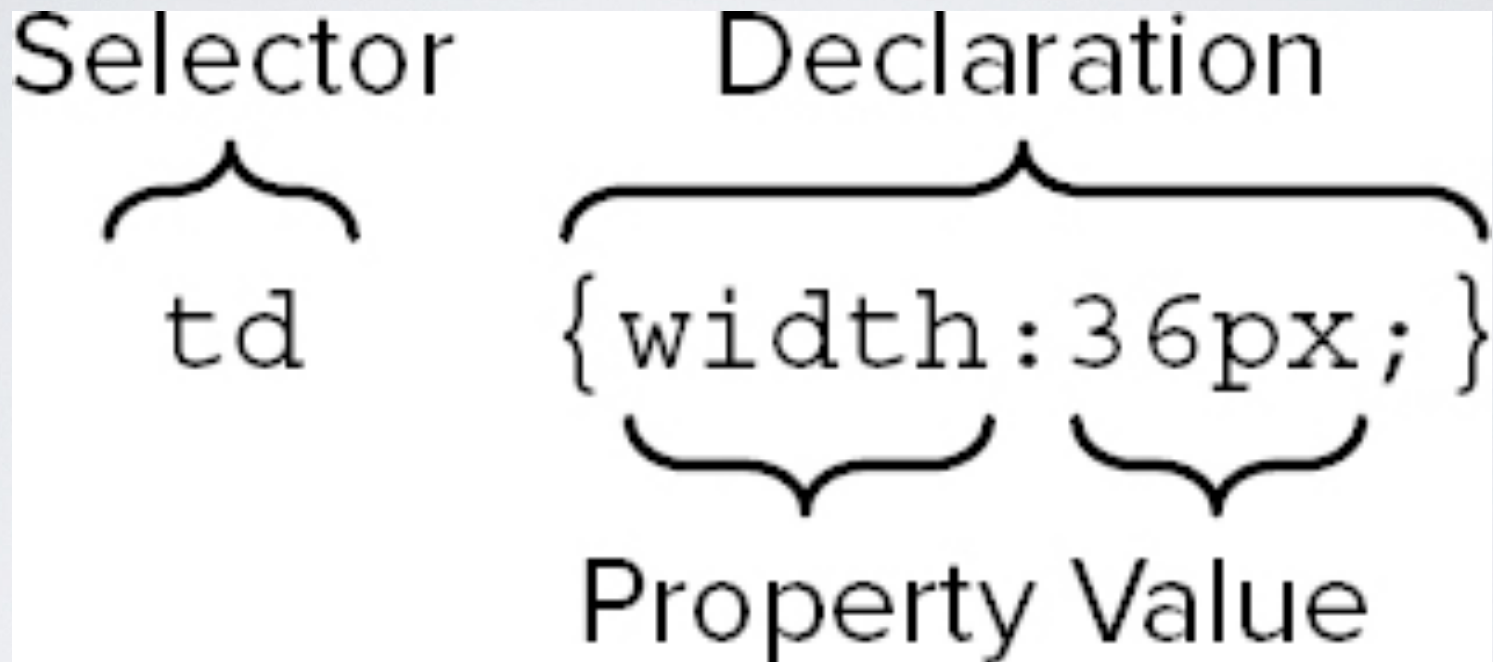


IA FOR THE WEB

CSS - Part 2

Fall 2014

REVIEW: SYNTAX



FACTORS IN RULE PRECEDENCE

- Property inheritance from parent
- Stylesheet origin
- Rule importance
- Rule specificity
- Rule order in stylesheet

ALGORITHM FOR PRECEDENCE

- A. Find all rules that apply to the current element, including those that could be **inheritable** from the parent. Remove parent rules that are overridden by this element.
- B. Sort rules according to **origin** and **importance** and remove rules that are overridden by higher numbers
 1. user agent (browser) declarations
 2. user normal declarations
 3. author normal declarations
 4. author important declarations
 5. user important declarations
- C. Within 1-5, sort by **specificity** and remove less specific rules
- D. Within 1-5, order by the **order** the rules appear in the stylesheet and remove earlier rules.

HTML

```
<p class="example">This is <em>fun</em></p>
```

Browser CSS

```
p{font-family: "Times New Roman", serif;  
  font-size: 16px;  
  font-weight: normal;  
  color: black;  
  background-color: white;}
```

```
em{font-style: italic;}
```

Author CSS

```
p{font-size: 18px;}  
p.example{color: blue; font-size: 17px;}  
p.example{color: orange; font-weight: bold;}  
em{font-size: 20px;}
```

PEXAMPLE: STEP A

- Find all rules that apply to the current element, including those that could be inheritable from the parent. Remove parent rules that are overridden by this element.

```
p{font-family: "Times New Roman", serif;}  
p{font-size: 16px;}  
p{font-weight: normal;}  
p{color: black;}  
p{background-color: white;}  
p{font-size: 18px;}  
p.example{color: blue;}  
p.example{font-size: 17px;}  
p.example{color: orange;}  
p.example{font-weight: bold;}
```


P.EXAMPLE: STEP B

- Sort rules by origin and importance and remove rules that are overridden by those with higher numbers

```
1. p{font-family: "Times New Roman", serif;}
```

```
1. p{font-size: 16px;}
```

```
1. p{font-weight: normal;}
```

```
1. p{color: black;}
```

```
1. p{background-color: white;}
```

```
3. p{font-size: 18px;}
```

```
3. p.example{color: blue;}
```

```
3. p.example{font-size: 17px;}
```

```
3. p.example{color: orange;}
```

```
3. p.example{font-weight: bold;}
```

PEXAMPLE: STEP C

- Within 1-5, sort by **specificity** and remove less specific rules.
 1. p{font-family: "Times New Roman", serif;}
 - ~~1. p{font-size: 16px;}~~
 - ~~1. p{font-weight: normal;}~~
 - ~~1. p{color: black;}~~
 1. p{background-color: white;}
 - ~~3. p{font-size: 18px;}~~
 3. p.example{color: blue;}
 3. p.example{font-size: 17px;}
 3. p.example{color: orange;}
 3. p.example{font-weight: bold;}

PEXAMPLE: STEP D

- Within 1-5, order by the **order** the rule appears in the stylesheet and remove earlier rules.

1. p{font-family: "Times New Roman", serif;}

~~1. p{font-size: 16px;}~~

~~1. p{font-weight: normal;}~~

~~1. p{color: black;}~~

1. p{background-color: white;}

~~3. p{font-size: 18px;}~~

~~3. p.example{color: blue;}~~

3. p.example{font-size: 17px;}

3. p.example{color: orange;}

3. p.example{font-weight: bold;}

P.EXAMPLE: RESULT

```
p.example{  
  font-family: "Times New Roman", serif;  
  background-color: white;  
  font-size: 17px;  
  color: orange;  
  font-weight: bold;  
}
```

EM: STEP A

- Find all rules that apply to the current element, including those that could be inheritable from the parent. Remove parent rules that are overridden by this element. (For our purposes, for parent rules, use the *result* of the prior computation)

```
p.example{font-family: "Times New Roman", serif;}
```

```
p.example{background-color: white;}
```

```
p.example{font-size: 17px;}
```

```
p.example{color: orange;}
```

```
p.example{font-weight: bold;}
```

```
em{font-style: italic;}
```

```
em{font-size: 20px;}
```


EM: STEP B

- Sort rules by origin and importance and remove rules that are overridden by those with higher numbers

```
1. p.example{font-family: "Times New Roman", serif;}
```

```
1. p.example{background-color: white;}
```

```
1. em{font-style: italic;}
```

```
3. p.example{font-size: 17px;}
```

```
3. p.example{color: orange;}
```

```
3. p.example{font-weight: bold;}
```

```
3. em{font-size: 20px;}
```

EM: STEP C

- Within 1-5, sort by **specificity** and remove less specific rules.
 - 1. p.example{font-family: "Times New Roman", serif;}
 - 1. p.example{background-color: white;}
 - 1. em{font-style: italic;}
 - ~~3. p.example{font-size: 17px;}~~
 - 3. p.example{color: orange;}
 - 3. p.example{font-weight: bold;}
 - 3. em{font-size: 20px;}

EM: STEP D

- Within 1-5, order by the **order** the rule appears in the stylesheet and remove earlier rules.
 1. `p.example{font-family: "Times New Roman", serif;}`
 1. `p.example{background-color: white;}`
 1. `em{font-style: italic;}`
 - ~~3. `p.example{font-size: 17px;}`~~
 3. `p.example{color: orange;}`
 3. `p.example{font-weight: bold;}`
 3. `em{font-size: 20px;}`

EM: RESULT

```
em{  
  font-family: "Times New Roman", serif;  
  background-color: white;  
  font-size: 20px;  
  color: orange;  
  font-weight: bold;  
  font-style: italic;  
}
```

REVIEW: BOX MODEL



MORE SELECTORS

Example	Selects
<code>*</code>	All elements
<code>ul, ol</code>	All ul and all ol
<code>main p</code>	All p that descend from main
<code>main > p</code>	All p with main parent
<code>p + ul</code>	All ul that come immediately after p
<code>p ~ ul</code>	All ul that come after p (but not necessarily immediately)

MORE SELECTORS

Example	Selects
<code>[alt]</code>	All elements with any <code>@alt</code>
<code>[href="<u>http://google.com</u>"]</code>	All elements with <code>@href</code> of <u>http://google.com</u>
<code>[href^="https"]</code>	All elements with an <code>@href</code> that starts with "https"
<code>[src\$=".png"]</code>	All elements with a <code>@src</code> that ends with ".png"
<code>[type*="order"]</code>	All elements with a class that contains "order"

COMBINING SELECTORS

- `header + section > a`
- Always selecting for what's on the right
- Read from right to left

PSEUDO-CLASS SELECTORS

- Selects for elements in a given state
- Like classes but different in that the states change
- Good reference at <https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

PSEUDO-CLASSES

- `a:hover{color:red;}`
- `a:active{color:#CCCCCC;}`
- `a:visited{color:#333333;}`

PSEUDO-CLASSES

- `li:first-child{font-weight:bold;}`
- `p:last-child{font-variant:small-caps;}`
- `tr:nth-child(odd){background-color:#CCC};`
- `li:first-of-type{font-family: Helvetica};`

PSEUDO-CLASSES

- `a:not([href*="indiana.edu"]){background-color: #FF0000;}`
- `p:not(:first-of-type){margin-top:10px;}`

BRAIN TEASERS

- `p.warning > span + a[href*="indiana.edu"]`
- `#products li.comingSoon + li.outOfStock`
- `ul > li > ul > li`
- `[class] [class] > img[src$=".gif"]`

DISPLAY

- inline
- block
- inline-block
- none

DISPLAY: INLINE-BLOCK

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod ultrices ante, ac laoreet nulla vestibulum adipiscing. Nam quis justo in augue auctor imperdiet. Curabitur aliquet orci sit amet est posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius condimentum dolor, quis ultricies ipsum porta quis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod ultrices ante, ac laoreet nulla vestibulum adipiscing. Nam quis justo in augue auctor imperdiet. Curabitur aliquet orci sit amet est posuere consectetur. Fusce nec leo ut massa viverra venenatis. Nam accumsan libero a elit aliquet quis ullamcorper arcu tincidunt. Praesent purus turpis, consectetur quis congue vel, pulvinar at lorem. Vivamus varius condimentum dolor, quis ultricies ipsum porta quis.

```
p {  
  width: 180px; margin: 10px; padding: 20px;  
  font-size: 14px;  
  background: #eee;  
  display: inline-block;  
}
```


VISIBILITY

- visible
- hidden

DISPLAY:NONE OR VISIBILITY:HIDDEN?

- A hidden element still takes up space
- A element with **display:none** takes up no space. It's like it's not there at all
- See http://www.w3schools.com/css/css_display_visibility.asp

POSITIONING

- By default, an element is statically positioned
- This means each element is rendered in the order it appears in the document (called “document flow”)
- Child elements are positioned with respect to their parent elements

POSITION

- static
- absolute
- fixed
- relative

POSITION: STATIC

- Default
- Elements are rendered in the order they appear in the document

POSITION: RELATIVE

- `top`, `bottom`, `left`, `right` move the element a certain number of units relative where it would have been statically positioned

POSITION: FIXED

- Fixed positioned elements are taken out of the document flow
- Positioned according to the browser window and doesn't move
- Use `top`, `bottom`, `left`, `right` to position

POSITION: ABSOLUTE

- Absolutely positioned elements are taken out of the document flow
- Positioned according to the first non-static ancestor (or the browser window if there is none)
- Use **top**, **bottom**, **left**, **right** to position
- If you need to define an non-static ancestor but don't want to change its position, make it **position: relative** but don't set **top**, etc.

POSITION RECAP

Type of Positioning	Meaning of top, bottom, left right	Element Stays in Document Flow	Notes
static	None	Yes	
relative	How much to move the element from where it would be statically positioned	Yes	Think of this as an adjustment
fixed	Where to position the element with respect to the browser window	No	Does not move when the page moves
absolute	Where to position the element with respect to the most recent non-static ancestor	No	

Z-INDEX

- A number that determines which elements are in front of other elements
- The higher z-index will be in front

FLOAT

- Floating takes an element out of the document flow and moves it to the left or right within its parent
- If you float multiple siblings, they will essentially stack up next to each other
- Floating an element can change the order of siblings

CLEAR

- Clearing an element makes it so that sibling elements cannot float next to each other

@FONT-FACE

- Defines a font to be used later
- `@font-face{font-family: grant-serif; src: url(grant-serif.woff);}`
- `p{font-family: grant-serif, "Times New Roman", Times, serif};`
- Easy to use with Google Fonts

CHECK COMPATIBILITY

- <http://css3test.com/>
- <http://caniuse.com/>

EXERCISES

- http://www.teaching-materials.org/htmlcss-1day/css-boxmodel/exercise_descrip.html
- http://www.teaching-materials.org/htmlcss-1day/css-positioning/exercise_descrip.html
- http://www.teaching-materials.org/htmlcss-1day/css-floats/exercise_descrip.html