HTML Forms

Agenda

- Semantic Elements
- HTML Form
- Form elements

Semantic Elements

What are Semantic Elements?

- A semantic element clearly describes its meaning to both the browser and the developer.
- Examples of **non-semantic** elements: and - Tells nothing about its content.
- Examples of **semantic** elements: <form>, <table>, and <img> - Clearly defines its content.
  - ***<header>***
  - ***<footer>***
  - ***<nav>***
  - ***etc.***

Elements

- ***<fieldset>*** tag is used to group related elements in a form.
- ***<legend>*** tag defines a caption for the <fieldset> element.
- ***<label>*** tag defines a label for an <input> element.
- ***<select>*** element is used to create a drop-down list.
  - ***<optgroup>*** is used to group related options in a drop-down list.
  - ***<option>*** tags inside the <select> element define the available options in the list.

HTML Form

- An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

- The <form> tag is used to create an HTML form:

  <form>

  input elements

  </form>

- The <input> element is used to select user information.

- An <input> element can vary in many ways, depending on the type attribute. An <input> element can be of type text field, checkbox, password, radio button, submit button, and more.

The <form> tag

- The <form *arguments*> … </form> tag encloses form elements (and probably other elements as well)
- The arguments to form tell what to do with the user input
  - action="*url*" (required)
    - Specifies where to send the data when the Submit button is clicked
  - method="get" (default)
    - Form data is sent as a URL with ?form_data info appended to the end
    - Can be used *only* if data is all ASCII and not more than 100 characters
  - method="post"
    - Form data is sent in the body of the URL request
    - Cannot be bookmarked by most browsers
  - target="*target*"
    - Tells where to open the page sent as a result of the request
    - *target*= _blank means open in a new window
    - *target*= _top means use the same window

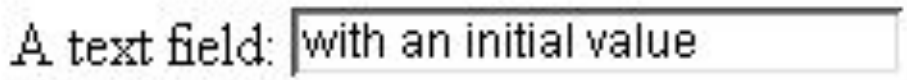
The <input> tag

- Most, but not all, form elements use the input tag, with a type="…" argument to tell which kind of element it is
  - type can be text, checkbox, radio, password, hidden, submit, reset, button, file, or image
- Other common input tag arguments include:
  - name: the name of the element
  - id: a unique identifier for the element
  - value: the "value" of the element; used in different ways for different values of type
  - readonly: the value cannot be changed
  - disabled: the user can't do anything with this element
  - Other arguments are defined for the input tag but have meaning only for certain values of type

Text input

A text field:

```
<input type="text" name="textfield" value="with an initial value" />
```



A multi-line text field

```
<textarea name="textarea" cols="24" rows="2">Hello</textarea>
```

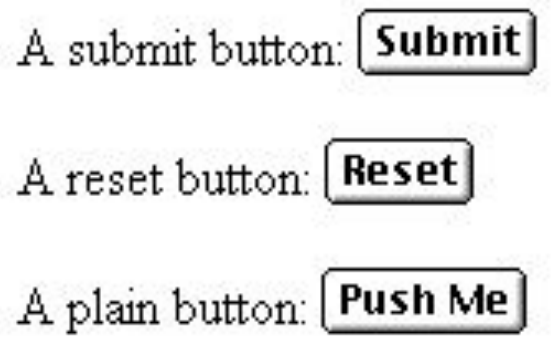
A password field:

```
<input type="password" name="textfield3" value="secret" />
```

- Note that two of these use the input tag, but one uses textarea

Buttons

- A submit button:
  `<input type="submit" name="Submit" value="Submit" />`
- A reset button:
  `<input type="reset" name="Submit2" value="Reset" />`
- A plain button:
  `<input type="button" name="Submit3" value="Push Me" />`



- submit: send data
- reset: restore all form elements to their initial state
- button: take some action as specified by JavaScript

- Note that the type is input, not "button"

The <button> Element

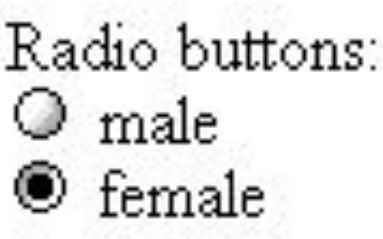
The <button> element defines a clickable button:

<button type="button" onclick="alert('Hello World!')"> Click me! </button>

The result:

Radio buttons

```
Radio buttons:<br>
   <input type="radio" name="radiobutton" value="myValue1" />
   male<br>
   <input type="radio" name="radiobutton" value="myValue2”
          checked="checked" />female
```



- If two or more radio buttons have the same name, the user can only select one of them at a time
  - This is how you make a radio button “group”
- If you ask for the value of that name, you will get the value specified for the selected radio button
- As with checkboxes, radio buttons do not contain any text

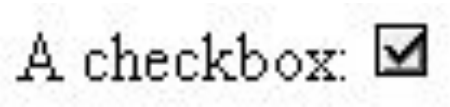
Labels

- In many cases, the labels for controls are not part of the control
  - <input type="radio" name="gender" value="m" />male
  - In this case, clicking on the word "male" has no effect
- A label tag will bind the text to the control
  - <label><input type="radio" name="gender" value="m" />male</label>
  - Clicking on the word "male" now clicks the radio button
- w3schools says that you should use the for attribute:
  - <label for="lname">Last Name:</label>
    <input type="text" name="lastname" id="lname" />
  - In my testing (Firefox and Opera), this isn't necessary, but it may be for some browsers
- Labels also help page readers read the page correctly
- Some browsers may render labels differently

Checkboxes

- A checkbox:

```
<input type="checkbox" name="checkbox"
       value="checkbox" checked="checked">
```

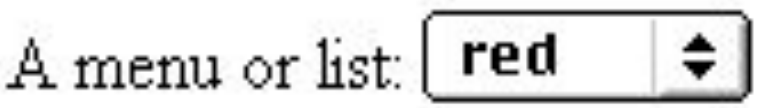


  - type: "checkbox"
  - name: used to reference this form element from JavaScript
  - value: value to be returned when element is checked
  - Note that there is *no text* associated with the checkbox
    - Unless you use a label tag, only clicking on the box itself has any effect

Drop-down menu or list

- A menu or list:

```
<select name="select">
    <option value="red">red</option>
    <option value="green">green</option>
    <option value="BLUE">blue</option>
</select>
```



- Additional arguments:
  - size: the number of items visible in the list (default is "1")
  - multiple
    - if set to "true" (or just about anything else), any number of items may be selected
    - if omitted, only one item may be selected
    - if set to "false", behavior depends on the particular browser

Hidden fields

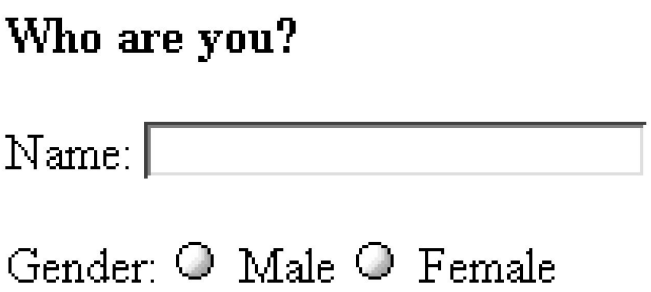
- <input type="hidden" name="hiddenField" value="nyah">
  <-- right there, don't you see it?

A hidden field: <-- right there, don't you see it?

- What good is this?
  - All input fields are sent back to the server, including hidden fields
  - This is a way to include information that the user doesn't need to see (or that you don't want her to see)
  - The value of a hidden field can be set programmatically (by JavaScript) before the form is submitted

A complete example

```
<html>
<head>
<title>Get Identity</title>
<meta http-equiv="Content-Type" content="text/html;
        charset=iso-8859-1">
</head>
<body>
<p><b>Who are you?</b></p>
<form method="post" action="">
  <p>Name:
    <input type="text" name="textfield">
  </p>
  <p>Gender:
    <label><input type="radio" name="gender" value="m" />Male<label>
    <label><input type="radio" name="gender" value="f" />Female</label>
  </p>
  </form>
</body>
</html>
```



Thank you for your attention!