CSCI 311

FORMS

Learning objectives

Quick (re)view of CSS

- Styles and selectors

By the end of these lectures you will

- Understand how forms can be used to collect information from users
- Be able to add basic forms to your web pages

Style declaration

General Form:

- `property:value`

Properties

- > 100 different properties
- some apply to any elements, others to specific elements or kinds of elements

Selectors

General Form:

- `selector { property1:value1; property2:value2;}`

Selectors

- let us designate which HTML elements are affected by which styles
- There are different kinds of selectors

Class and id

- in our html we can define the class and id attributes of html elements
- many elements can share a class
- id must be unique

Kinds of Selectors

Type (element) selector:

- associates a rule with every instance of that element
- example:
  - h3 {line-height:140%;}

Universal selector:

- * symbol selects every HTML element
- for example, can be used for all elements in a class:
  - *.class

Class selector:

- .className selects the elements in that named class

Kinds of Selectors

Attribute Selector:

- [attr] selects elements with a given attribute (with any value)
- example
  - [title]{ color: blue;}
- can also specify the value:
  - [alt=“bug”] {background-color: red;}

id selector:

- associates a rule with the HTML element that matches the unique id
- example:
  - #menuItem {color:blue;}

Kinds of Selectors

Descendent selector

- sequence of space separated selectors
- selects the last selector contained in the previous (descendants)
- example:
  - nav#left a img (selects a link anchoring img in nav with id="left")

Child selector

- selects elements that are **immediate** children of a specified element:
- example:
  - div > h3 (selects all h3 elements that are **immediate** children of a div

Kinds of Selector

Adjacent Sibling Selector

- selects all elements are immediate siblings
- example:
  - div + h3 (selects all h3 that are declared at same level and immediately after a div)

General Sibling Selector

- selects all elements are siblings
- example:
  - div ~ h3 (selects all h3 that are declared at same level as a div)

Kinds of selectors

Pseudo-class selectors

- permits selection based on conditions at run-time or hierarchical structure of the document
- section:target selects section immediately after the user clicks it
- examples:
  - a:link {color:#00c;}
  - a.visited {color:#300;}
  - a.active
  - a.hover…

In other news

Permanent Redirect

HTML Form

A form structure is a practical method for collecting info from the user

Forms are set up using a combination of

- HTML form tag elements
- scripts written in PHP, ASP, Perl, python or other script languages

Scripts are commonly called CGI

- Common Gateway Interface

Scripts are run on the web server (we will get to this later)

How do they work?

Front-end HTML markup is what users see:

- buttons, input fields, drop-downs, radio buttons
- These are also called Form Controls

Together with server-side scripts we can:

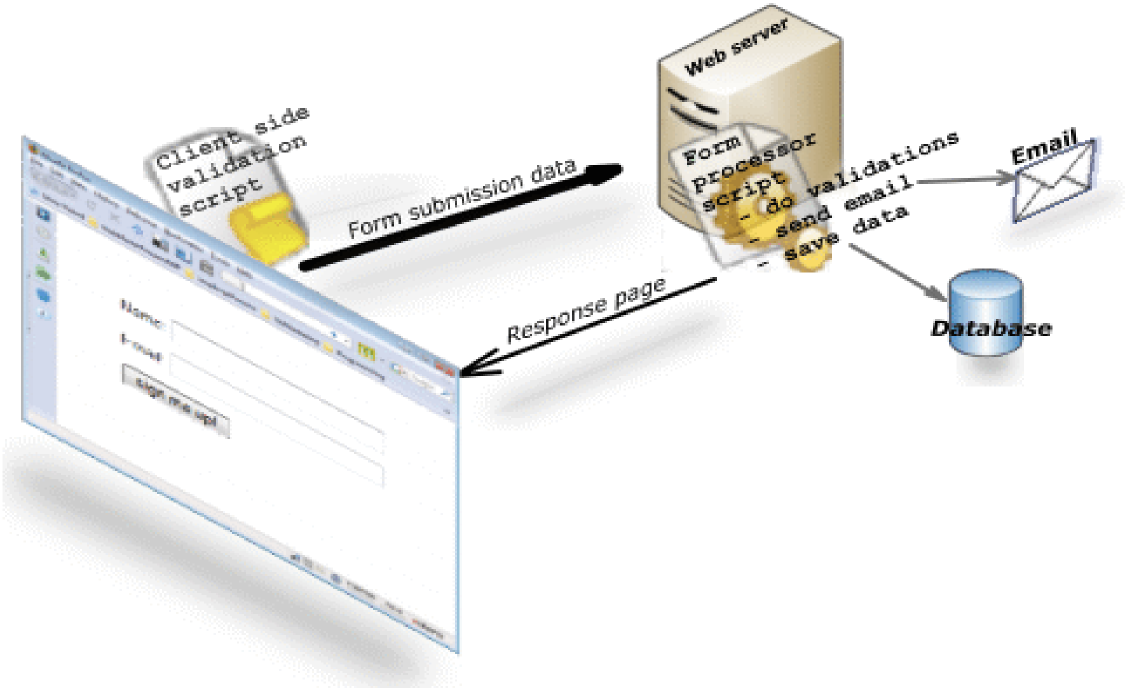
- store data from the form
- email it
- verify input
- return appropriate responses to users

Scripts are what make the form’s action attributes work

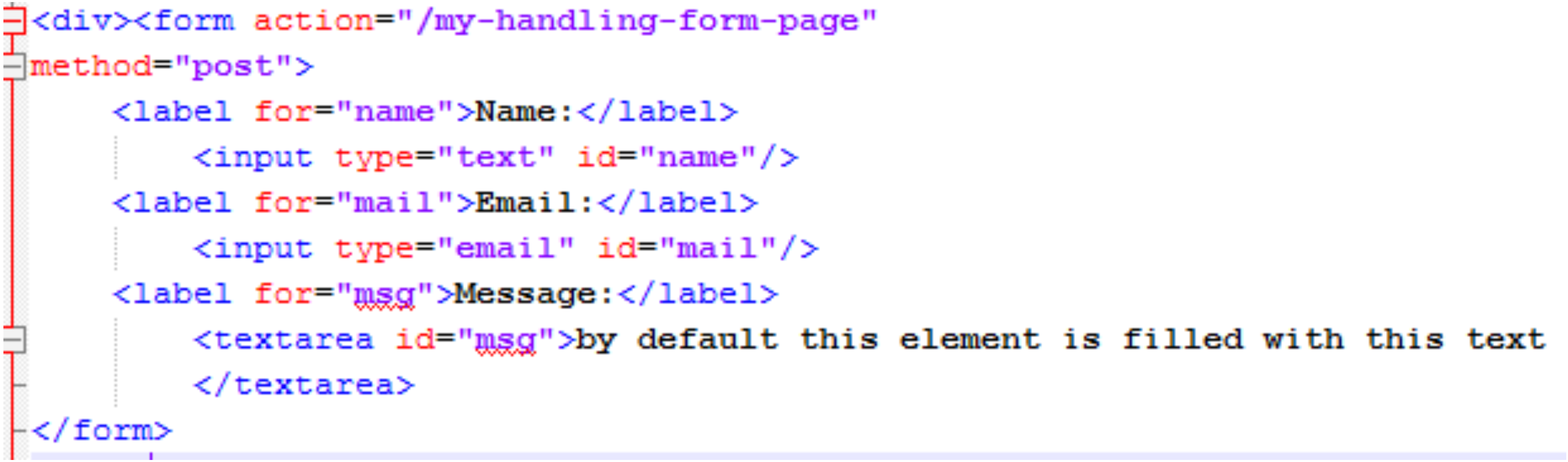
HTML Form

Each form element in which the user can enter information is called a *field*

Information entered into a field is called the *field value*, or simply the *value*



Simple Example



```
<div><form action="/my-handling-form-page"
method="post">
    <label for="name">Name:</label>
        <input type="text" id="name"/>
    <label for="mail">Email:</label>
        <input type="email" id="mail"/>
    <label for="msg">Message:</label>
        <textarea id="msg">by default this element is filled with this text
        </textarea>
</form>
```

html form controls

Controls (form elements) have attributes

- name, used to associate the form control’s current data when it is sent to the server (variable name – should be simple and descriptive)
- id, used to uniquely identify a form control in the DOM and associate a label (optional)
- value, used to provide an initial value to a form control (optional)

```
<label for=“fname”>Enter name:</label>
    <input type=“text” name=“firstname”
id=“fname” value=“Your first name”>
```

Form tag

The <form> element includes properties that control

- how the form is processed
- information on which script is used
- how the data is transferred to the script

```
<form name=“myForm” action=“script name”
method=“method name” >

...form elements

</form>
```

Form element

No nesting forms

Multiple forms per page ok, but each should have a unique *name* attribute value

Form names should not contain spaces

Omitting the form’s action attribute will not break the form, but will mean nothing happens when user clicks on the submit button

Form controls must be defined inside the <form> elements

input form element

input element is self-closing

Used for

- single line text
- check boxes
- radio buttons
- passwords
- ...

Attributes:

- type, tells the browser what kind of information is being input
- name, the name of the data field
- size, the length of the field in characters, default is 20
- maxlength, max number of chars to accept
- value, depends on type of input element

input element

There a number of types of input element:

- buttons, date pickers etc.

example:

- `<input name="firstname" type="text" size="15" maxlength="25" />`
- field will be 15 characters wide, and will allow a maximum of 25 characters to be input
- the name attribute can be used as a key to access the value
  - acts as a key-value pair
  - firstname="Sarah"
- value attribute can be used to set initial value
- placeholder attribute can be used to give an example or hint
  - often displayed greyed out

input element – text type

Type attributes

- text

```
<input name=“firstname”>

<input type=“text” name=“phone” size=“15”
maxlength=“12”>

<input name=“country” value=“Canada”>
```

input form element

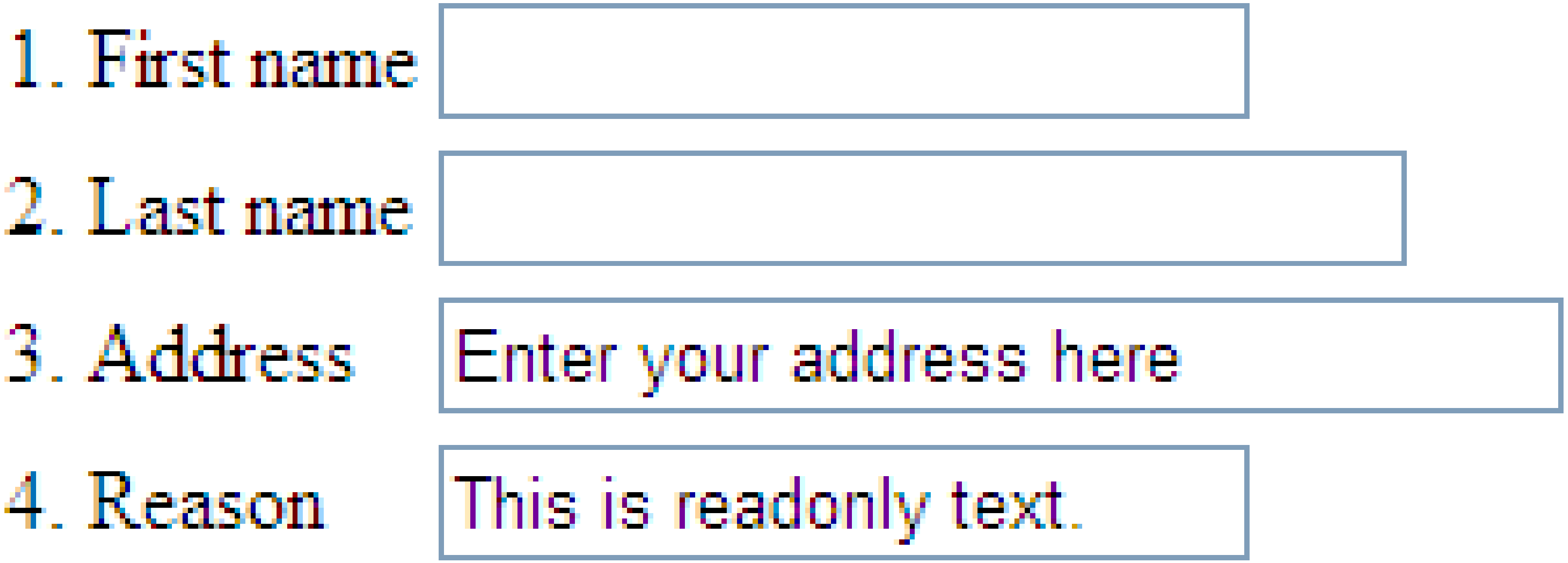
value attribute

- if the input element type is text or password, this is the default text to display
- for checkbox or radio buttons the value returned to the server
- for submit and reset buttons, the text to show inside the button

checked attribute

- sets a radio button or check box to “on”
- usage: checked=“checked”

HTML input text example



1. First name

2. Last name

3. Address Enter your address here

4. Reason This is readonly text.

Input element: password type

password

- a type of text field where the user’s typed characters are shown onscreen as bullets
- prevents echoing of characters
- does not **encrypt** data sent to server!

```
<input type=“password” name=“pw” size=“30” maxlength=“30”>
```

Input Element: Checkbox

checkbox

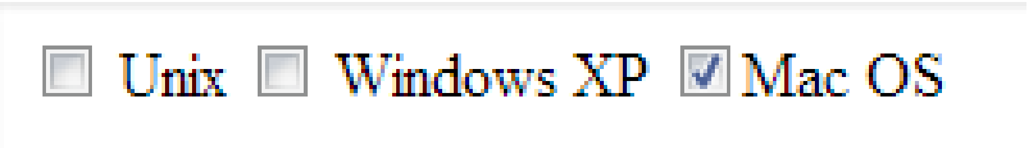
- a switch control is “on” when its checked attribute is set (e.g., checked)

checkbox controls within the form may share the same name value

- lets the user select zero, one or many selectable options
- make them preselected by setting checked=“checked”
- to make PHP scripts work easier, place [] at the end of the name attribute
- If value attribute is not provided then either 0 or 1 is used

Checkbox example

```
<input type=“checkbox” name=“os[]” value=“unix”> Unix
<input type=“checkbox” name=“os[]” value=“win”> Windows
<input type=“checkbox” name=“os[]” value=“mac”> Mac OS
```



□ Unix □ Windows XP ☑ Mac OS

input element: radio button

Allows only one option to be selected

All radio elements must have same name attribute

Can supply optional checked=“checked” to preselect which is default

```
<input type=“radio” name=“user” value=“home”>Home

<input type=“radio” name=“user” value=“business”
checked=“checked”>Business
```

input element: hidden type

creates an undisplayed field in the form

used to send calculated or default info to the server when the data is submitted

```
<input type="hidden" name="IM" value=abc@email.com>
```

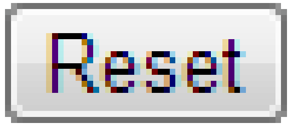
Input element: reset

displays a push button with the preset function of clearing all the data in the form back to its original state

name and value attributes are optional unless there is more than one reset button

value attribute can be used to show text in the reset button

```
<input type="reset">
```



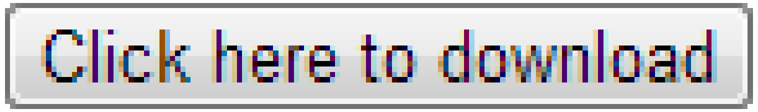
input element submit

displays a push button with the preset function of sending the entered data in the form to the server for processing

value attribute defines what text to show in the button

name and value attributes optional unless there is more than one submit button

```
<input type="submit" value="Click here to download"
name="download">
```



input element: file

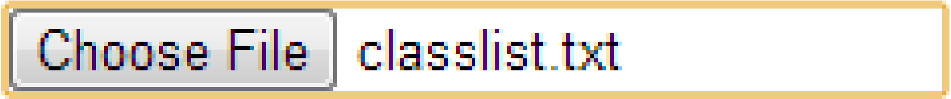
used to upload a file to the web server

```
<form action=“/client.php” method=“post”
enctype=“multipart/form+data”>

<input type=“file” name=“classlist” size=“28”/>

</form>
```

size indicates the character width of the text input field in Firefox



classlist.txt

Textarea

- A field for entering multiple lines of information
- by default, the same is a blank field of four lines, 40 characters wide each
- *name* attribute is required
- *maxlength* sets the limit of the number of characters that can be set into a field
- to control how the text is wrapped in a text area, use the *wrap* attribute (soft is default)

wrap="off" – turns off wrapping

wrap="soft" – turns text wrapping on, but does not send text wrapping data to web server

wrap="hard" – turns on text wrapping, and also sends text wrapping data to web server

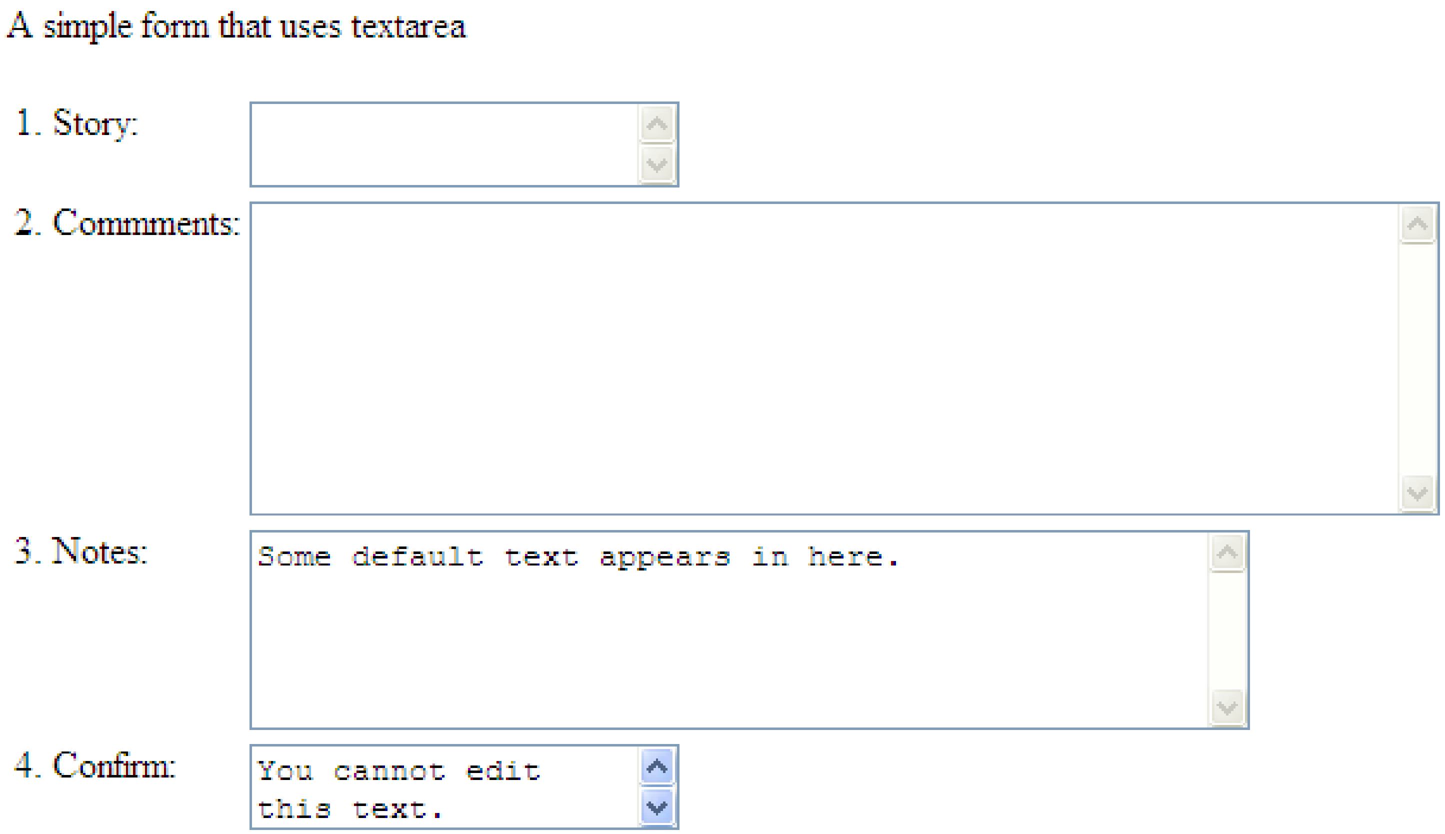
Textarea form controls

for more than one line of text use textarea instead of input

by default the text area in the form will be four lines and 40 chars wide

```
<form name=“myForm” ... >
     <textarea name=“notes”
          rows=“4”
          columns=“20”
          placeholder=“some default text here.”>
     </textarea>
</form>
```

HTML textarea example



A simple form that uses textarea

1. Story:

2. Commments:

3. Notes: Some default text appears in here.

4. Confirm: You cannot edit this text.

select form element

let’s you include pull-down menus

has two parts:

- <select> tag
- <option> tag

option tags are nested within the select tag

shows a list of options as a drop-down menu

attributes of select:

- number of options to show a user, size=“n”
- multiple selections use “multiple”

attributes of option:

- preselect options with selected
- value provides an overriding value to the server

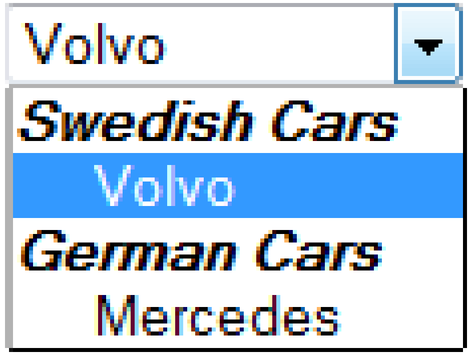
select example

```
<select name=“payment”>
      <option selected value=“credit”>Credit Card</option>
      <option value=“debit”>Debit Card</option>
      <option value=“cheque”>Cheque</option>
</select>
```

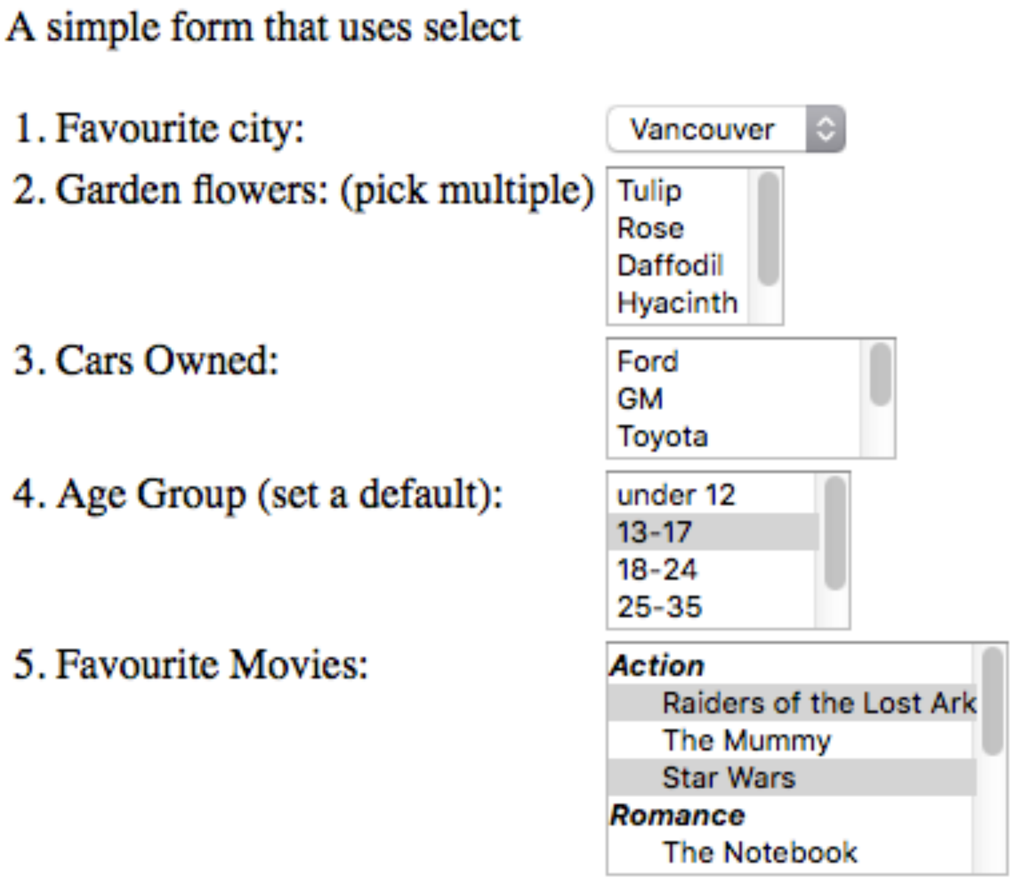
select optgroup

items can be grouped into a separate submenu

```
<select>
      <optgroup label=“Swedish Cars”>
            <option value=“Volvo”>Volvo</option>
      </optgroup>
      <optgroup label=“Gernal Cars”>
            <option value=“Mercedes”>Mercedes</option>
      </optgroup>
</select>
```



HTML select example



A simple form that uses select

1. Favourite city: Vancouver
2. Garden flowers: (pick multiple) Tulip
Rose
Daffodil
Hyacinth
3. Cars Owned: Ford
GM
Toyota
4. Age Group (set a default): under 12
13-17
18-24
25-35
5. Favourite Movies: ***Action***
Raiders of the Lost Ark
The Mummy
Star Wars
Romance
The Notebook

Find the errors

1.
   ```
   <input name=“country” value=“Your country here”>
   ```

2.
   ```
   <checkbox name=“color” value=“teal”>
   ```

3.
   ```
   <select name=“popsicle”>
   <option value=“orange”>
   <option value=“grape”>
   <option value=“cherry”>
   </select>
   ```

4.
   ```
   <input type=“password”>
   ```

5.
   ```
   <textarea name=“essay” height=“6” width=“100”>Your story:
   </textarea>
   ```

Find the errors

1. <input name=“country” value=“Your country here” type=“text”>
2. <checkbox name=“color” value=“teal”> not a real element.
   Should be: <input type=“checkbox” value=“teal”>
3. <select name=“popsicle”>
   <option value=“orange”>orange</option>
   <option value=“grape”> grape</option>
   <option value=“cherry”> cherry</option>
   </select>
4. <input type=“password” name=“psswd”>
5. <textarea name=“essay” rows=“6” cols=“100”>Your story:
   </textarea>

Which control to use?

If the answer is **input**, don’t forget the type attribute!

a) Choose your astrological sign (12 choices)

b) Indicate whether you have a history of heart disease (yes or no)

c) Write-up a review of a book

d) Select your favourite ice cream flavours from a list of 8

Which control to use?

If the answer is **input**, don’t forget the type attribute!

a) Choose your astrological sign (12 choices) <select>

b) Indicate whether you have a history of heart disease (yes or no) <input type=“radio”>

c) Write-up a review of a book <textarea>

d) Select your favourite ice cream flavours from a list of 8 <input type=“checkbox”>

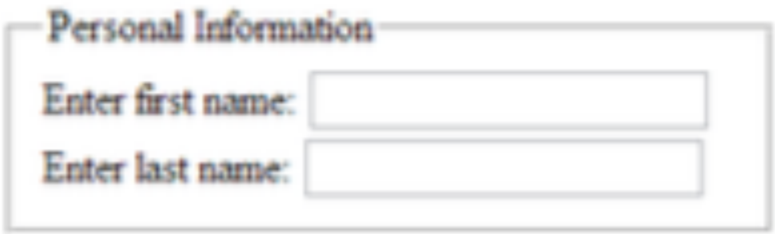
Organizing form elements

You can group different form components into a related group called a fieldset

Makes it easier for visitors to understand the form and fill it out properly

The legend tag is descriptive text

```
<fieldset>
        <legend>Personal Information</legen
        …
</fieldset>
```



Personal Information

Enter first name:

Enter last name:

Form label

Label tag enables CSS formatting to the form elements

```
<label for=“fname”>First Name:</label>

<input type=“text” name=“firstname” size=“15” id=“fname”>
```

Title attribute

Allows for tool-tips pop-up

Provides meaningful explanation or a short description of an element

```
<input type=“text” name=“dept” title=“Department Category”/>
```

Button element

User-defined image buttons

Define an **onclick** attribute and an image

```
<button onclick=“script/something”>
        <img src=“bluebutton.png”/>
        Click this blue button.
</button>
```

html5

New form elements and attributes in HTML5

Not supported by all browsers, however the features degrade nicely (e.g., the become regular text boxes)

<input type="tel> for phone numbers

<input type="number"> for numbers only

<input type="search"> search box

<input type="email"> email address

<input type="url"> for URL addresses

<input type="date"> for date information

<input type="color"> for colour selections

input element

Attributes for the <input> element

- placeholder – shows a light coloured hint of input
- autofocus – focuses on element when page is loaded
- required – if present, element cannot be left blank by user when submit is clicked
- pattern – the input data must match the provided regular expression
- autocomplete – default is on, set it to off for elements you don’t want autofilled (credit card info)

input element

The pattern attribute specifies a regular expression

- \d indicates a single number 0-9 (same is [0-9])
- \d{3} indicates exactly three digits
- \d{4,7} indicates min 4 max 7 digits needed

Use title attribute to let users know input

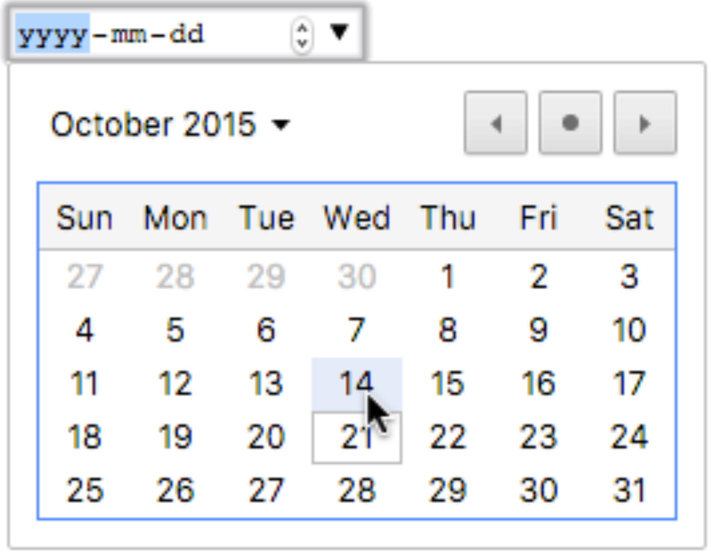
Special characters like hyphen, (, and) are enclosed by square brackets preceded by \

- [\-] or [\(] or [\)]

input element

<input type=“date”>

- Doesn’t work in all browsers



yyyy-mm-dd

October 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

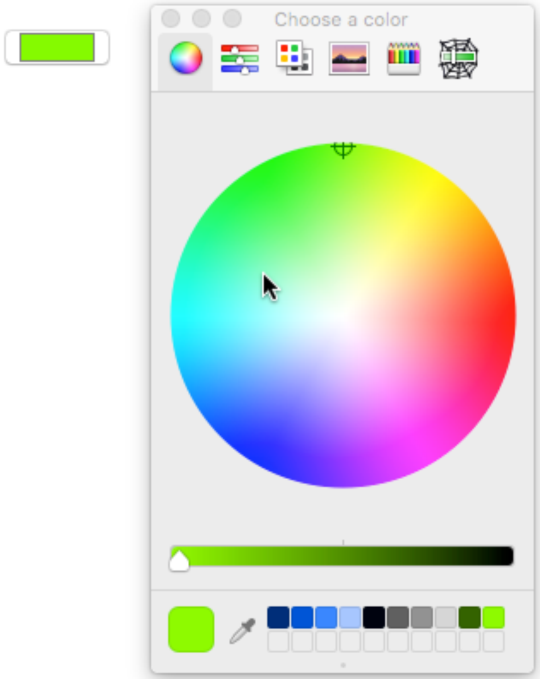
input element

<input type=“range” name=“skill” min=“1” max=“10” value=“1”>

```
<form oninput=
"amount.value=rangeInput.value">
<input type="range" id="rangeInput"
name="rangeInput" min="0" max="100">
<output name="amount"
for="rangeInput">0 </output>
</form>
```

input element

<input type=“color”>



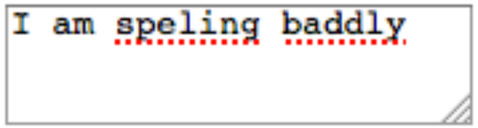
Spell check

Text fields can be spell checked using HTML5 spellcheck attributes

Based on the lang attribute in <html>:

- <html lang=“en”> means use English spelling

```
<textarea spellcheck=“true”></textarea>
```



I am speling baddly

Form design guidelines

Label input boxes with clear instructions

Group related form items into a fieldset

Control user’s entries with radio buttons, checkboxes and selection lists when possible

Let users know the correct format to enter date fields (yyyy/mm/dd)

Use selection lists for many possible options

Use radio buttons for five or fewer options

Form anomalies

Can a form prevent users from hitting submit twice?

- Nope, unless you use cookies or a unique id in the form

Some form elements can be easily styled with CSS:

- Form, fieldset, label, input, output

Other form elements are harder to style:

- Date, select, option, progress, meter, legend

Form action attribute

The <form> tag uses an action attribute

- Specifies the name of the script to run when the form’s submit button is clicked
- If no action attribute is provided, or the script does not exist, nothing will happen when submit is clicked
- CGI scripts are usually organized into their own folder sometimes called script or cgi-bin

```
<form name=“myform” action=“script/runform.php”>
```

Form method attribute

Method attribute controls how your browser sends data to the web server

- **Get** method is default. This sends the form data as a complete text appended to the URL. All the form data are down in the URL separated by ampersands

  http://www.myscript.com/…cgi?name=John+Smith&email=jsmith@mail.com
- **Post** method is *preferable*. This sends the form data in a separate data stream and is more flexible as most web servers truncate the amount of data they receive from the GET method

Summary

A quick look at the basics of making HTML forms

Go practice: http://www.w3schools.com/html/html_forms.asp