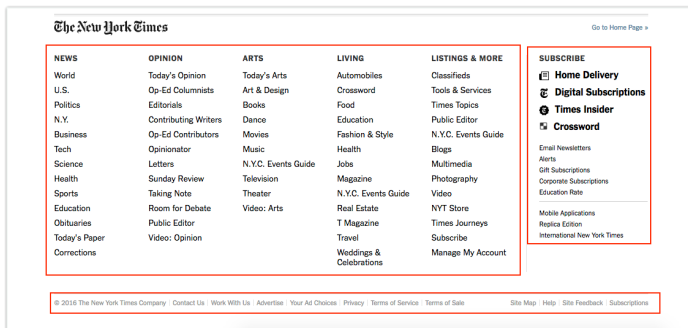# Comm 244
# Week 3

# Navigation

# Navigation

Websites need a formalized system of links to allow users to navigate the site

# Navigation

Many larger websites have multiple forms of navigation

For example, look at the New York Times website

# Adding Navigation

Navigation elements are really just **lists** of links

# Adding Navigation

Navigation elements are really just **lists** of links

All of the pages in our Recipe Book need a basic navigation structure

The Recipe Book website has 5 main pages:
- Home (index.html)
- Recipes
- Tips
- Contributors
- About

# Adding Navigation

The 5 pages will form the basis of our navigation

- Home (index.html)
- Recipes
- Tips
- Contributors
- About

Eventually we will also use a logo to link to the home page, but for now, just link the title of the website.

# HTML Validation

# What is a Validator

A program that looks at your HTML...

# What is a Validator

A program that looks at your HTML...

and then tells you if it is **valid**

# Why You Care

HTML is designed to be forgiving of errors.

Not all browsers interpret errors the same.

The only way to ensure that your site will look the *closest* to the same in all browsers is to use valid HTML.

# Why You Care

A validator will tell you when something in your code is wrong, and help you write better code.

Writing good code now will help ensure that your site will still work in the future.

# World Wide Web Consortium (W3C)

The W3C is an international group that develops web standards

They helped develop specifications for HTML, XHTML, CSS (among others)

# Validators and Doctypes

Remember the HTML <!DOCTYPE>
Declaration we talked about?

```
                                    template.html
1   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"¬
2       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">¬
3   ¬
4   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">¬
5   <head>¬
6       <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>¬
7   ¬
8       <title>XHTML Strict Template</title>¬
9   ¬
10  </head>¬
11  ¬
12  <body>¬
13  ¬
14  ¬
```

---

# Validators and Doctypes

This tells the Validator how it should
test the document

```
                                    template.html
1   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"¬
2       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">¬
3   ¬
4   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">¬
5   <head>¬
6       <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>¬
7   ¬
8       <title>XHTML Strict Template</title>¬
9   ¬
10  </head>¬
11  ¬
12  <body>¬
13  ¬
14  ¬
```

---

# Validators and Doctypes

This document will be XHTML1 Strict

```
                                    template.html
1   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"¬
2       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">¬
3   ¬
4   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">¬
5   <head>¬
6       <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>¬
7   ¬
8       <title>XHTML Strict Template</title>¬
9   ¬
10  </head>¬
11  ¬
12  <body>¬
13  ¬
14  ¬
```

## Validators and Doctypes

HTML5 simplified the doctype substantially:

`<!DOCTYPE html>`

It's a whole different approach. Easier.

## W3C Validator

http://validator.w3.org/

And also,

http://validator.nu/

## Cascading Style Sheets ( CSS )

# CSS Syntax

---

# Attaching Your Styles

There are three ways to attach CSS an HTML document

  External Style Sheet

  Internal Style Sheet

  Inline Styles

---

# Attaching Your Styles

**External Style Sheet**

```
<head>
  <link rel="stylesheet" href="mystyle.css"
  media="screen" />
</head>
```

# Attaching Your Styles

**Internal Style Sheet**

```
<head>
  <style>
    p { color: red; }
  </style>
</head>
```

*Also called Embedded*

# Attaching Your Styles

**Inline Styles**

```
<p style="color:red">A red paragraph.</p>
```

# Attaching Your Styles

**Inline Styles**

```
<p style="color:red">A red paragraph.</p>
```

Inline styles are bad practice. Forget you saw them.

# CSS Shorthand

---

There are a lot of CSS border properties

**For borders on all sides:**
- border
- border-color
- border-style
- border-width

**For top borders:**
- border-top
- border-top-color
- border-top-style
- border-top-width

**For bottom borders:**
- border-bottom
- border-bottom-color
- border-bottom-style
- border-bottom-width

**For right borders:**
- border-right
- border-right-color
- border-right-style
- border-right-width

**For left borders:**
- border-left
- border-left-color
- border-left-style
- border-left-width

---

Most of them are redundant

If all of your borders are the same on every side, you can just use the border property

border
border-color
border-style
border-width

border-top
border-top-color
border-top-style
border-top-width

border-right
border-right-color
border-right-style
border-right-width

border-bottom
border-bottom-color
border-bottom-style
border-bottom-width

border-left
border-left-color
border-left-style
border-left-width

**Border property**

If all of your borders are the same on every side,
you can just use the border property

**Syntax:**

```
p {
    border: width style color;
}
```

**Example:**

```
p {
    border: 1px solid #333;
}
```

*This defines a one pixel wide solid gray border for all paragraphs*

---

Example:

You want a three pixel dashed green border
But, you only want it on the left and bottom sides

```
border
border-color
border-style
border-width

border-top
border-top-color
border-top-style
border-top-width

border-right
border-right-color
border-right-style
border-right-width
```

```
border-bottom
border-bottom-color
border-bottom-style
border-bottom-width

border-left
border-left-color
border-left-style
border-left-width
```

---

**border-left and border-bottom properties**

If your borders differ on some sides,
you must use the longer border-*side* properties.

**Example:**

```
p {
    border-bottom: 3px dashed green;
    border-left: 3px dashed green;
}
```

*This defines a three pixel wide dashed green border for all paragraphs*

You should always use the CSS shorthand properties when possible

# Thinking About Design

# Content

Websites are about conveying information

That "information" is your content

# Content is the most important part of a website

# Audience

Who is your audience?

What do they want?

Does your website provide it?

# Content and Audience

Know your audience

Know your objective

Design your site to effectively convey your content to your audience

# www.stackoverflow.com

# Exercise

What is the objective of stackoverflow?

What is the content? How is it generated?

Who is the target audience?

How does the design of the website facilitate the website objective?

Is the site easy and intuitive to use?