



CS472 WAP

Input Widgets

Forms

Connecting with the Source

Except where otherwise noted, the contents of this document are Copyright 2012 Marty Stepp, Jessica Miller, Victoria Kirst and Roy McElmurry IV. All rights reserved. Any redistribution, reproduction, transmission, or storage of part or all of the contents in any form is prohibited without the author's expressed written permission. Slides have been modified for Maharishi University of Management Computer Science course CS472 in accordance with instructors agreement with authors.

**Maharishi International University Fairfield,
Iowa** © 2020



All rights reserved. No part of this slide presentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or recording, or by any information storage and retrieval system, without permission in writing from Maharishi International University.

Main Point Preview

- HTML provides many different types of input widgets, including text fields, text areas, check boxes, radio buttons, and dropdown lists, this is also an area HTML 5 is expanding to make form filling more efficient and effortless.
- *Nature supports the growth of things that are efficient. Do less and accomplish more.*

Form controls: <input>



```
<input type="text" name="q" value="Colbert Report" />
```

```
<input type="submit" value="Booyah!" />
```







- ▶ `input` element is used to create many UI controls
 - ▶ an inline element that **MUST** be self-closed
- ▶ `name` attribute specifies name/key of query parameter to pass to server
- ▶ `type` **can be** button, checkbox, file, hidden, password, radio, reset, submit, text, ...
- ▶ `value` attribute specifies control's initial text

Text fields: <input>



```
<input type="text" name="username" size="10" maxlength="8" />
```

```
<input type="text" name="password" size="8" />
```

Attribute	Value(s)	Description
value	text	Initial text to appear in text box
size	integer	Visible length of text box, in characters
maxlength	integer	Maximum number of chars that may be typed into text box
autocomplete 	on, off	Whether to offer suggestions of text to auto-complete the field
autofocus 	autofocus	Makes control initially receive keyboard focus
novalidate 	novalidate	Indicates browser should not check value before submitting
placeholder 	text	A hint or example of what the user should type;
pattern 	regular expr.	A regular expression indicating what input is valid
required 	required	Whether browser should display an error if blank

Checkboxes



- ▶ yes/no choices that can be checked and unchecked (**inline**)
 - ▶ none, 1, or many checkboxes can be checked at same time
 - ▶ Use the **checked** attribute in HTML to initially check the box

```
<input type="checkbox" name="condiment" /> Lettuce
```

```
<input type="checkbox" name="condiment" checked /> Tomato
```

```
<input type="checkbox" name="condiment" checked /> Pickles
```

☐ Lettuce ☒ Tomato ☒ Pickles

Radio buttons



► Sets of mutually exclusive choices (**inline**)

- Grouped by **name** attribute (only one can be checked at a time)
- Must specify a **value** for each one or else it will be sent as value **on**

```
<input type="radio" name="cc" value="visa" checked /> Visa
```

```
<input type="radio" name="cc" value="mastercard" /> MasterCard
```

```
<input type="radio" name="cc" value="amex" /> American Express
```

☒ Visa ☐ MasterCard ☐ American Express

<textarea>

- ▶ The <**textarea**> tag defines a multi-line text input control. (inline)
- ▶ holds unlimited number of characters
- ▶ text renders in a fixed-width font (usually Courier).
- ▶ size specified by **cols** and **rows** attributes
 - ▶ Or better, through CSS **height** and **width** properties.
 - ▶ `textarea { width: 300px; height: 200px }`

```
<textarea rows="4" cols="20">
```

Type your comments here.

```
</textarea>
```

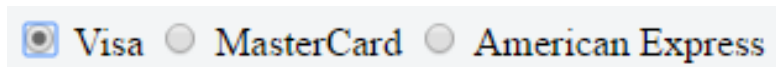
See example: `lecture4_examples/textarea.html`

Text labels: `<label>`



- ▶ Associates nearby text with control, so you can:
 - **click text to activate control**
- ▶ Can be used with **checkboxes** or **radio** buttons
- ▶ **label** element can be targeted by CSS style rules

```
<label> <input type="radio" name="cc" value="visa" checked="checked" /> Visa</label>  
<label> <input type="radio" name="cc" value="mastercard" /> MasterCard</label>  
<label> <input type="radio" name="cc" value="amex" /> American Express</label>
```



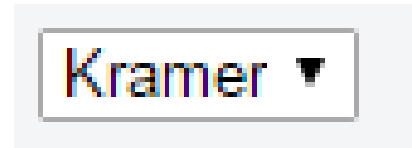
- ▶ See example: [lecture4_examples/label1.html](#), [lecture4_examples/label2.html](#)

Drop-down list `<select>` and `<option>`



- ▶ Menus of choices that collapse and expand (inline)
 - ▶ **option** element represents each choice
 - ▶ **select** optional attributes: **disabled**, **multiple**, **size**
 - ▶ optional **selected** attribute sets which one is initially chosen

```
<select name="favoritecharacter">  
  <option>Jerry</option>  
  <option>George</option>  
  <option selected>Kramer</option>  
  <option>Elaine</option>  
</select>
```

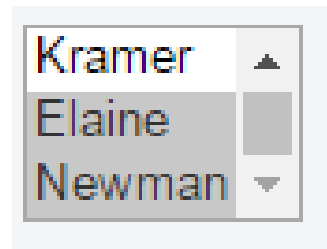




Multiple <select>

- ▶ optional **multiple** attribute allows selecting multiple items with shift- or ctrl- click
- ▶ **option** tags can be set to be initially **selected**

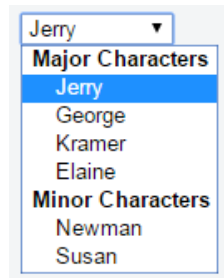
```
<select    name="favoritecharacter"    size="3"    multiple>  
  <option>Jerry</option>  
  <option>George</option>  
  <option>Kramer</option>  
  <option>Elaine</option>  
  <option selected>Newman</option>  
</select>
```



Option groups: <optgroup>



```
<select name="favoritecharacter">
  <optgroup label="Major Characters">
    <option>Jerry</option>
    <option>George</option>
    <option>Kramer</option>
    <option>Elaine</option>
  </optgroup>
  <optgroup label="Minor Characters">
    <option>Newman</option>
    <option>Susan</option>
  </optgroup>
</select>
```

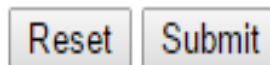


Reset and Submit buttons

- ▶ When we click **reset** button, it returns all form controls to their initial values
- ▶ When we click **submit** buttons, it sends all data with the specified **method** (Get/Post) to the **action** page in the form
- ▶ Specify custom text on the button by setting its **value** attribute

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

```
<input type="submit" />
```



Hidden input parameters



- ▶ An invisible parameter that is still passed to the server when form is submitted, it's useful for passing on additional state that isn't modified by the user

```
<input type="text" name="username" /> Name
```

```
<br />
```

```
<input type="text" name="sid" /> SID
```

```
<br />
```

```
<input type="hidden" name="school" value="MUM" />
```

```
<input type="hidden" name="year" value="2048" />
```

<input type="text"/>	Name
<input type="text"/>	SID



Grouping `<fieldset>`, `<legend>`

- ▶ Groups of input fields with optional caption (legend)

```
<fieldset>
  <legend>Credit cards:</legend>
  <input type="radio" name="cc" value="visa" checked="checked" /> Visa
  <input type="radio" name="cc" value="mastercard" /> MasterCard
  <input type="radio" name="cc" value="amex" /> American Express
</fieldset>
```

Credit cards: ☒ Visa ☐ MasterCard ☐ American Express

New Form Controls in HTML5



Input type	Description
color	A color from a palette of available choices
range	A slider for selecting values in a given range
date	A date such as August 29, 2016
time	A time of day such as 11:15 PM
datetime	A date and time such as 11:15 PM, August 29, 2016
month	A month of a particular year, such as August, 2016
week	A week of a particular year, such as August 35, 2016

Styling forms – attribute selector



► Because most input elements are created using input tag, we can target groups of elements using this CSS selector:

```
element[attribute="value"] {  
  property: value;  
  property: value;  
  ... property: value;  
}  
  
input[type="text"] {  
  background-color: yellow;  
  font-weight: bold;  
}
```

CS472

Main Point

- HTML provides many different types of input widgets, including text fields, text areas, check boxes, radio buttons, and dropdown lists, this is also an area HTML 5 is expanding to make form filling more efficient and effortless.
- *Nature supports the growth of things that are efficient. Do less and accomplish more.*

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

HTML Forms: Connecting with the Source

1. Forms let us submit data to the web server, which can then generate a custom response based on server side information.
2. GET requests are intended to only retrieve information and should be idempotent. POST requests are intended to submit data and not request a direct response.

3. Transcendental consciousness is the experience of the source of thought.

4.Impulses within the Transcendental field: inputs and perceptions are appreciated in their full value at the quiet levels of the mind.

5.Wholeness moving within itself: In Unity Consciousness one appreciates the full value of all inputs and perceptions. Everything is appreciated as some expression of Totality.



