

Objectives

- CSS Review
- Discuss:
 - How Google Search Works
 - What Images You Can Use
- HTML Forms

April 28, 2016

Sprenkle - CSC1335

1

CSS Review

- Why CSS?
- What is the syntax of a CSS rule?
- What is the order of applying rules in the cascade?
- How can we apply a rule to a particular element?
- How did Lab 1 go?

April 28, 2016

Sprenkle - CSC1335

2

Identify the Errors

```
<style>
body { background-color: white
h1, { gray; font-family: sans-serif; }
h2, p { color: }
<em> { font-style: italic; }
</style>
```

April 28, 2016

Sprenkle - CSC1335

3

Fixed CSS

```
<style>
body { background-color: white; }
h1, { color: gray; font-family: sans-serif; }
h2, p { color: blue; }
<em> { font-style: italic; }
</style>
```

Could be other ways to fix the errors

April 28, 2016

Sprenkle - CSC1335

4

CSS Organization

- Using Bootstrap
 - Create your own separate CSS file and put your changes in that file
 - Your custom CSS file should go after the Bootstrap CSS file in the head element

April 28, 2016

Sprenkle - CSC1335

5

The image shows the Google logo in its multi-colored font, followed by the word "SEARCH" in a blue, sans-serif font.

April 28, 2016

Sprenkle - CSC1335

6

Google Search Discussion

- How does Google's search work?
- What are some of its novel features?
- What are some recommendations for web masters?
 - Any surprises?
- Why is search engine optimization important?
 - It's so important that it has its own acronym: SEO

April 28, 2016

Sprengle - CSC1335

7

Google Search Discussion

- What are Google search's limitations?
 - How can it be manipulated?
- What features would you like to add to Google's searching technique (or other search engine)?
- What is your preferred search engine? Why?

April 28, 2016

Sprengle - CSC1335

8

Google Search Discussion

- Knowing more about Google search, will you change your queries any?
 - How can you make them more effective?
 - How can you save yourself time?
- Now that you're an author of HTML documents, will you change them to get a higher rank on Google?
 - How could you do that?

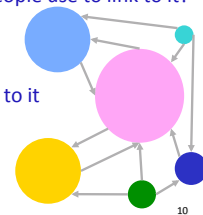
April 28, 2016

Sprengle - CSC1335

9

Simplified Google PageRank

- Developed by Larry Page and Sergei Brin
 - Then graduate students at Stanford University
- How do we know what a page is about?
 - One approach: What text do people use to link to it?
- How should we rank pages?
 - By number of links to it
 - By popularity of pages that link to it



April 28, 2016

Sprengle - CSC1335

10

INTERNET COPYRIGHT

April 28, 2016

Sprengle - CSC1335

11

What Can You (Legally) Take From the Web?

April 28, 2016

Sprengle - CSC1335

12

What Can You (Legally) Take From the Web?

- **What** you take, **why** you take, **who** you are, what technology you use
 - Commercial or non-profit
 - Amount and substantiality of portion used
 - Nature of copyrighted work
- What is fair use? *Tough question!*
 - Rules change as technologies change

April 28, 2016

Sprengle - CSC1335

13

Web Interfaces

- Menus
 - Sidebars, links, navigation
- GUIs
- Forms ← Today's focus

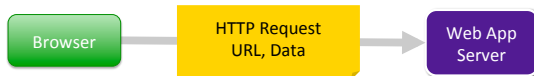
April 28, 2016

Sprengle - CSC1335

14

HTML form Tag

- Used to provide web application input from the user
- Contains various **controls** or **widgets** (sources of input) and labels for controls
- Must have a **submit** button that transmits all input data to server as a query string

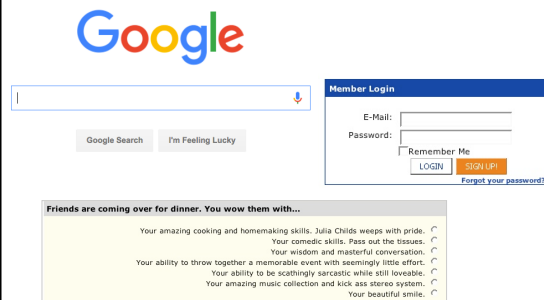


April 28, 2016

Sprengle - CSC1335

15

Examples of Forms/Input



April 28, 2016

Sprengle - CSC1335

16

HTML form Tag

- **form** attributes:
 - **action** (required): URL to send request to
 - Relative or absolute
 - **method**: **get** or **post**
 - Default: Get
 - More on differences later...

```
<form action="/search">
  <!-- input fields, etc. -->
</form>
```

April 28, 2016

Sprengle - CSC1335

17

Input From Forms

- Types of input data
 - Text
 - Checkbox
 - Radio buttons
 - Select from list
 - Password
- Data is identified by a **name**, has a **value**
 - Specified by input fields' **name** attributes

April 28, 2016

Sprengle - CSC1335

18

Simplified Form for Google Search

```
<form action="/search">
  <p>
    <input type="text" name="q" size="55"/>
    <input type="submit" value="Google Search"
      name="btnG"/>
  </p>
</form>
```

- Form is submitted to Google's `/search` page with data `q` set to whatever user entered in box, e.g.,
 - `http://www.google.com/search?q=user_query`

April 28, 2016

Sprenkle - CSC1335

search.html

19

Another Way to Invoke A Web Application

- Example: My digital library
 - Keeps publications, presentations
- Create a link to
 - `http://servo.cs.wlu.edu/pubs/browse?type=author&starts_with=M`
- Browser sends a **GET** request to the server's **dspace/browse** page
 - 2 Data Parameters:
 - name is **type**, which has value **author**
 - name is **starts_with**, which has value **M**

April 28, 2016

Sprenkle - CSC1335

20

Query Strings

- Do you always want your input data to show up in the URL?

April 28, 2016

Sprenkle - CSC1335

21

get vs post

- get** passes parameters to server as a query string
 - Limited to browser's URL length (2048 characters)
- post** embeds the parameters in *HTTP request*
 - Not in the URL

April 28, 2016

Sprenkle - CSC1335

22

get vs post

Advantages of **post**

- Information is more private (not shown in URL)
- Can't be bookmarked

Disadvantages of **post**

- Can't be bookmarked
- Browser can't easily go back (POSTDATA error)

April 28, 2016

Sprenkle - CSC1335

23

input Tag

- Used to create many of the widgets
- type** attribute specifies the type of widget
- Must be inside a block-level element
- Always close tag when opened
 - Contains attributes
- Examples:
 - text, checkbox, radio
- Often requires **name** attribute
 - Names the data that will be sent to the Web application

April 28, 2016

Sprenkle - CSC1335

24

Text input

Enter Text:

- A horizontal box that the user can input text into

```
<input type="text" name="name" size="25"/>
```

Attributes:

size	Width of text box in characters; scrolls if user types more
maxlength	Maximum number of characters browser accepts in a box
value	Provide a default value

Examples of when to use this?

April 28, 2016 Sprengle - CSC1335 25

Password input

Password:

- A horizontal box that the user can input text into **but** the text displays as *s or *s

```
<input type="password" name="mypassword" size="10" maxlength="10"/>
```

Attributes:

size	Width of text box in characters; scrolls if user types more
maxlength	Maximum number of characters browser accepts in a box

April 28, 2016 Sprengle - CSC1335 26

Labeling input fields

- Label a control with the **label** tag
- Better than labeling with other text because
 - Can get separated during maintenance
 - label** has special presentation
 - Improved usability

```
<p>
<label>Password:
<input type="password"
name="mypassword" size="10"
maxlength="10"/>
</label>
</p>
```

textbox.html

April 28, 2016 Sprengle - CSC1335 27

Multiple Choice Input: checkbox

- Use when user has multiple choices for something and can choose ≥ 1

Milk Bread Eggs

- Only items that user checks are sent by the form to the action location

checkbox.html

April 28, 2016 Sprengle - CSC1335 28

Multiple Choice Input: checkbox

- All checkboxes in a group have the same **name**
- Checkbox requires a **value** attribute
 - value** is submitted in the form data iff the checkbox is 'checked'
- To make a checkbox checked, the **checked** attribute is set to **checked**

```
<label>
<input type="checkbox" name="groceries"
value="milk" checked="checked"/>Milk
</label>
```

April 28, 2016 Sprengle - CSC1335 29

Multiple Choice Input: checkbox

```
<label>
<input type="checkbox" name="groceries"
value="milk" checked="checked"/>Milk
</label>
<label>
<input type="checkbox" name="groceries"
value="bread"/>Bread
</label>
<label>
<input type="checkbox" name="groceries"
value="eggs"/>Eggs
</label>
```

Notice order of label/input, Label and value are different

Milk Bread Eggs

April 28, 2016 Sprengle - CSC1335 30

Multiple Choice Input: **checkbox**

- Discussion: When designing a form, when should a checkbox be **checked** by default?

```
<label>
  <input type="checkbox" name="groceries"
    value="milk" checked="checked"/>Milk
</label>
```

April 28, 2016

Sprengle - CSC1335

31

Multiple Choice Input: **checkbox**

- Discussion: When designing a form, when should a checkbox be **checked** by default?
 - Common value that people will always want
- Have you ever seen a checkbox turned on when you think it shouldn't be?

```
<label>
  <input type="checkbox" name="groceries"
    value="milk" checked="checked"/>Milk
</label>
```

April 28, 2016

Sprengle - CSC1335

32

Multiple Choice Input: **radio**

- *Only one* radio button in a group can be on or pressed
 - Groups of radio buttons are identified with the same **name**

Why is **radio** appropriate?

```
<label> <input type="radio" name="age"
  value="under20" checked="checked"/>0-19
</label>
<label> <input type="radio" name="age"
  value="20-35"/>20-35
</label> ...
<label> <input type="radio" name="age"
  value="over65"/>&gt;65
</label>
```

Same name

0-19 20-35 ... >65

April 28, 2016

Sprengle - CSC1335

33

Alternative for **label**

- Use **for** attribute to specify which control you're labeling
 - **for**'s value is the control's **id**

```
<label for="age_under18">0-17</label>
<input id="age_under18" type="radio"
  name="age" value="under18"/>
```

Important for usability

April 28, 2016

Sprengle - CSC1335

34

Menus with **<select>**

- Displays large number of options more compactly
- Can emulate radio buttons (only one selection, default) or checkboxes (multiple selections)

name	Name of the data
size	# of items to display
multiple	Allows multiple selections if value is multiple

```
<select name="age">
  ...
</select>
```

April 28, 2016

Sprengle - CSC1335

35

option tag

- Value options are in **option** tags, nested inside of **select** tags
- Can preselect an option with **selected** attribute set to "selected"

```
<select name="age">
  <option value="under20"
    selected="selected">0-19</option>
  <option value="20-35">20-35</option>
  ...
  <option value="over65">&gt; 65</option>
</select>
```

What is your age?

- 0-17
- 18-25
- 26-45
- 46-64
- 65 and older

April 28, 2016

Sprengle - CSC1335

36

select Tag Example

- Emulating checkboxes

```
<select name="groceries" multiple="multiple">
  <option value="milk"
    selected="selected">Milk</option>
  <option value="bread">Bread</option>
  <option value="eggs">Eggs</option>
</select>
```

Alternative Example:

Which types of films do you like to watch?

 Action
 Comedy
 Foreign

April 28, 2016

Sprengle - CSC1335

37

Using select

- Any advantages or disadvantages to using **select** rather than radio buttons or checkboxes?

April 28, 2016

Sprengle - CSC1335

38

Using select

- Any advantages or disadvantages to using **select** rather than radio buttons or checkboxes?
 - **select** takes up less space when only one selection BUT user must click to see all options

April 28, 2016

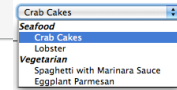
Sprengle - CSC1335

39

Option Groups: optgroup

- Tag used to group options with a label
 - Can also apply a style to label

```
<select name="entree">
  <optgroup label="Seafood">
    <option value="crabcakes">Crab Cakes</option>
    ...
  </optgroup>
  <optgroup label="Vegetarian">
    <option value="spaghetti">Spaghetti</option>
    ...
  </optgroup>
</select>
```



April 28, 2016

Sprengle - CSC1335

40

textarea Tag

- Allows users to input multiple lines of text

name	Name of the data
rows	# of rows for text area
cols	# of characters wide for text area

```
<p>Please provide your yearbook memories:</p>
<p><textarea name="memories" rows="3"
  cols="40"> (Be brief and concise.)
</textarea></p>
```

Can't close the tag when opened. Needs content, even if empty.

Displayed by default in the text area.

April 28, 2016

Sprengle - CSC1335

41

Grouping Input: fieldset, legend

- **fieldset**: groups related **input** fields
- **legend**: supplies an optional caption

```
<fieldset>
  <legend>Credit cards:</legend>

  <input type="radio" name="creditcards" id="visa" />
  <label for="visa">Visa</label> <br />

  <input type="radio" name="creditcards"
    id="mastercard" />
  <label for="mastercard">MasterCard</label> <br />

  <input type="radio" name="creditcards" id="amex" />
  <label for="amex">American Express</label><br />
</fieldset>
```

April 28, 2016

Sprengle - CSC1335

See buttons.html

42

submit and reset Buttons

- type = **submit**
 - When clicked, browser sends parameters to the server
 - Browser shows server's response
- type = **reset**
 - when clicked, browser changes the controls back to their original state

```
<input type="submit" value="Submit Query"/>
<input type="reset" value="Reset"/>
```

Submit Query Reset

April 28, 2016

Sprengle - CSC335

43

Styling Forms with CSS Attribute Selectors

```
element[attribute=value] {
  properties; ...
}
```

```
input[type="text"] {
  color: blue;
  font-style: italic;
  margin-bottom: 2em;
}
```

example text

- CSS attribute selector affects an element only if it has the given attribute set to the given value
- Often used with forms because **input** element represents many different controls

April 28, 2016

Sprengle - CSC335

44

Considerations for Choosing Controls

- How easy is it for the user to enter the input?
- What is the range of possible values?
- How many possible values are there?
- How much flexibility does the user have?
 - How much *should* they have?
- Should the data be hidden in the browser?

April 28, 2016

Sprengle - CSC335

45

What **input** would you use?

Input Data	Input Type
Gender	
Sensitive data	
Date (month, day, and year)	
Abstract for a paper	
Title for a song	
Household Income Bracket	

April 28, 2016

Sprengle - CSC335

46

TODO

- Lab 2: HTML Forms
 - Due tonight at midnight
- Requirements, Design, Work Plan Documents on Wiki – Friday
 - Coordination/cooperation, division of responsibilities
 - Does this work?
 - Due Friday at midnight

April 28, 2016

Sprengle - CSC335

47