

Advanced Selectors

Styling Specific Objects

- We have focused on *type* selectors.
- What if you don't want to style *all* of the links, just some? Or just some of the lists?
- CSS gives you options

CSS Selectors

- **Some selectors follow the DOM**
- **Descendant selectors (nav a)**
 - Style all of the anchor links inside a nav tag
- **Child selectors (nav > a)**
 - more constraining The anchor elements must be a child of the nav, no intermediate tags, e.g. paragraph
- **Adjacent sibling (h1 + o)**
 - elements must be at same level and follow each other

id Selectors

- **# id selector**
 - Used to identify a single element in the DOM.
 - Was used extensively for `<div id = “header”>`, `<div id=“footer”>`, etc.
 - There is a small movement to move the use of id **OUT** of CSS

```
#mainLogo{  
  border: 5px solid #0006CC;  
  margin:0 auto;  
}
```

```

```


class Selector

- **. class selector**
 - Used to identify an element in the DOM that is part of a special class of items
 - Think of thumbnail images, all of the links that are in the navigation, your social media images, etc....

```
.thumb{  
  border: 1px solid #0006CC;  
  width:20%;  
}
```

```
  
  

```

classes vs. ids

- Syntax is “.” and “#”
- classes can be used multiple times
- id should be unique
- Think of images and navigation bars
 - Format numerous (but not all) images the same way
 - Visually signify the current page

Example

Narrowing the Scope

- As you get more advanced pages, you will want to narrow the scope of the of action
- **p.main** → paragraphs using main class
- **header img.special** → paragraphs inside header that use special class

Expanding the scope

- You can combine elements with a comma
 - `p, h1, #main, .special`{...rules to apply to all of them...}
- Review : What happens when there are multiple rules for the same selector?
 - When there are conflicts, use the one processed most recently
 - UNLESS a rule has `!important`

More Attribute Selectors

- **Universal**
 - * applies styling to every element on the page
 - Ackk!! Try this!
- **Attribute Selectors**
 - `a[href="info.html"]`
- **PseudoClasses**
- **Pseudo Elements**

Attribute selectors

- You may want to search the DOM for certain elements that have an attribute you are looking for
 - All the images that use gif files.....
 - All of the images that have empty alt text....
 - All of the links that go to government sites....

Using Operators

- Operators can be used to find those attribute values you are looking for

^ : match the beginning exactly

`a [href^='http://umich']`

\$: match the end exactly

`img[src$ = '.png']` → apply to .png images

***** : wildcard

`a [href*='umich']`

Example

Whew!!!

- We have actually covered a lot in this short video
- Know that each of these ideas can merge. One element can have many classes and ids associated with it

```
<li class="special early dark" id="main"/>
```

- Browser “starts at the top” and applies each rule, sometimes overriding earlier rules.

The Good News

- You can use style sheets from others to style your code, just by adding class!!
- You can override style sheets from others just by rewriting the class, or making your own version of it and linking it last.

Review

- **Type selectors can be combined to narrow the scope of where rules are applied**
- **An id is used to specify a specific element in a page**
- **Classes can be used to associate elements that should be treated in a similar manner**

Acknowledgements/Contributions

These slides are Copyright 2015- Colleen van Lent as part of <http://www.intro-webdesign.com/> and made available under a Creative Commons Attribution Non-Commercial 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Colleen van Lent , University of Michigan School of Information

