace();
}
counterforurl.add(siteurl + "\t\tInfo messages:" + listOfInfo.size() + "\t\tErrors messages:" + listOfErrors.size() + "\n");

public void printMessages() {
    String common_info_messages = "";
    String common_error_messages = "";
    for (String term : listOfInfo) {
        int score = 0;
        if (MapOfInfo.containsKey(term)) {
            score = MapOfInfo.get(term);
        }
        MapOfInfo.put(term, score + 1);
    }
    for (String term : listOfErrors) {
        int score = 0;
        if (MapOfErrors.containsKey(term)) {
            score = MapOfErrors.get(term);
        }
        MapOfErrors.put(term, score + 1);
    }
    Map<String, Integer> InfoSort = sortByValue(MapOfInfo);
Map<String, Integer> ErrorSort =
sortByValue(MapOfErrors);

System.out.println("\n*************************** Info
***************************");
   for (Map.Entry<String, Integer> entry :
InfoSort.entrySet()) {
   System.out.println(entry.getKey() + " : " +
   entry.getValue());
   common_info_messages=entry.getKey() + " : " +
   entry.getValue();
   }
System.out.println("\n*************************** Errors
***************************");
   for (Map.Entry<String, Integer> entry :
ErrorSort.entrySet()) {
   System.out.println(entry.getKey() + " : " +
   entry.getValue());
   common_error_messages=entry.getKey() + " : " +
   entry.getValue();
   }
System.out.println("\n****************************************
*****");
System.out.println("Common_Info_messages and high appearance is:"+common_info_messages);
System.out.println("Common_Error_messages and high appearance is:"+common_error_messages);

System.out.println("\n****************************************
*****");
   info=InfoSort.size();
   Errors=ErrorSort.size();
   System.out.println("Total number of Errors type:"+Errors);
   System.out.println("Total number of Info type:"+info);
}
public void sortingBySites(){
System.out.println("\n****************************************
*****");
for(int i=0;i<counterforurl.size();i++){
   System.out.println(counterforurl.get(i));
}
private static Map<String, Integer>
sortByValue(Map<String, Integer> unsortMap) {
}
// Convert Map to List of Map
List<
Map.Entry<String, Integer> > list = new
LinkedHashSet<
Map.Entry<String, Integer> > (unsortMap.entrySet());
/**
* Sorting the List using comparable
*/
Collections.sort(list, new
Comparator<
Map.Entry<String, Integer> > () {
public int compare(Map.Entry<String, Integer> o1,
Map.Entry<String, Integer> o2) {
return
o1.getValue().compareTo(o2.getValue());
}
});

// LinkedHashMap
Map<String, Integer> sortedMap = new
LinkedHashMap<String, Integer>();
for (Map.Entry<String, Integer> entry : list) {
sortedMap.put(entry.getKey(), entry.getValue());
}
return sortedMap;
}
public static <K, V> void printMap(Map<K, V> map) {
for (Map.Entry<K, V> entry : map.entrySet()) {
System.out.println("Key : " + entry.getKey() + " Value : " + entry.getValue());
}
}
public class URLs_Conversion {
    public static int count = 0;
    public static void main(String[] args) {
        TopSites();
    }
    public static ArrayList<String> TopSites() {
        String csvFile = "'/Users/Imad Collin/Desktop/200.csv";
        ArrayList<String> siteList = new ArrayList<String>();
        try {
            BufferedReader br = null;
            String line = "";
            String str = "";
            String append = "'http://www.";
            while ((line = br.readLine()) != null) {
                int index = line.indexOf("", "");
                if (index >= 0) {
                    str = line.substring(index + 1, line.length());
                    str = append.concat(str).concat("", ""); // System.out.println(str);
                    count++;
                    //siteList.add(str);
                }
            }
            System.out.println("number of sites:" + count);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (br != null) {
                try {
                    br.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
return siteList;
}