

HTML Tables and Forms

Chapter 4

Objectives

1

Introducing **Tables**

2

Styling Tables

3

Introducing **Forms**

4

Form Control
Elements

5

Table and Form
Accessibility

6

Microformats

Section 1 of 6

INTRODUCING TABLES

HTML Tables

A grid of cells

A [table](#) in HTML is created using the **<table>** element

Tables can be used to display:

- Many types of content
 - Calendars, financial data, lists, etc...
 - Any type of data
 - Images
 - Text
 - Links
 - Other tables
-

HTML Tables




Example usages

Chapter 4

	Free	Basic	Premium
Upload Space	50MB	200MB	Unlimited
Daily Uploads	1	10	Unlimited
Total Uploads	20	100	Unlimited
Social Sharing		✓	✓
Analytics			✓
Price per year	Free	\$ 9.99	\$ 19.99

Chapter 4

Artist Inventory

Artist	Work Details		
	Title	Year	Home
 Jacques-Louis David	 The Death of Marat	1793	Royal Museums of Fine Arts of Belgium
	 The Intervention of the Sabine Women	1793	Royal Museums of Fine Arts of Belgium

Chapter 4

October 2014						
S	M	T	W	T	F	S
			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	
« Sep					Nov »	

Paintings

Title	Artist	Year	Genre
 <i>Death of Marat</i>	David, Jacques-Louis	1793	Romanticism
 <i>Lictors Bearing to Brutus the Bodies of his Sons</i>	David, Jacques-Louis	1789	Romanticism
 <i>Liberty Leading the People</i>	Delacroix, Eugene	1830	Romanticism
 <i>Arrangement in Grey and Black</i>	Whistler, James Abbott	1871	Realism
 <i>Mademoiselle Caroline Riviere</i>	Ingres, Jean-Auguste	1806	Neo-Classicism

Tables Basics

Rows and cells

- an HTML **<table>** contains any number of rows (**<tr>**)
- each row contains any number of table data cells (**<td>**)
- Content goes inside of **<td></td>** tags

<table>

<tr>

<td>The Death of Marat**</td>**

</tr>

</table>



content

A basic Example

`<table>`

<code><tr></code>	The Death of Marat <code><td></code>	Jacques-Louis David <code><td></code>	1793 <code><td></code>	162cm <code><td></code>	128cm <code><td></code>
<code><tr></code>	Burial at Ornans <code><td></code>	Gustave Courbet <code><td></code>	1849 <code><td></code>	314cm <code><td></code>	663cm <code><td></code>

`<table>`

`<tr>`

`<td>The Death of Marat</td>`
`<td>Jacques-Louis David</td>`
`<td>1793</td>`
`<td>162cm</td>`
`<td>128cm</td>`

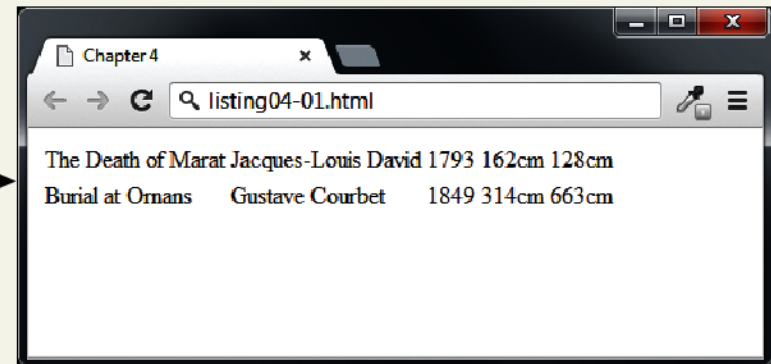
`</tr>`

`<tr>`

`<td>Burial at Ornans</td>`
`<td>Gustave Courbet</td>`
`<td>1849</td>`
`<td>314cm</td>`
`<td>663cm</td>`

`</tr>`

`</table>`

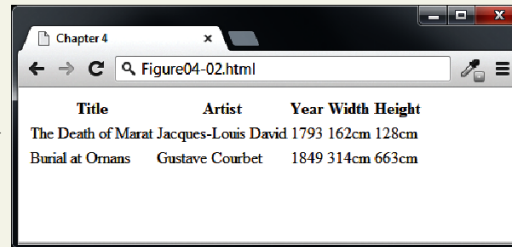


With Table Headings

<table>				
<tr>	Title	Artist	Year	Width
	<th>	<th>	<th>	<th>
<tr>	The Death of Marat	Jacques-Louis David	1793	162cm
	<td>	<td>	<td>	<td>
<tr>	Burial at Ornans	Gustave Courbet	1849	314cm
	<td>	<td>	<td>	<td>

th

```
<table>
  <tr>
    <th>Title</th>
    <th>Artist</th>
    <th>Year</th>
    <th>Width</th>
    <th>Height</th>
  </tr>
  <tr>
    <td>The Death of Marat</td>
    <td>Jacques-Louis David</td>
    <td>1793</td>
    <td>162cm</td>
    <td>128cm</td>
  </tr>
  <tr>
    <td>Burial at Ornans</td>
    <td>Gustave Courbet</td>
    <td>1849</td>
    <td>314cm</td>
    <td>663cm</td>
  </tr>
</table>
```



Title	Artist	Year	Width	Height
The Death of Marat	Jacques-Louis David	1793	162cm	128cm
Burial at Ornans	Gustave Courbet	1849	314cm	663cm

Why Table Headings

A table heading `<th>`

- Browsers tend to make the content within a `<th>` element bold
 - `<th>` element for accessibility (it helps those using screen readers)
 - Provides some semantic info about the row being a row of headers
-

Spanning Rows and Columns

Span Span Span a Row

Each row must have the same number of `<td>` or `<th>` containers. If you want a given cell to cover several columns or rows,

<table>				
<tr>				
<th>		<th colspan=2>		
<th>				
<td>		<td>		
<td>				
<td>		<td>		

Notice that this row now only has four cell elements.

```
<table>
<tr>
  <th>Title</th>
  <th>Artist</th>
  <th>Year</th>
  <th colspan="2">Size (width x height)</th>
</tr>
<tr>
  <td>The Death of Marat</td>
  <td>Jacques-Louis David</td>
  <td>1793</td>
  <td>162cm</td>
  <td>128cm</td>
</tr>
...
</table>
```

use the **colspan** or **rowspan** attributes

Using Tables for Layout

It works in many situations

- Popular in 1990s
- Results in table bloat
- Not semantic
- Larger HTML pages
- Browser quirks

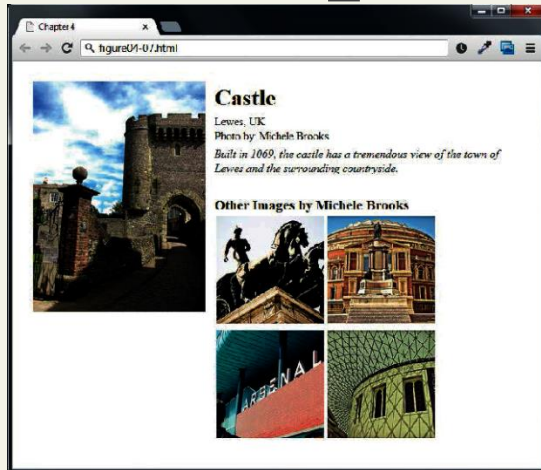
`<table>`

Artist	Title	Year
Jacques-Louis David	The Death of Marat	1793
	The Intervention of the Sabine Women	1799
	Napoleon Crossing the Alps	1800

```
<table>
<tr>
  <th>Artist</th>
  <th>Title</th>
  <th>Year</th>
</tr>
<tr>
  <td rowspan="3">Jacques-Louis David</td>
  <td>The Death of Marat</td>
  <td>1793</td>
</tr>
<tr>
  <td>The Intervention of the Sabine Women</td>
  <td>1799</td>
</tr>
<tr>
  <td>Napoleon Crossing the Alps</td>
  <td>1800</td>
</tr>
...
</table>
```

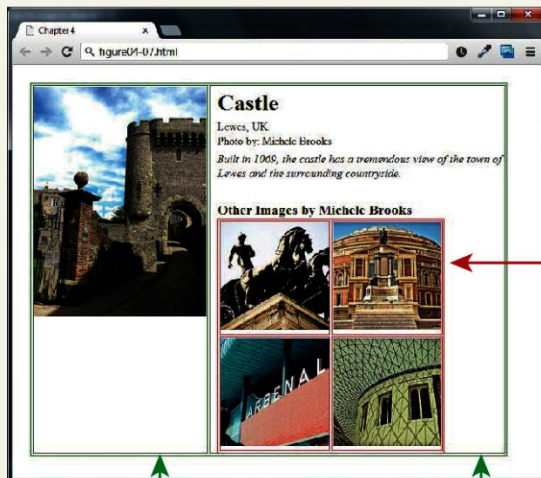
Notice that these two rows now only have two cell elements.

Example Table layouts



```
<table>
<tr>
  <td>
    
  </td>
  <td>
    <h2>Castle</h2>
    <p>Lewes, UK</p>
    <p>Photo by: Michele Brooks</p>
    <p>Built in 1069, the castle has a tremendous
      view of the town of Lewes and the
      surrounding countryside.
    </p>
  </td>
</tr>
</table>
```

<h3>Other Images by Michele Brooks</h3>



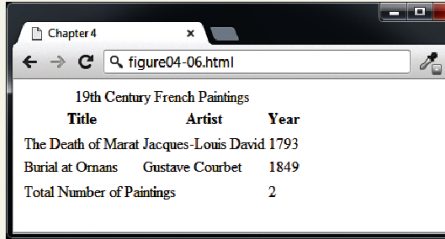
```
<table>
<tr>
  <td></td>
  <td></td>
</tr>
<tr>
  <td></td>
  <td></td>
</tr>
</table>
```

Additional table tags

- `<caption>`
A title for the table is good for accessibility.
 - `<col>`, `<colgroup>`
These describe our columns, and can be used to aid in styling.
 - `<thead>`
Table header could potentially also include other `<tr>` elements.
 - `<tfoot>`
Yes, the table footer comes *before* the body.
 - `<tbody>`
Potentially, with styling the browser can scroll this information, while keeping the header and footer fixed in place.
- ```
<table>
 <caption>19th Century French Paintings</caption>
 <col class="artistName" />
 <colgroup id="paintingColumns">
 <col />
 <col />
 </colgroup>

 <thead>
 <tr>
 <th>Title</th>
 <th>Artist</th>
 <th>Year</th>
 </tr>
 </thead>

 <tfoot>
 <tr>
 <td colspan="2">Total Number of Paintings</td>
 <td>2</td>
 </tr>
 </tfoot>

 <tbody>
 <tr>
 <td>The Death of Marat</td>
 <td>Jacques-Louis David</td>
 <td>1793</td>
 </tr>
 <tr>
 <td>Burial at Ornans</td>
 <td>Gustave Courbet</td>
 <td>1849</td>
 </tr>
 </tbody>
</table>
```
- 

Section 2 of 6

# STYLING TABLES

---

# Styling Tables

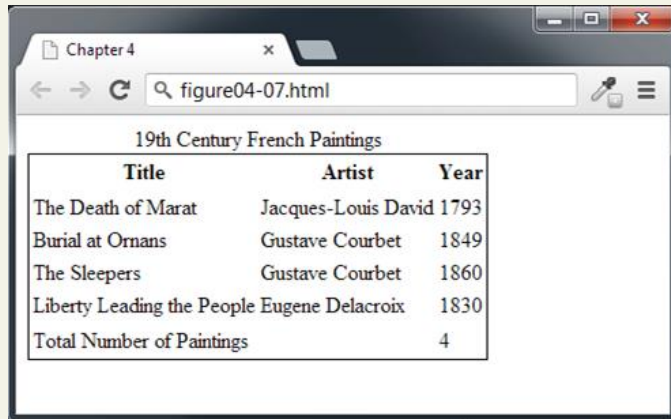
The old way's deprecated

In HTML5 it is left to CSS, However legacy support for deprecated HTML attributes still exist

- **width, height**—for setting the width and height of cells
  - **cellspacing**—for adding space between every cell in the table
  - **cellpadding**—for adding space between the content of the cell and its border
  - **bgcolor**—for changing the background color of any table element
  - **background**—for adding a background image to any table element
  - **align**—for indicating the alignment of a table in relation to the surrounding container
-

# Styling Tables

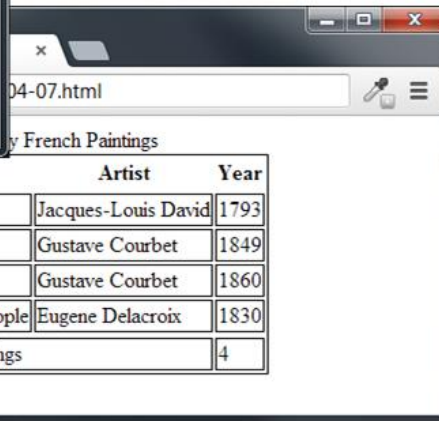
## Borders



A screenshot of a web browser window titled 'Chapter 4' showing a table with borders. The table is titled '19th Century French Paintings' and has three columns: Title, Artist, and Year. The data rows are: The Death of Marat (Jacques-Louis David, 1793), Burial at Ornans (Gustave Courbet, 1849), The Sleepers (Gustave Courbet, 1860), Liberty Leading the People (Eugene Delacroix, 1830), and Total Number of Paintings (4).

Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Total Number of Paintings		4

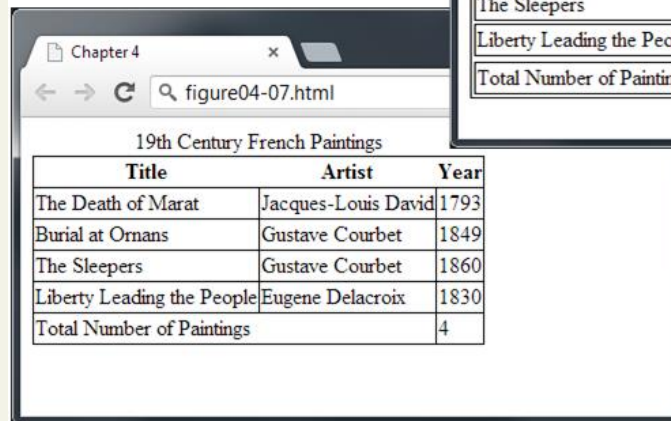
```
table {
 border: solid 1pt black;
}
```



A screenshot of a web browser window showing a table with borders. The table is titled '19th Century French Paintings' and has three columns: Title, Artist, and Year. The data rows are: The Death of Marat (Jacques-Louis David, 1793), Burial at Ornans (Gustave Courbet, 1849), The Sleepers (Gustave Courbet, 1860), Liberty Leading the People (Eugene Delacroix, 1830), and Total Number of Paintings (4).

Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Total Number of Paintings		4

```
table {
 border: solid 1pt black;
}
td {
 border: solid 1pt black;
}
```



A screenshot of a web browser window showing a table with borders. The table is titled '19th Century French Paintings' and has three columns: Title, Artist, and Year. The data rows are: The Death of Marat (Jacques-Louis David, 1793), Burial at Ornans (Gustave Courbet, 1849), The Sleepers (Gustave Courbet, 1860), Liberty Leading the People (Eugene Delacroix, 1830), and Total Number of Paintings (4).

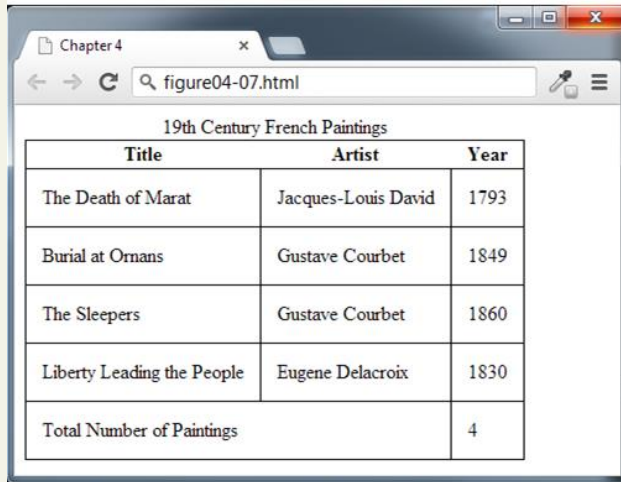
Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Total Number of Paintings		4

```
table {
 border: solid 1pt black;
 border-collapse: collapse;
}
td {
 border: solid 1pt black;
}
```



# Styling Tables

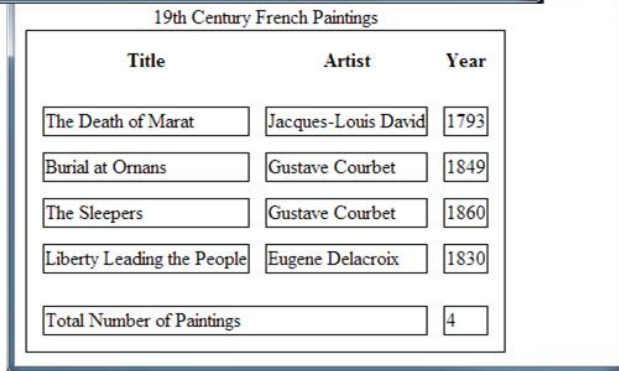
## Padding and spacing



A screenshot of a web browser window titled 'Chapter 4' showing a file named 'figure04-07.html'. The browser displays a table titled '19th Century French Paintings'. The table has three columns: 'Title', 'Artist', and 'Year'. The data rows are: 'The Death of Marat' by Jacques-Louis David in 1793, 'Burial at Ornans' by Gustave Courbet in 1849, 'The Sleepers' by Gustave Courbet in 1860, and 'Liberty Leading the People' by Eugene Delacroix in 1830. A final row shows 'Total Number of Paintings' as 4. The table cells have no padding, making the text appear cramped.

Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Total Number of Paintings		4

```
table {
 border: solid 1pt black;
 border-collapse: collapse;
}
td {
 border: solid 1pt black;
 padding: 10pt;
}
```



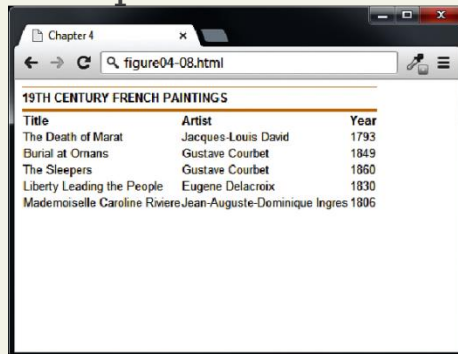
A screenshot of a web browser window showing the same table as the previous image, but with padding applied. The text is now clearly separated from the cell borders, making the table much more readable.

Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Total Number of Paintings		4

```
table {
 border: solid 1pt black;
 border-spacing: 10pt;
}
td {
 border: solid 1pt black;
}
```

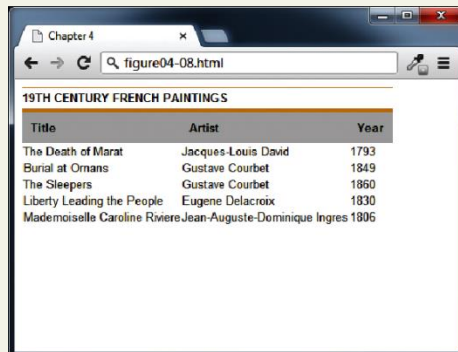
# Styling Tables

## Examples



Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

```
table {
 font-size: 0.8em;
 font-family: Arial, Helvetica, sans-serif;
 border-collapse: collapse;
 border-top: 4px solid #DCA806;
 border-bottom: 1px solid white;
 text-align: left;
}
caption {
 font-weight: bold;
 padding: 0.25em 0 0.25em 0;
 text-align: left;
 text-transform: uppercase;
 border-top: 1px solid #DCA806;
}
```



Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

```
thead tr {
 background-color: #CACACA;
}
th {
 padding: 0.75em;
}
```

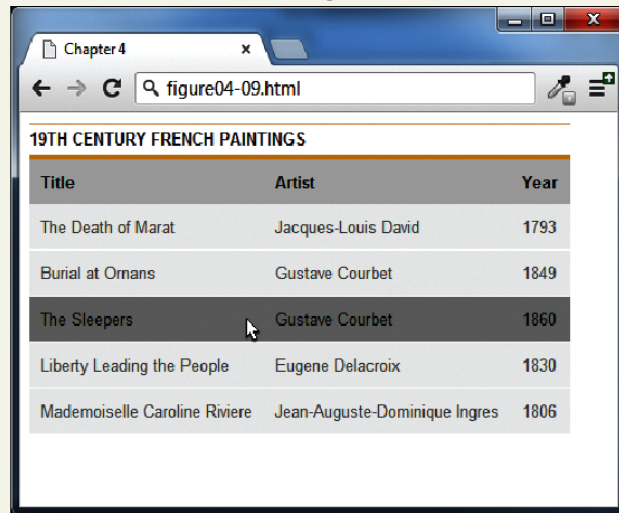


Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Ornans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

```
tbody tr {
 background-color: #F1F1F1;
 border-bottom: 1px solid white;
 color: #6E6E6E;
}
tbody td {
 padding: 0.75em;
}
```

# Nth-Child

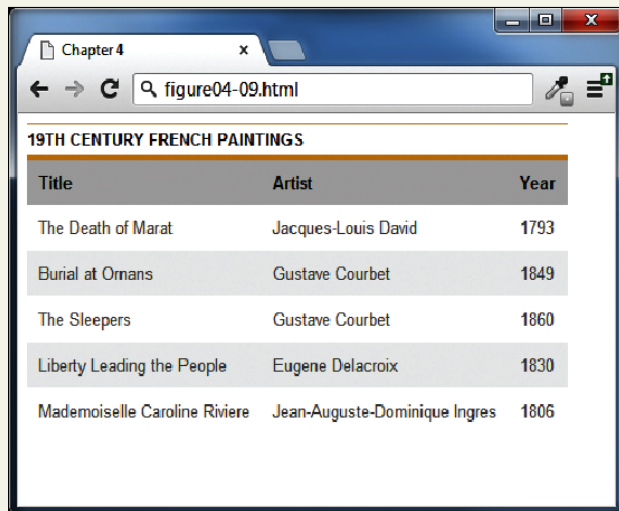
Nifty Table styling tricks: hover effect and zebra-stripes



A screenshot of a web browser window titled 'Chapter4' showing a table of 19th-century French paintings. The table has three columns: Title, Artist, and Year. The row for 'The Sleepers' by Gustave Courbet is highlighted with a dark background, indicating a hover effect.

Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Omans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

```
tbody tr:hover {
 background-color: #9e9e9e;
 color: black;
}
```



A screenshot of a web browser window titled 'Chapter4' showing the same table of 19th-century French paintings. The table has three columns: Title, Artist, and Year. The rows alternate between white and light gray backgrounds, demonstrating a zebra-stripe effect.

Title	Artist	Year
The Death of Marat	Jacques-Louis David	1793
Burial at Omans	Gustave Courbet	1849
The Sleepers	Gustave Courbet	1860
Liberty Leading the People	Eugene Delacroix	1830
Mademoiselle Caroline Riviere	Jean-Auguste-Dominique Ingres	1806

```
tbody tr:nth-child(odd) {
 background-color: white;
}
```

Section 3 of 6

# INTRODUCING FORMS

---

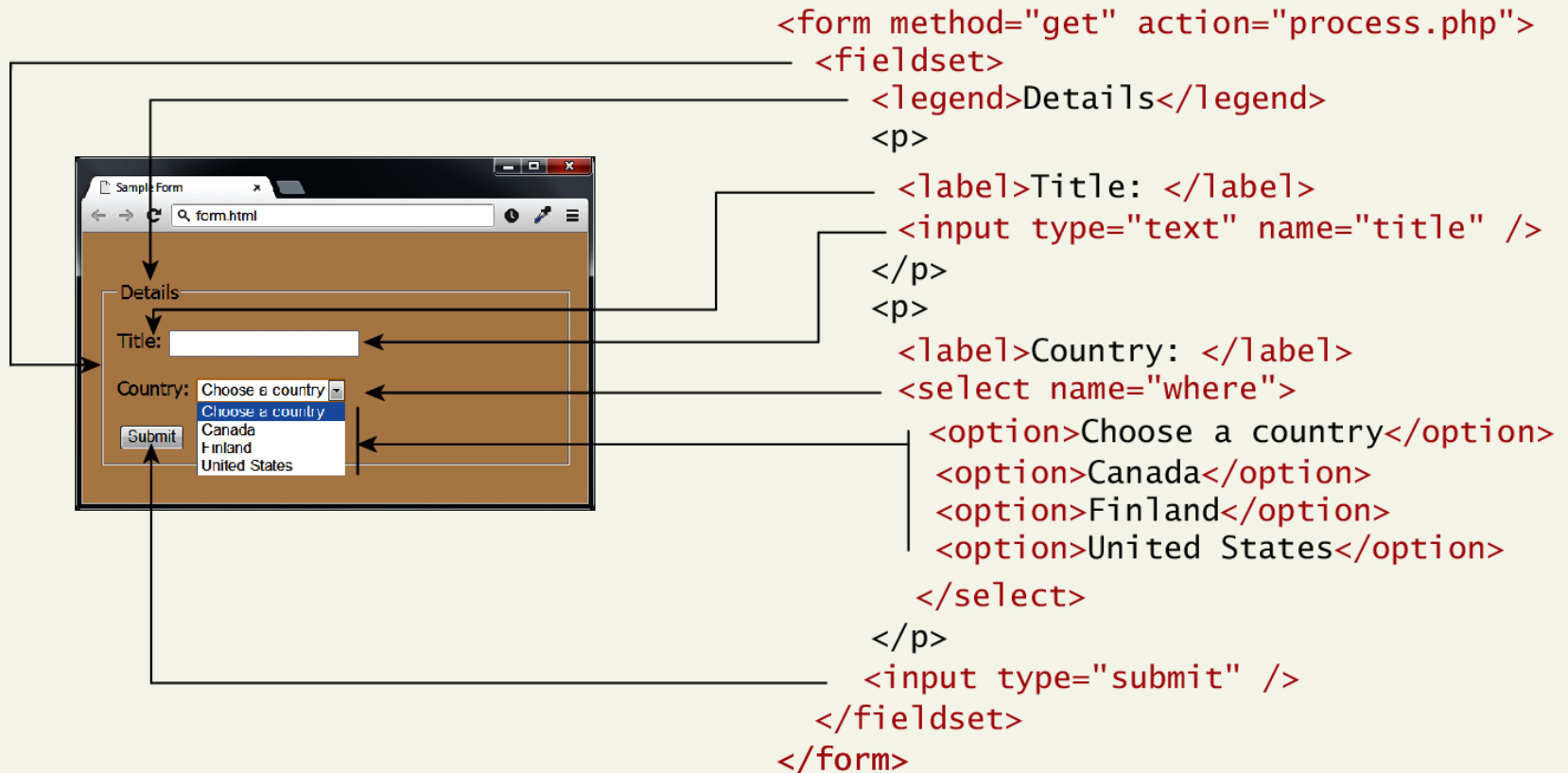
# HTML Forms

Richer way to interact with server

**Forms** provide the user with an alternative way to interact with a web server.

- Forms provide rich mechanisms like:
    - Text input
    - Password input
    - Options Lists
    - Radio and check boxes
-

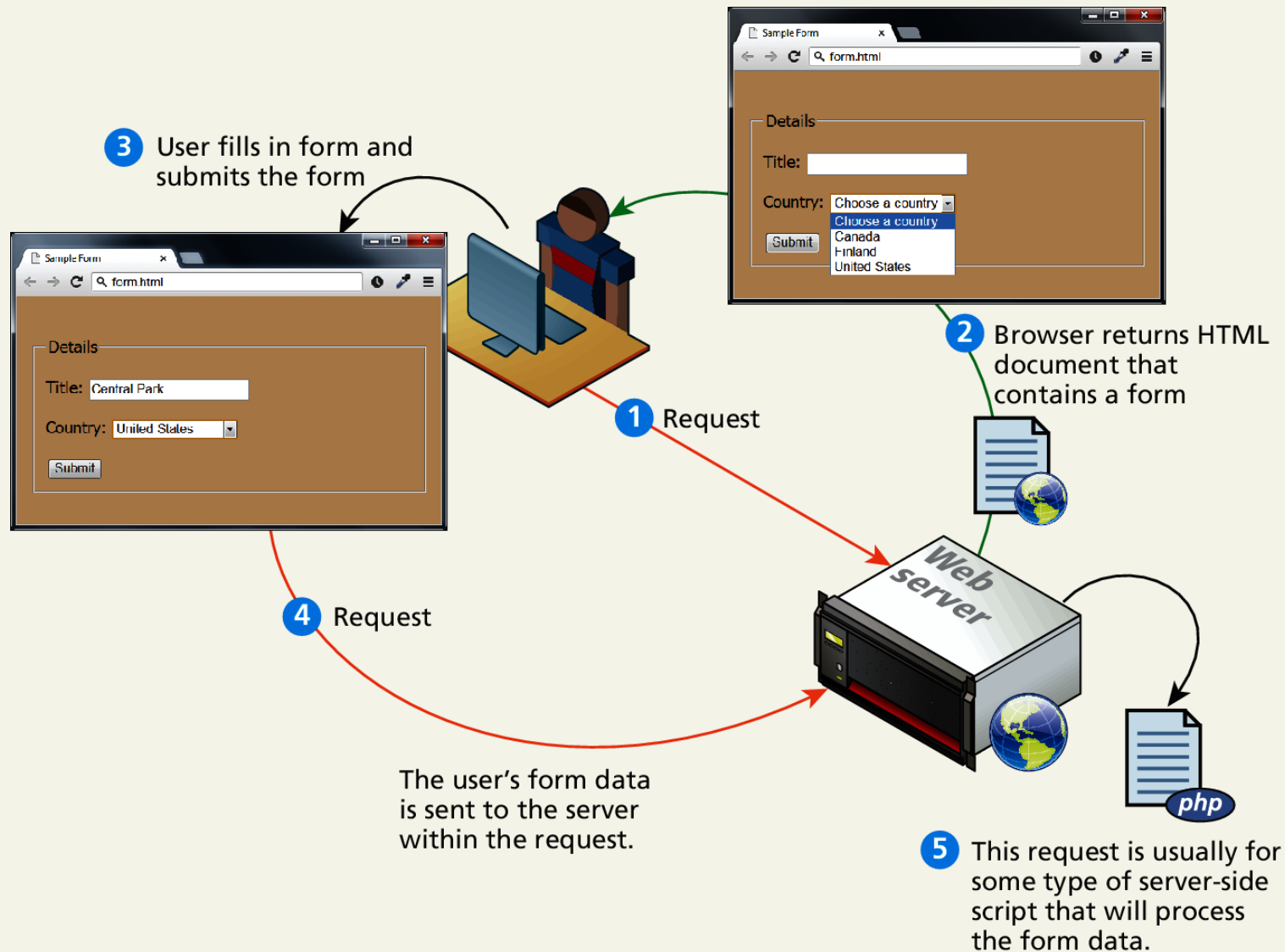
# Form Structure



```
<form method="get" action="process.php">
 <fieldset>
 <legend>Details</legend>
 <p>
 <label>Title: </label>
 <input type="text" name="title" />
 </p>
 <p>
 <label>Country: </label>
 <select name="where">
 <option>Choose a country</option>
 <option>Canada</option>
 <option>Finland</option>
 <option>United States</option>
 </select>
 </p>
 <input type="submit" />
 </fieldset>
</form>
```

The diagram illustrates the structure of an HTML form. On the left, a browser window titled 'Sample Form' displays a form with a 'Details' section. This section contains a 'Title' text input, a 'Country' dropdown menu with options 'Choose a country', 'Canada', 'Finland', and 'United States', and a 'Submit' button. Arrows point from the form elements to the corresponding HTML code on the right. The code defines the form's method and action, creates a fieldset with a legend, and uses labels and input tags to structure the form fields and the submit button.

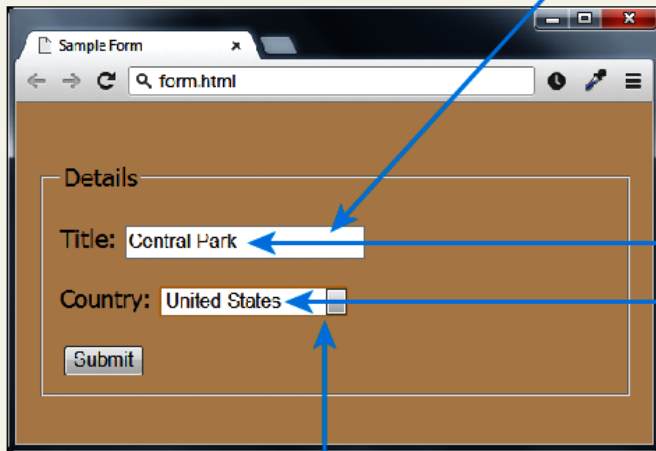
# How forms interact with servers



# Query Strings

At the end of the day, another string

```
<input type="text" name="title" />
```



A screenshot of a web browser window titled 'Sample Form' showing a form with a 'Details' section. The 'Title' field is a text input containing 'Central Park'. The 'Country' field is a dropdown menu showing 'United States'. A 'Submit' button is at the bottom. Blue arrows point from the 'title' attribute in the code above to the 'Title' input field, and from the 'where' attribute in the code below to the 'Country' dropdown field. A blue line connects the values of both fields to the query string on the right.

title=Central+Park&where=United+States

```
<select name="where">
```



# URL encoding

Special symbols

**Browser**

← →

Artist:

*Notice how the spaces and the accented é are URL encoded (in red).*

artist=Pablo+Jos%E9+Picasso

URL Encoding

# <form> element

Two essential features of any form, namely the **action** and the **method** attributes.

- The **action** attribute specifies the URL of the server-side resource that will process the form data
  - The **method** attribute specifies how the query string data will be transmitted from the browser to the server.
    - GET
    - POST
-

# GET vs POST

A screenshot of a web browser window titled 'Sample Form'. The address bar shows 'form.html'. The form contains a 'Details' section with a 'Title' input field containing 'Central Park' and a 'Country' dropdown menu set to 'United States'. A 'Submit' button is located below the form fields.

`<form method="get" action="process.php">`

`GET /process.php?title=Central+Park&where=United+States http/1.1`

querystring

`<form method="post" action="process.php">`

```
POST /process.php http/1.1
Date: Sun, 20 May 2012 23:59:59 GMT
Host: www.mysite.com
User-Agent: Mozilla/4.0
Content-Length: 47

title=Central+Park&where=United+States
```

HTTP Header

querystring

# GET vs POST

## Advantages and Disadvantages

- Data can be clearly seen in the address bar.
- Data remains in browser history and cache.
- Data can be bookmarked
- Limit on the number of characters in the form data returned.

## POST

- Data can contain binary data.
  - Data is hidden from user.
  - Submitted data is not stored in cache, history, or bookmarks.
-

Section 4 of 6

# FORMS CONTROL ELEMENTS

---

# Form-Related HTML Elements

Type	Description
<code>&lt;button&gt;</code>	Defines a clickable button.
<code>&lt;datalist&gt;</code>	An HTML5 element form defines lists to be used with other form elements.
<code>&lt;fieldset&gt;</code>	Groups related elements in a form together.
<code>&lt;form&gt;</code>	Defines the form container.
<code>&lt;input&gt;</code>	Defines an input field. HTML5 defines over 20 different types of input.
<code>&lt;label&gt;</code>	Defines a label for a form input element.
<code>&lt;legend&gt;</code>	Defines the label for a fieldset group.
<code>&lt;option&gt;</code>	Defines an option in a multi-item list.
<code>&lt;optgroup&gt;</code>	Defines a group of related options in a multi-item list.
<code>&lt;select&gt;</code>	Defines a multi-item list.
<code>&lt;textarea&gt;</code>	Defines a multiline text entry box.

# Text Input Controls

Type	Description
text	Creates a single line text entry box. <code>&lt;input type="text" name="title" /&gt;</code>
textarea	Creates a multiline text entry box. <code>&lt;textarea rows="3" ... /&gt;</code>
password	Creates a single line text entry box for a password <code>&lt;input type="password" ... /&gt;</code>
search	Creates a single-line text entry box suitable for a search string. This is an HTML5 element. <code>&lt;input type="search" ... /&gt;</code>
email	Creates a single-line text entry box suitable for entering an email address. This is an HTML5 element. <code>&lt;input type="email" ... /&gt;</code>
tel	Creates a single-line text entry box suitable for entering a telephone. This is an HTML5 element. <code>&lt;input type="tel" ... /&gt;</code>
url	Creates a single-line text entry box suitable for entering a URL. This is an HTML5 element. <code>&lt;input type="url" ... /&gt;</code>

# Text Input Controls

## Classic

```
<input type="text" ... />
```

Text:

```
<textarea>
 enter some text
</textarea>
```

TextArea:

```
<textarea placeholder="enter some text">
</textarea>
```

TextArea:

```
<input type="password" ... />
```

Password:  Password:



# Text Input Controls

## HTML5

```
<input type="search" placeholder="enter search text" ... />
```

Search:       Search:

```
<input type="email" ... />
```

Email:  *In Opera*

Please enter a valid email address

Email:  *In Chrome*

! Please enter an email address.

```
<input type="url" ... />
```

url:    
! Please enter a URL.

```
<input type="tel" ... />
```

Tel:

# HTML5 advanced controls

## Pattern attribute

```
<input type="text" ... placeholder="L#L #L#" pattern="[a-z][0-9][a-z] [0-9][a-z][0-9]" />
```

Postal:

Postal:

! Please match the requested format.

## datalist

Search City:

Paris  
Prague

```
<input type="text" name="city" list="cities" />

<datalist id="cities">
 <option>Calcutta</option>
 <option>Calgary</option>
 <option>London</option>
 <option>Los Angeles</option>
 <option>Paris</option>
 <option>Prague</option>
</datalist>
```

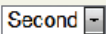
# Select Lists

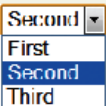
Chose an option, any option.

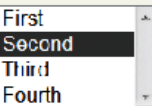
- **<select>** element is used to create a multiline box for selecting one or more items
    - The options are defined using the **<option>** element
    - can be hidden in a dropdown or multiple rows of the list can be visible
    - Option items can be grouped together via the **<optgroup>** element.
-

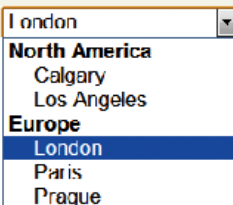
# Select Lists

## Select List Examples

Select: 

Select: 

Select: 

Cities: 

```
<select name="choices">
 <option>First</option>
 <option selected>Second</option>
 <option>Third</option>
</select>
```

```
<select size="3" ... >
```

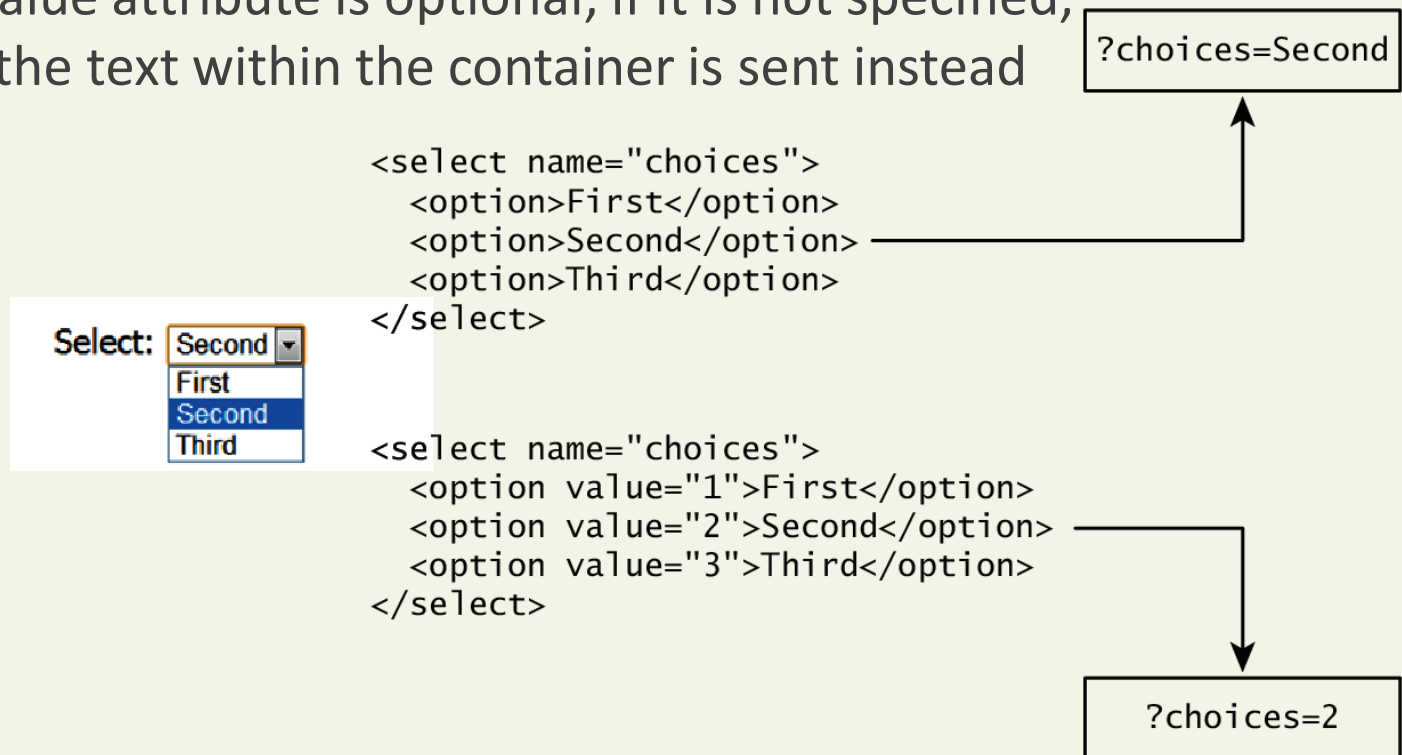
```
<select ... >
 <optgroup label="North America">
 <option>Calgary</option>
 <option>Los Angeles</option>
 </optgroup>
 <optgroup label="Europe">
 <option>London</option>
 <option>Paris</option>
 <option>Prague</option>
 </optgroup>
</select>
```

# Which Value to send

## Select Lists Cont.

The **value** attribute of the `<option>` element is used to specify what value will be sent back to the server.

The value attribute is optional; if it is not specified, then the text within the container is sent instead



# Radio Buttons

**Radio buttons** are useful when you want the user to select a single item from a small list of choices and you want all the choices to be visible

- radio buttons are added via the `<input type="radio">` element
  - The buttons are mutually exclusive (i.e., only one can be chosen) by sharing the same name attribute
  - The checked attribute is used to indicate the default choice
  - the value attribute works in the same manner as with the `<option>` element
-

# Radio Buttons

Continent:

- ☐ North America
- ☒ South America
- ☐ Asia

```
<input type="radio" name="where" value="1">North America

<input type="radio" name="where" value="2" checked>South America

<input type="radio" name="where" value="3">Asia
```

# Checkboxes

**Checkboxes** are used for getting yes/no or on/off responses from the user.

- checkboxes are added via **the `<input type="checkbox">` Element**
  - You can also group checkboxes together by having them share the same name attribute
  - Each checked checkbox will have its value sent to the server
  - Like with radio buttons, the checked attribute can be used to set the default value of a checkbox
-



# Checkboxes

I accept the software license ☒

```
<label>I accept the software license</label>
<input type="checkbox" name="accept" >
```

Where would you like to visit?

- ☒ Canada
- ☐ France
- ☒ Germany

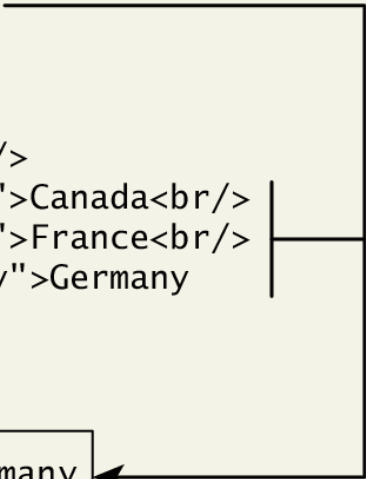
```
<label>Where would you like to visit? </label>

<input type="checkbox" name="visit" value="canada">Canada

<input type="checkbox" name="visit" value="france">France

<input type="checkbox" name="visit" value="germany">Germany
```

?accept=on&visit=canada&visit=germany

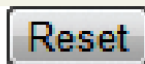
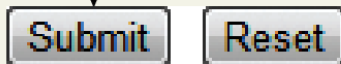


# Button Controls

Type	Description
<code>&lt;input type="submit"&gt;</code>	Creates a button that submits the form data to the server.
<code>&lt;input type="reset"&gt;</code>	Creates a button that clears any of the user's already entered form data.
<code>&lt;input type="button"&gt;</code>	Creates a custom button. This button may require Javascript for it to actually perform any action.
<code>&lt;input type="image"&gt;</code>	Creates a custom submit button that uses an image for its display.
<code>&lt;button&gt;</code>	<p>Creates a custom button. The <code>&lt;button&gt;</code> element differs from <code>&lt;input type="button"&gt;</code> in that you can completely customize what appears in the button; using it, you can, for instance, include both images and text, or skip server-side processing entirely by using hyperlinks.</p> <p>You can turn the button into a submit button by using the <code>type="submit"</code> attribute.</p>

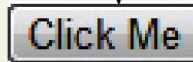
# Button Controls

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

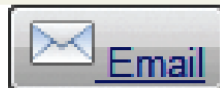
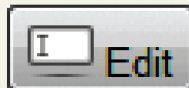


```
<input type="reset" />
```

```
<input type="button" value="Click Me" />
```



```
<input type="image" src="appointment.png" />
```



```
<button>

 Email

</button>
```

```
<button type="submit" >

 Edit
</button>
```

# Specialized Controls

I'm so special

- `<input type=hidden>`
- `<input type=file>`

Upload a travel photo  
 No file chosen



Upload a travel photo  
 IMG\_0020.JPG

```
<form method="post" enctype="multipart/form-data" ... >
 ...
 <label>Upload a travel photo</label>
 <input type="file" name="photo" />
 ...
</form>
```

# Number and Range

Typically input values need be **validated**. Although server side validation is required, optional client side pre-validation is good practice.

The number and range controls Added in HTML5 provide a way to input numeric values that **eliminates the need for JavaScript numeric validation!!!**

---

# Number and Range

Rate this photo:

2

Grumpy  Ecstatic

```
<label>Rate this photo:

```

```
<input type="number" min="1" max="5" name="rate" />
```

Grumpy

```
<input type="range" min="0" max="10" step="1" name="happiness" />
```

Ecstatic

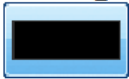
Rate this photo:

Grumpy  Ecstatic

Controls as they appear in browser  
that doesn't support these input types

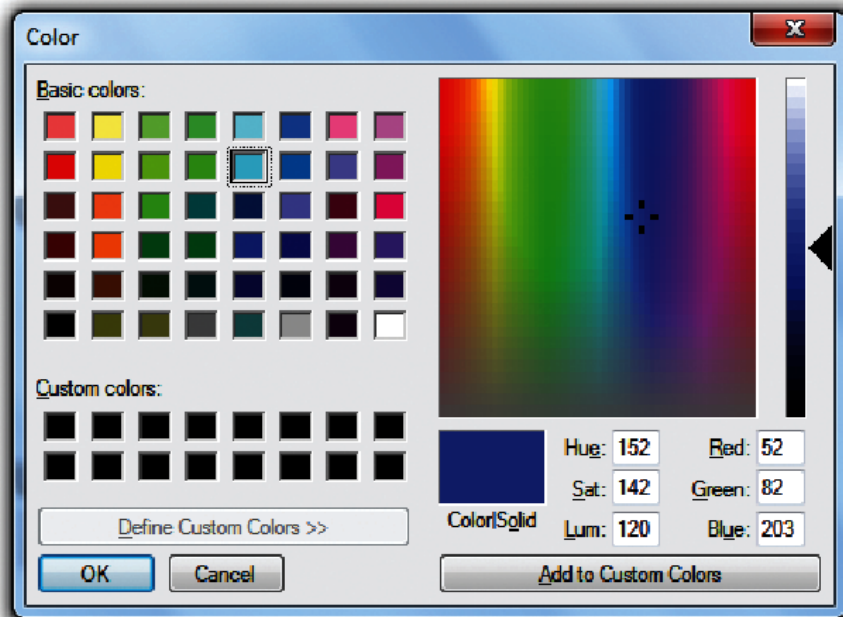
# Color

Background Color:



```
<label>Background Color:

<input type="color" name="back" />
```



Background Color:

Control as it appears in browser that  
doesn't support this input type

# Date and Time Controls

Dates and times often need validation when gathering this information from a regular text input control.

From a user's perspective, entering dates can be tricky as well: you probably have wondered at some point in time when entering a date into a web form, what format to enter it in, whether the day comes before the month, whether the month should be entered as an abbreviation or a number, and so on.

---



# HTML5 Date and Time Controls

Date:

March 2013

Mon	Tue	Wed	Thu	Fri	Sat	Sun
25	26	27	28	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
1	2	3	4	5	6	7

Today

```
<label>Date:

<input type="date" ... />
```

Time:

02:02 AM

```
<input type="time" ... />
```

DateTime:

2013-03-08 05:46 UTC

```
<input type="datetime" ... />
```

DateTime Local:

2013-03-13 12:02

```
<input type="datetime-local" ... />
```

---

# HTML5 Date and Time Controls

Month:

March, 2013

March, 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	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	1	2	3	4	5	6

This month Clear

```
<input type="month" ... />
```

Week:

2013-W10

March 2013

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

Today

```
<input type="week" ... />
```

# HTML Controls

Type	Description
<b>date</b>	Creates a general date input control. The format for the date is "yyyy-mm-dd".
<b>time</b>	Creates a time input control. The format for the time is "HH:MM:SS", for hours:minutes:seconds.
<b>datetime</b>	Creates a control in which the user can enter a date and time.
<b>datetime-local</b>	Creates a control in which the user can enter a date and time without specifying a time zone.
<b>month</b>	Creates a control in which the user can enter a month in a year. The format is "yyyy-mm".
<b>week</b>	Creates a control in which the user can specify a week in a year. The format is "yyyy-W##".

# Other Controls

You mean there's more

- The `<progress>` and `<meter>` elements can be used to provide feedback to users,
    - but requires JavaScript to function dynamically.
  - The `<output>` element can be used to hold the output from a calculation.
  - The `<keygen>` element can be used to hold a private key for public-key encryption
-

Section 5 of 6

# **TABLE AND FORM ACCESSIBILITY**

---

# Web Accessibility

Not all web users are able to view the content on web pages in the same manner.

The term **web accessibility** refers to the assistive technologies, various features of HTML that work with those technologies, and different coding and design practices that can make a site more usable for people with visual, mobility, auditory, and cognitive disabilities.

In order to improve the accessibility of websites, the W3C created the **Web Accessibility Initiative (WAI)**

- [Web Content Accessibility Guidelines](#)
-

# Web Content Accessibility Guidelines

- Provide text alternatives for any nontext content so that it can be changed into other forms people need, such as large print, braille, speech, symbols, or simpler language.
  - Create content that can be presented in different ways (for example simpler layout) without losing information or structure.
  - Make all functionality available from a keyboard.
  - Provide ways to help users navigate, find content, and determine where they are.
-

# Accessible Tables

1. Describe the table's content using the <caption> element
2. Connect the cells with a textual description in the header

```
<table>
 <caption>Famous Paintings</caption>
 <tr>
 <th scope="col">Title</th>
 <th scope="col">Artist</th>
 <th scope="col">Year</th>
 <th scope="col">Width</th>
 <th scope="col">Height</th>
 </tr>
 <tr>
 <td>The Death of Marat</td>
 <td>Jacques-Louis David</td>
 <td>1793</td>
```



# Accessible Forms

Recall the <fieldset>, <legend>, and <label> elements.

Each <label> element should be associated with a single input element.

```
<label for="f-title">Title: </label>
```

```
<input type="text" name="title" id="f-title"/>
```

```
<label for="f-country">Country: </label>
```

```
<select name="where" id="f-country">
 <option>Choose a country</option>
 <option>Canada</option>
 <option>Finland</option>
 <option>United States</option>
</select>
```

Section 6 of 6

# MICROFORMATS

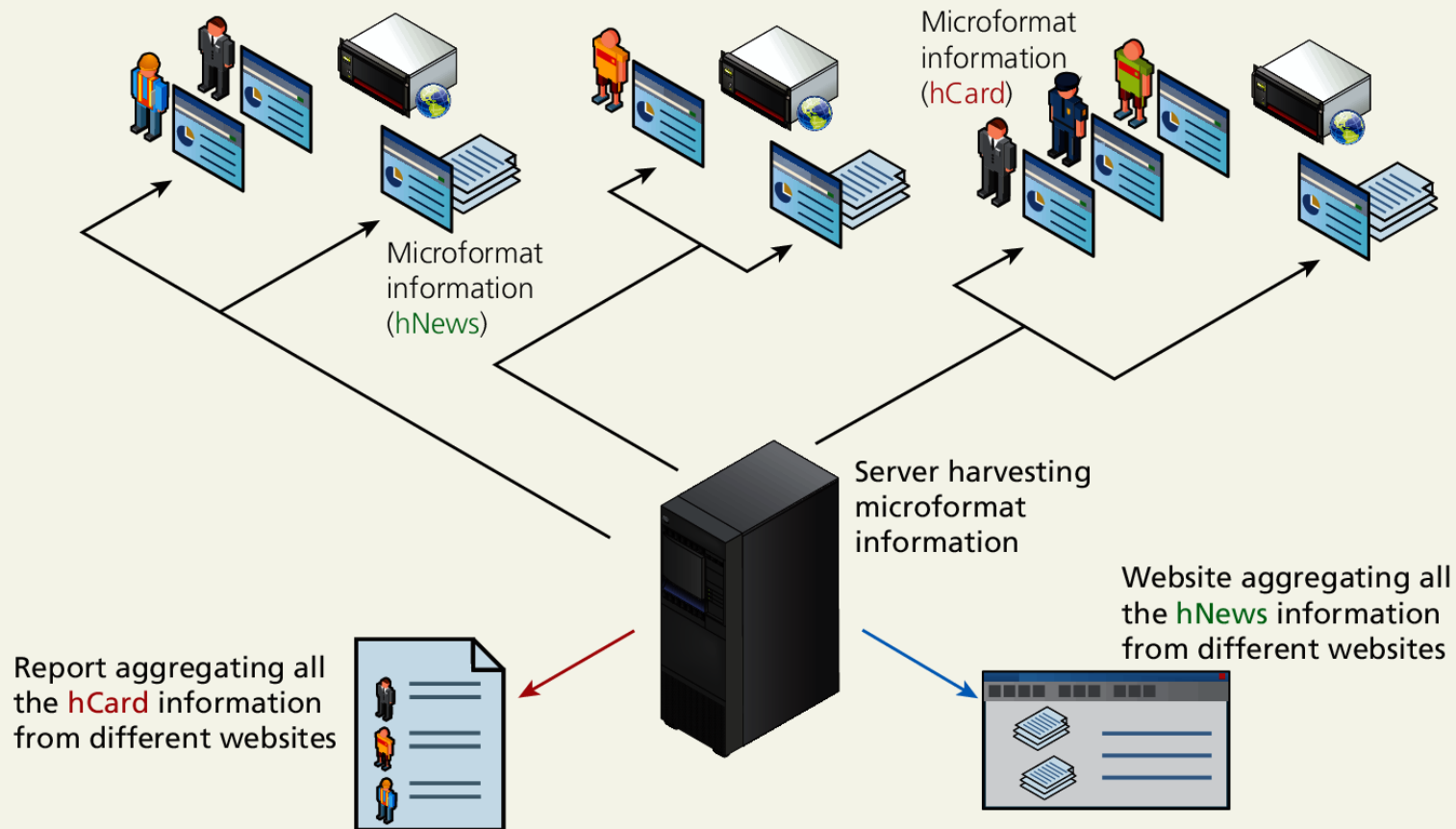
---

# Microformats

A **microformat** is a small pattern of HTML markup and attributes to represent common blocks of information such as people, events, and news stories so that the information in them can be extracted and indexed by software agents

---

# Microformat



# What you've learned

**1**

Introducing **Tables**

**2**

**Styling** Tables

**3**

Introducing **Forms**

**4**

**Form Control**  
Elements

**5**

Table and Form  
**Accessibility**

**6**

Microformats

---