What do you look for in a web page - how do you judge - what is important - what brings you back...

minimal
organized
colors
font
easy to navigate - menus back to home
interesting - style but easy to use

graphics
emphasize good first/home pg
loading speed esp 1st page
style fit page
intuitive Go back content
date
owner
Evaluation list:

Below is a list of some of the things that should be considered in evaluating sites.

- Web design can be said to have three parts: Information design, interaction design and presentation design. Were all done effectively?
- The designer should think about the message, the audience (including their goals and the back ground of information they bring to the site), purpose, structure including look and navigation. Did the designer of this site think about these things?
- Does site convey its message, does it do it effectively, is scope and purpose appropriate?
- Does the site have a good idea of the audience they want to attract and is site appropriate to that audience?
- Does site use resources (i.e. the Internet) well?
- Is the information the audience needs readily available? One of my pet peeves - is there an address and phone number readily available?
- Is the site easy to navigate? Is it well organized? Are their navigation tools like indexes, overview or maps? Is it easy to retrieve your steps or get to something? Is it easy to get lost?
- What user enters the site via a link to a page, is it easy to get to the home page?
- Does site load quickly?
- If site does not load quickly does it give alternatives that allow you to start exploring before all of the graphics are loaded?
- Is there content? Is there something worth a revisit - information, fun to explore etc? Does it meet needs of the audience?
- Is there style? Things to look at?
- Is general layout, arrangement of information effective?
- Is it too crowded, too basic, too confusing?
- Is "white space" used effectively?
- Is underlining used and if so it is easy to differentiate between underlining and links?
- Is graphic style consistent enough to give the site "a look"?
- Is color used effectively? What about font, graphics, animation, sound? Do they add to or distract from the message?
- Is there too much that is too cute and has been used too often?
- Does background help or overwhelms the site?
- Is there advertising - is it for the company or advertising from other sources to get a fee site?
- Is there a list of irrelevant stuff like icons that do not things, banner, awards, counters, animations which may be meaningless to the point you are trying to make?
- Is what you are looking at old or current - has it been updated recently etc?
- Are all of the links active?
- If site is static, is that appropriate? Should there be interactive information like reading for more information or ordering something
- Is there too much text - is it avoidable - is their medium awareness?
- If people surf with their graphics off is their enough information or text alternatives?
- If appropriate, are there good links for more information? How are the links handled? How does the user handle returning to the site?
February 29, 2016

I would like you to look through these examples. We went over the Selector examples in class.
A list and explanation of CSS selectors.

<table>
<thead>
<tr>
<th>Selector</th>
<th>Example</th>
<th>Example description</th>
<th>CSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>.class</td>
<td>.intro</td>
<td>Selects all elements with class=&quot;intro&quot;</td>
<td>1</td>
</tr>
<tr>
<td>#id</td>
<td>#firstname</td>
<td>Selects the element with id=&quot;firstname&quot;</td>
<td>1</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>Selects all elements</td>
<td>2</td>
</tr>
<tr>
<td>element</td>
<td>p</td>
<td>Selects all &lt;p&gt; elements</td>
<td>1</td>
</tr>
<tr>
<td>element+element</td>
<td>div, p</td>
<td>Selects all &lt;div&gt; elements and all &lt;p&gt; elements</td>
<td>1</td>
</tr>
<tr>
<td>element element</td>
<td>div p</td>
<td>Selects all &lt;p&gt; elements inside &lt;div&gt; elements</td>
<td>1</td>
</tr>
<tr>
<td>element+element</td>
<td>div &gt; p</td>
<td>Selects all &lt;p&gt; elements where the parent is a &lt;div&gt; element</td>
<td>2</td>
</tr>
<tr>
<td>element+element</td>
<td>div + p</td>
<td>Selects all &lt;p&gt; elements that are placed immediately after a &lt;div&gt; element</td>
<td>2</td>
</tr>
<tr>
<td>element+element2</td>
<td>p = ul</td>
<td>Selects every &lt;ul&gt; element that are preceded by a &lt;p&gt; element</td>
<td>3</td>
</tr>
<tr>
<td>[attribute]</td>
<td>[target]</td>
<td>Selects all elements with a target attribute</td>
<td>2</td>
</tr>
<tr>
<td>[attribute=value]</td>
<td>[target=blank]</td>
<td>Selects all elements with target=&quot;blank&quot;</td>
<td>2</td>
</tr>
<tr>
<td>[attribute=value]</td>
<td>[title=flower]</td>
<td>Selects all elements with a title attribute containing the word &quot;flower&quot;</td>
<td>2</td>
</tr>
<tr>
<td>[attribute=value]</td>
<td>[lang=en]</td>
<td>Selects all elements with a lang attribute value starting with &quot;en&quot;</td>
<td>2</td>
</tr>
<tr>
<td>[attribute=value]</td>
<td>[href=HTTPS]</td>
<td>Selects every &lt;a&gt; element whose href attribute value begins with &quot;HTTPS&quot;</td>
<td>3</td>
</tr>
</tbody>
</table>

Use the CSS Reference tool to demonstrate the different selectors.
Here the things I wanted to try. I commented out all but one so I could test.
This is a stand alone header

This is a header in a division

This is a paragraph in a division under a section

This is a paragraph under a division under a section

This is a paragraph after a division.

Paragraph within a section

- div, p makes all font red within a paragraph or division - selects all divs
- div > p impacts all paragraphs within divisions
- div + p means all paragraphs where the div is a parent
- p ~ ul means all ul elements that are preceded by a parent

w3school reference
This worked for div > h1 and also for p.

---

This is a stand alone header

**This is a header in a division**

This is a paragraph in a division below a header.
This worked for div > li and for div > p.

This is a header in a division

This is a paragraph under a division under a section.

This is a stand alone div

This is a stand alone paragraph

This is a stand alone header
This is a section
This is a division under a section
This is a paragraph under a division under a section
This is a stand alone div
This is a stand alone paragraph

This is a stand alone header

This is a header in a division

This is a paragraph in a division below a header
This is a paragraph after a division
Paragraph within a section
This is a paragraph.
- div, p makes all font red within a paragraph or division. - selects all paragraphs within divisions
- div > p selects all paragraphs where the div is a parent
- p means all paragraphs immediately after a division
- ul means all ul elements that are preceded by a paragraph

w3schools reference
This is a header in a division

This is a paragraph in a division below a header

Paragraph within a section

This is a paragraph.

- div > p selects all paragraphs where the div is a parent
- div + p means all paragraphs immediately after a division
- p ~ ul means all ul elements that are preceded by a paragraph
The document tree - Parent

A parent is an element that is directly above and connected to an element in the document tree. In the diagram below, the 'div' is a parent to the 'hr'.

```
body
  div Parent
    h1 Child
```

Other Max Design articles and presentations
Associated with: maxdesigngroup.org
The document tree - Descendant

A descendant refers to any element that is connected but lower down in the document tree, no matter how many levels lower. In the diagram below, all elements that are connected below the value element are descendants of that value.

```
body
  ↓
  div
    ↓
    h1  p  p  hr
    ↓
    div
      ↓
      b  b  b  b
```

Other Mat Design articles and presentations
Associated with matdesigngroup.org
Selectutorial: Should you use ID or class?

There is a lot of debate about whether you should use classes or IDs. Your decision should be based on the following:

1. Repeated use within a document
   - Classes can be used as many times as needed within a document.
   - IDs can only be applied once within a document.
   So, if you need to use the same specific selector more than once, classes are a better choice.

2. Combining class selectors
   You can use multiple classes to style an HTML element but you can only use one ID when styling an HTML element. This means that class selectors have a wider range of applications. An example of combining classes would be:

   ```html
   <p class="highlight indent">
     .highlight { font-weight: bold; }
     .indent { padding-left: 2em; }
   </p>
   ```

3. IDs have higher specificity than classes
   There may be times when a declaration conflicts with another declaration. These conflicts are resolved using the "cascading rule." In simple terms, if a class selector and ID selector were to be in conflict, the ID selector would be chosen.

Other CSS Design resources and documentation
Associated with https://www.test.com
Selectutorial
CSS selectors

Should you use ID or class?

There is a lot of debate about whether you should use classes or IDs. Your decision should be based on the following:

1. Repeated use within a document
   - Classes can be used as many times as needed within a document
   - IDs can only be applied once within a document

   So, if you need to use the same specific selector more than once, classes are a better choice.

2. Combining class selectors
   You can use multiple classes to style an HTML element, but you can only use one ID when styling an HTML element. This means that class selectors have a wider range of applications. An example of combining classes would be:

   ```html
   <p class="highlight indent">  
     .highlight { font-weight: bold; }  
     .indent { padding-left: 2em; }
   </p>
   ```

3. IDs have higher specificity than classes
   There may be times when a declaration conflicts with another declaration. These conflicts are resolved using the cascade rule. In simple terms, if a class selector and ID selector were to be in conflict, the ID selector would be chosen.

Other Web Design articles and presentations
Associated with https://designorganization.org
Class selectors

Introduction

While type selectors target every instance of an element, class selectors can be used to select any HTML element that has a class attribute, regardless of their position in the document tree.

For example, if you want to target the first paragraph and first list items on a page to make them stand out, you could mark up the page in the following way:

```html
<body>
  <p class="big">This is some content</p>
  <p>Or this is some text</p>
  <ul>
    <li class="big">List item 1</li>
    <li class="big">List item 2</li>
    <li class="small" style="list-style:none;">No list items here</li>
  </ul>
</body>
```

The tree diagram would be:

```
body
  ⍟ big
    ⍋ a
  ⍋ ul
    ⍋ li
      ⍋ big
      ⍋ li
      ⍋ li
      ⍋ li
```

The rule used could then be:

```
.big { font-size: 110%; font-weight: bold; }
```
ID selectors

ID selectors are similar to class selectors. They can be used to select any HTML element that has an ID attribute, regardless of their position in the document tree. Examples of ID selectors:

```css
#navigation { width: 12em; color: #333; }
#dirNavigation { width: 12em; color: #333; }
```

The major difference is that IDs can only be applied once per page, while classes can be used as many times as needed.

ID selectors are well supported across standards-compliant browsers.

Further information

- [W3C CSS ID Selector](http://www.w3.org/TR/CSS1#syntax)
- [W3C CSS Specification](http://www.w3.org/TR/CSS1)
- [W3C CSS ID Selector Support Chart](http://www.w3.org/TR/CSS1-support)
- [W3C CSS1 Checkpoint](http://www.w3.org/TR/CSS1-checkpoint)

Descendant selectors »

Other WAI Design articles and presentations
Associated with: [www standardsgroup.org](http://www standardsgroup.org)
About This Course:

Read Me First
Introduction to CIS122/44
Syllabus
College Policies

Site Resources:

Notes and Handouts
definitions (1/31)
definitions (2/1)
definitions (2/3/16)
definitions (3/14)

HTML5:

HTML5 sample code
CSS sample code
CSS3 sample code
JavaScript samples
Audio/Video samples under XHTTP

XHTML:

XHTML sample code
CSS sample code
JavaScript
Previews
Smartboard notes
Audio notes (usually accompany Smartboard)

HTML:

HTML sample code
Images sample code
CSS sample code
JavaScript
Previews

Weekly Schedule:

Weekly schedule - CIS122/44
Assignment Summary

Links:

Information & Tutorials
Some useful books

Week #6

Week of February 22nd

CSS reference - interesting examples
CSS reference - one of many from about.com
Internet CSS positioning experiment
Reference - recommended by previous class

Wide variety of references

Another wide variety of references

Please check out these sites:

CSS selectors
Selector tutorial on CSS

CSS tests to go with Selector tutorial and w3schools.

Selector example

div, span, p

div > span

div, span, p

CSS tests to go with Selector tutorial

Uploading to the Web: In my notes I talk about closing bristol-mass edge. Things have changed and you should now use close-bristol-edge. Please note that we now prefix close-bristol-edge instead of close-bristol-mass edge.

We will look at how to close this week so check the notes on the Smartboard and the accompanying audio as well as the notes below.

To load things up on the Internet, we use WinSCP. Please read the WinSCP and closed-bristol edge (note that in my examples I use close-bristol-mass edge as the host, we are now using close-bristol-edge as the host although the old one is still supported).

Here is information about using closed-bristol and WINSCP at RCC:

Talks more specifically about portfolio. We will be looking at making a portfolio later in the semester.

If you want to download WinSCP at home, you can find it at this address:

WinSCP

Some articles I would like you to read on paths since there seems to be a lot of questions:

Article on relative and absolute paths

We will talk a little about what makes a good site: A starting point for thought and discussion

Assignments to turn in:

CSS Assignment

First project now assigned. First web project! Note that a working website is required for this course. You cannot pass the course if you do not have a working website written using HTML5 and CSS. I would like the project by next April (or try for April 11th)

We will look at images, mapping, forms, iframes and possibly CSS. You need to check back!

Assignments to turn in:

In-class exercise that must also be done and submitted by the people at home. I want you to work on circle, rectangle and polygon. I want you to download an image - you can use something from my site or find something yourself. Once you have downloaded and niced it up you need to work with it. I want you to map the image. The mapping should send control to an area within your page so we can keep the tech.

I also want you to experiment with frames. Do the following: 1) a static frame with one row on the top and one row on the bottom and the bottom one divided into 3 columns. 2) a row at the top divided into two frames and a row at the bottom divided into two frames with the first frame taking 60% of the area and the second frame taking 40% of the area.
Validation of CSS both in an HTML document or alone.
The W3C CSS Validation Service

W3C CSS Validator results for styleexperiment.html (CSS level 3)

Congratulations! No Error Found.

This document validates as CSS level 3!

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<IMG border="0" vspace="0" style="text-align: right; float: right;"
     src="http://www.w3.org/Icons/valid-xhtml10.png"
     alt="Valid XHTML 1.0!"/>
```

```
<IMG border="0" vspace="0" style="text-align: left; float: left;"
     src="http://www.w3.org/Icons/valid-css3.png"
     alt="Valid CSS!"/>
```

(close the img tag with > instead of /> if using HTML <= 4.01)

The W3C validators rely on community support for hosting and development. Donate and help us build better tools for a better web.

If you like, you can download a copy of this image to keep in your local web directory, and change the XHTML fragment above to reference your local image rather than the one on this server.
WinSCP site - can download for use at home.

WinSCP 5.8.1 beta released

The most important changes/additions are:

- Support for ECDSA and Ed25519 keys
- Dual code signed binaries with both SHA-1 and SHA-256.
- Improvements to code/script generation

Eager to learn more about the new features? Follow us on Facebook, Twitter, Google+ or LinkedIn.

[Go to Download page]  [Complete list of changes]
This is a stand alone header

This is a header in a division

This is a paragraph in a division below a header
This is a paragraph after a division.
Paragraph within a section

This is a paragraph.

- *dir*: makes all font red within a paragraph or division. - selects all divisions and paragraphs.
- *dir p*: makes all paragraphs within a division.
- *dir > p*: selects all paragraphs where the div is a parent.
- *dir > p*: makes all paragraphs immediately after a division.
- *p*: means all text elements that are preceded by a paragraph.

file name 3.html

username@host
This is the menu:

1. first
2. second
3. third
4. fourth
5. fifth

This is the text:

There is a lot of information here. There is a lot of information here.
This is the menu.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.
This is the information I am providing.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.
This is the information I am providing.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.

See code on next page.
Changes that led to the white space on the previous slide.

See below.
This is the information I am providing.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.

Experimenting...
Essentially the class had a variety of things we tried.
Experimented with change the px we had to auto.
Experiment continuing...
This is the information I am providing.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.
Putting in a header.

This is the banner at the top in the header.

This is the information I am providing.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.
We also experimented with position - fixed and absolute were tested with the same results.
This is the banner at the top in the header.

This is the information I am providing.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.
This is the banner at the top in the header.

This is the information I am providing.

There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here. There is a lot of information here.
We looked at these.
We are starting to look at JavaScript. JavaScript provides more power to websites because you can add decisions and loops. More user interactive...
```
<html>
<script type="text/javascript">
    document.write("Hello world!");
    alert("Hello World!");
</script>
</html>
```

Write function (II) means ( ) literal.
HELLO WORLD!

Hello World!
HTML5 with html and script.

This had a popup.
Multiply two numbers taken in through prompts

The answer is 20.

```html
<html>
<head>
<title>Multiply</title>
<meta charset="utf-8">
</head>
<body>
<h1>Multiply two numbers taken in through prompts</h1>
<script type="text/javascript">
var ans = 0;
var firstnum = 0;
var secondnum = 0;
firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);
ans = firstnum * secondnum;
document.write(ans);
document.write("<br>");
document.write("The answer is ", ans);
document.write("<br>");
</script>
</body>
</html>
```
Steps:

- Write the HTML code with JavaScript.
- Save with a .html extension.
- Load into your browser.
Multiply two numbers taken in through prompts

?2
The answer is 72
```html
document.write("The dataset is ", n);
document.write("\n");
</body>
</html>
```
The second number I entered 5.
+ can mean add or concatenate

If I enter 4 and 5, then 45 is the concatenated version
and 9 is the result of adding.
The answer is 45. The answer is 9.

```html
<!DOCTYPE html>
<html>
<title>Addition</title>
<meta charset="utf-8">
<body>
<script type="text/javascript">
var ans = 0;
var otherans = 0;
var firstnum = 0;
var secondnum = 0;
firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);
ans = firstnum + secondnum;
document.write("The answer is ", ans);
otherans = parseFloat(firstnum) + parseFloat(secondnum);
document.write("The answer is ", otherans);
</script>
</body>
</html>
```

The `<br>` will move me down a line.
This shows a variety of ways to get the add to happen

```html
<!DOCTYPE html>
<html>
<head>
<title>Addition</title>
<meta charset="utf-8">
</head>
<body>
<h1>This shows a variety of ways to get the add to happen</h1>
<script type="text/javascript">
var ans = 0;
var otherans = 0;
var intans = 0;
var firstnum = 0;
var seconndum = 0;
firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);
ans = firstnum + secondnum;
document.write("The answer is ", ans);
otherans = parseFloat(firstnum) + parseFloat(secondnum);
document.write("This is using parseFloat<br>");
intans = parseInt(firstnum) + parseInt(secondnum);
document.write("This is using parseInt<br>");
plusans = +(firstnum) + +(secondnum);
document.write("<br>This is using Number<br>");
numberans = Number(firstnum) + Number(secondnum);
document.write("<br>This is using Number<br>");
</script>
</body>
</html>
```
This shows a variety of ways to get the add to happen

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Addition</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p>This shows a variety of ways to get the add to happen</p>
    <script type="text/javascript">
      var ans = 0;
      var otherans = 0;
      var intans = 0;
      var firstnum = 0;
      var secondnum = 12;
      firstnum = prompt("Enter the first number",0);
      secondnum = prompt("Enter the second number",0);
      ans = firstnum + secondnum;
      document.write("The answer is ", ans);
      otherans = parseInt(firstnum) + parseFloat(secondnum);
      document.write("<br>This is using parseFloat<br>");
      document.write("The answer is ", otherans);
      intans = parseInt(firstnum) + parseInt(secondnum);
      document.write("<br>This is using parseInt<br>");
      document.write("The answer is ", intans);
      plusans = +(firstnum) + +(secondnum);
      document.write("<br>This is using +<br>");
      document.write("The answer is ", plusans);
      numberans = Number(firstnum) + Number(secondnum);
      document.write("<br>This is using Number<br>");
      document.write("The answer is ", numberans);
    </script>
  </body>
</html>
```
The answer is 12

```html
<!DOCTYPE html>
<html>
<head>
<title>Decision</title>
<meta charset="utf-8">
</head>
<body>
<script type="text/javascript">
var ans = 0;
var firstnum = 0;
var secondnum = 0;
var whattodo;
firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);
whattodo = prompt("Enter * or /","");
if (whattodo == "+")
{
  ans = firstnum + secondnum;
}
else
{
  ans = firstnum / secondnum;
}
document.write("the answer is ", ans);
</script>
</body>
</html>
```
The answer is 12

```html
<!DOCTYPE html>
<html>
<head>
<title>Decision</title>
<meta charset="utf-8">
</head>
<body>
<script type="text/javascript">
var ans = 0;
var firstnum = 0;
var secondnum = 0;
var whattodo;
firstnum = prompt("Enter the first number",0);
secondnum = prompt("Enter the second number",0);
whattodo = prompt("Enter * or /","");
if (whattodo == "+")
{
    ans = firstnum + secondnum;
}
else
{
    ans = firstnum / secondnum;
}
document.write("The answer is ", ans);
</script>
</body>
</html>
```
If I put in a + what will happen?
It ends up doing a divide because + is not an * so the NO side is taken and a divide is done.
The if statement uses the == to compare whattodo to ".
I have now added a second if to check for /.
```html
<!DOCTYPE html>
<html>
<head>
  <title>Decision</title>
  <meta charset='utf-8'>
</head>
<body type="text/javascript">
  var ans = 0;
  var firstnum = 0;
  var seconndnum = 0;
  var whattodo;
  firstnum = prompt("Enter the first number",0);
  seconndnum = prompt("Enter the second number",0);
  whattodo = prompt("Enter * or /","");
  if (whattodo == "+")
  {
    ans = firstnum + seconndnum;
  }
  else
  {
    if (whattodo == "/")
    {
      ans = firstnum / seconndnum;
    }
    else
    {
      ans = "try again";
    }
  }
  document.write("The answer is ", ans);
</script>
</body>
</html>
```