HTML Forms

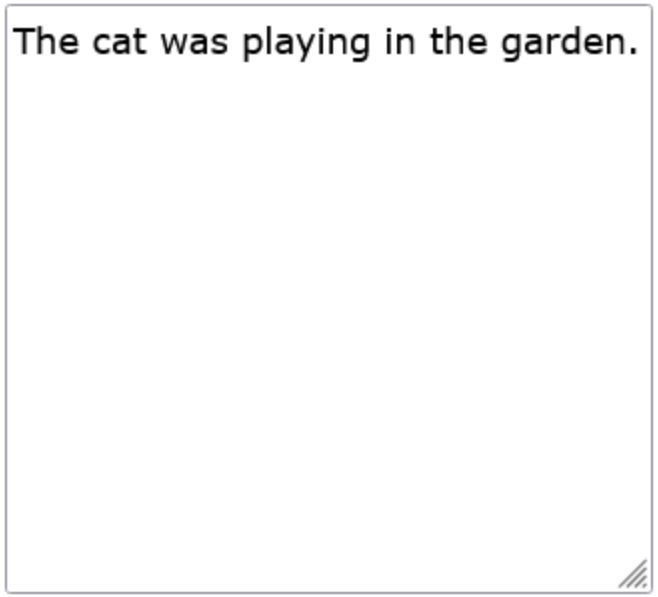
The <textarea> Element

- The <textarea> element defines a multi-line input field (a text area):

Example

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

- The rows attribute specifies the visible number of lines in a text area.
- The cols attribute specifies the visible width of a text area.
- This is how the HTML code above will be displayed in a browser



The <button> Element

- The <button> element defines a clickable button:

Example

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

This is how the HTML code above will be displayed in a browser:

- **Note:** Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

The <fieldset> and <legend> Elements

- The <fieldset> element is used to group related data in a form.

- The <legend> element defines a caption for the <fieldset> element.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname"
value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname"
value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Personalia:

First name:

John

Last name:

Doe

Submit

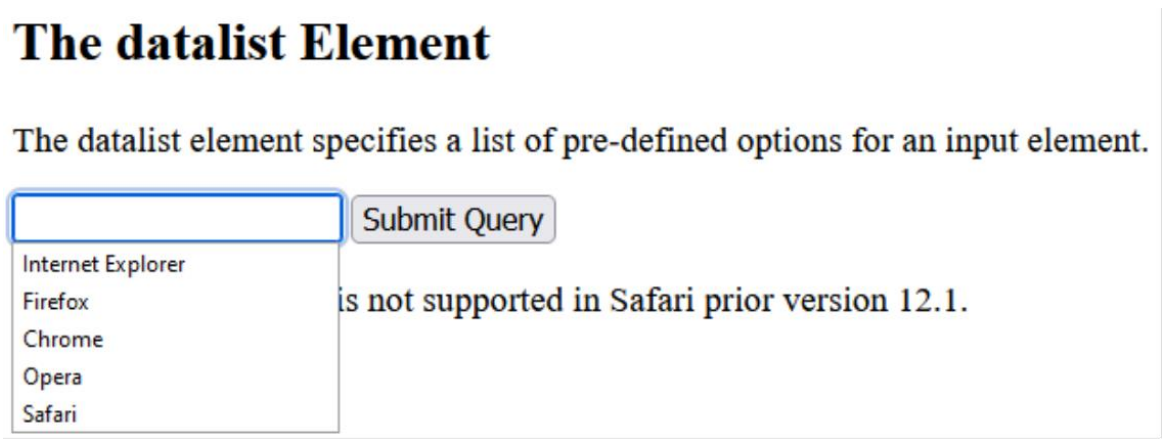
The <datalist> Element

- The <datalist> element specifies a list of pre-defined options for an <input> element.

- Users will see a drop-down list of the pre-defined options as they input data.

- The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```



The <output> Element

- The <output> element represents the result of a calculation (like one performed by a script).

```
<form action="/action_page.php"
 oninput="x.value=parseInt(a.value)+parseInt(b.valu
e)">
 0
 <input type="range"  id="a" name="a" value="50">
 100 +
 <input type="number" id="b" name="b"
value="50">
 =
 <output name="x" for="a b"></output>
 <br><br>
 <input type="submit">
</form>
```

The output Element

The output element represents the result of a calculation.

0 100 + 30 = 130

Submit Query

Note: The output element is not supported in Edge prior version 13.

Other Elements

<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list

HTML Input Types

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`

- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

Input attributes

Attribute	Description
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)

required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

HTML Block and Inline Elements

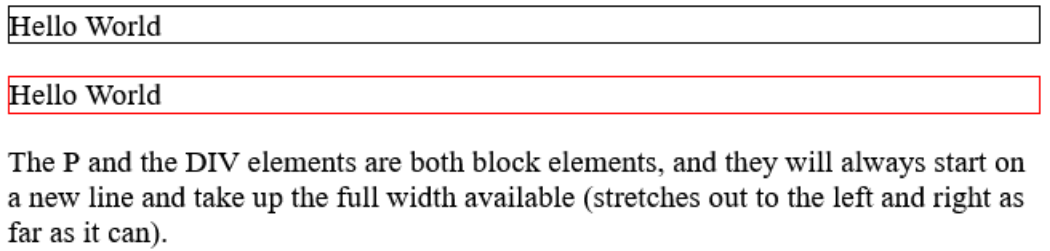
HTML Block and Inline Elements

- Every HTML element has a default display value, depending on what type of element it is.
- There are two display values: block and inline.

Block-level Elements

- A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.
- A block-level element always takes up the full width available (stretches out to the left and right as far as it can).
- Two commonly used block elements are: <p> and .
- The <p> element defines a paragraph in an HTML document.
- The element defines a division or a section in an HTML document.
- The <p> element is a block-level element.
- The element is a block-level element.

```
<!DOCTYPE html>
<html>
<body>
<p style="border: 1px solid black">Hello World</p>
<div style="border: 1px solid red">Hello World</div>
<p>The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).</p>
</body>
</html>
```



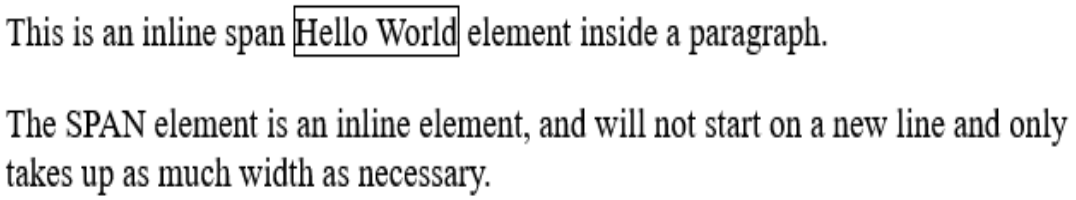
Here are the block-level elements in HTML:

<address>	<article>	<aside>	<blockquote>	<canvas>
<dd>		<dl>	<dt>	<fieldset>
<figcaption>	<figure>	<footer>	<form>	<h1>-<h6>
<header>	<hr>		<main>	<nav>
<noscript>		<p>	<pre>	<section>
<table>	<tfoot>		<video>	

Inline Elements

- An inline element does not start on a new line.
- An inline element only takes up as much width as necessary.
- **Note:** An inline element cannot contain a block-level element!

```
<!DOCTYPE html>
<html>
<body>
<p>This is an inline span <span style="border: 1px solid black">Hello World</span> element inside a paragraph.</p>

<p>The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.</p>
</body>
</html>
```



Here are the inline elements in HTML:

<a>	<abbr>	<acronym>	<b>	<bdo>
<big>	 	<button>	<cite>	<code>
<dfn>		<i>	<img>	<input>
<kbd>	<label>	<map>	<object>	<output>
<q>	<samp>	<script>	<select>	<small>
		<sub>	<sup>	<textarea>
<time>	<tt>	<var>		

HTML class Attribute

- The HTML class attribute is used to specify a class for an HTML element.
- Multiple HTML elements can share the same class using The class Attribute

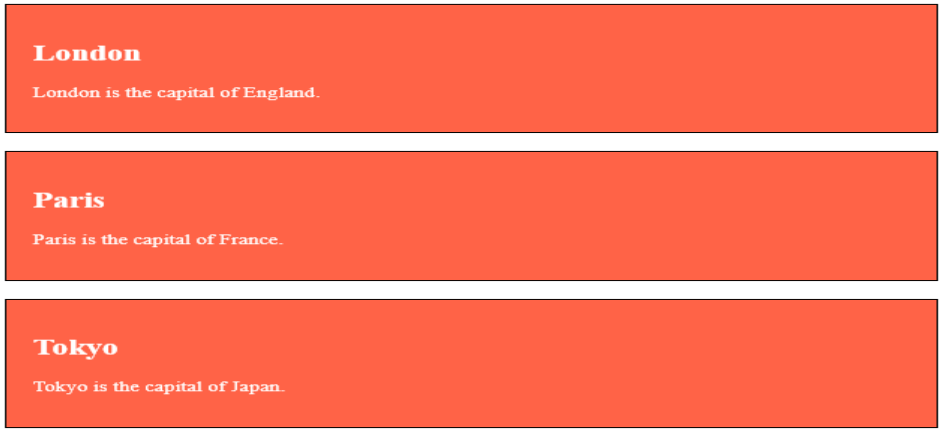
- The class attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.

- In the following example we have three elements with a class attribute with the value of "city". All of the three elements will be styled equally according to the .city style definition in the head section:

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: orange;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>
```

```
<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

(Future modifications: Best practice)



- Tip: The class attribute can be used on any HTML element.

- Note: The class name is case sensitive!

HTML id Attribute

- The HTML id attribute is used to specify a unique id for an HTML element.

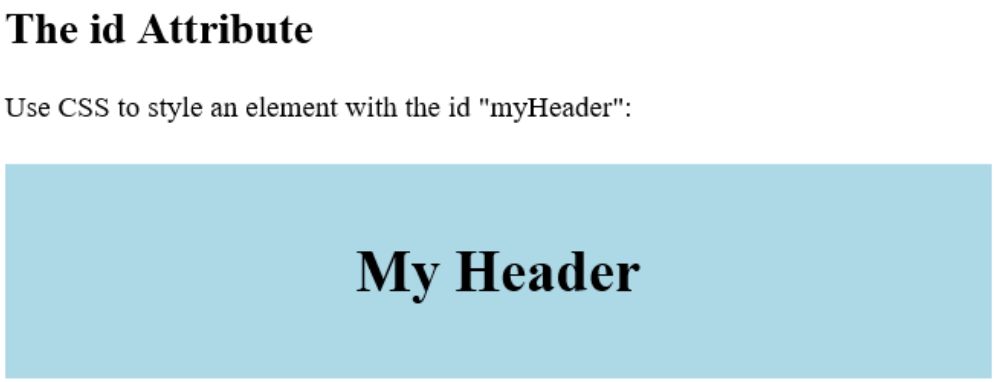
- You cannot have more than one element with the same id in an HTML document.

- The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

- The syntax for id is: write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

- In the following example we have an <h1> element that points to the id name "myHeader". This <h1> element will be styled according to the #myHeader style definition in the head section

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h2>The id Attribute</h2>
<p>Use CSS to style an element with the id "myHeader":</p>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

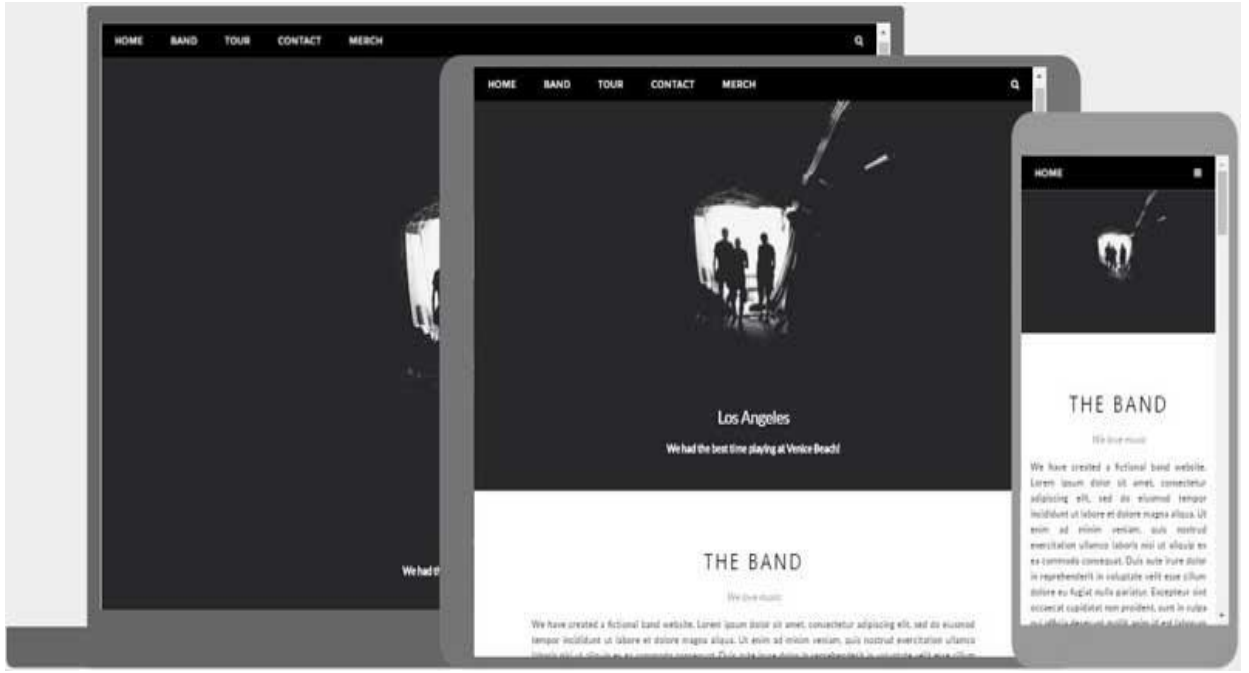


- Note: The id name is case sensitive!

- Note: The id name must contain at least one character, cannot start with a number, and must not contain whitespaces (spaces, tabs, etc.).

HTML Responsive Web Design

- Responsive web design is about creating web pages that look good on all devices!
- A responsive web design will automatically adjust for different screen sizes and viewports. Ex: gus.pittstate.edu



Setting The Viewport

- To create a responsive website, add the following <meta> tag to all your web pages:
- Example
- <meta name="viewport" content="width=device-width, initial-scale=1.0">

- This will set the viewport of your page, which will give the browser instructions on how to control the page's dimensions and scaling.
- Here is an example of a web page *without* the

Without the viewport meta tag:

With the viewport meta tag:

Responsive Images

- Responsive images are images that scale nicely to fit any browser size.
- Using the width Property

- If the CSS width property is set to 100%, the image will be responsive and scale up and down

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h2>Responsive Image</h2>
<p>When the CSS width property is set in a percentage value, the image will scale up and down when resizing the browser window. Resize the browser window to see the effect.</p>

<img src="img_goat.jpg" style="width:100%;">

</body>
</html>
```

- In the example above, the image can be scaled up to be larger than its original size. A better solution, in many cases, will be to use the max-width property instead.
- **Using the max-width Property**
- If the max-width property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size
- <img src="img_goat.jpg" style="max-width:100%;height:auto;">

Show Different Images Depending on Browser Width

- The HTML <picture> element allows you to define different images for different browser window sizes.
- Resize the browser window to see how the image below changes depending on the width

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h2>Show Different Images Depending on Browser Width</h2>
<p>Resize the browser width and the image will change at 600px and 1500px.</p>

<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  <img src="img_flowers.jpg" alt="Flowers" style="width:auto;">
</picture>

</body>
</html>
```

Responsive Text Size

- The text size can be set with a "vw" unit, which means the "viewport width".
- That way the text size will follow the size of the browser window
- Example

```
<h1 style="font-size:10vw">Hello World</h1>
```

- Viewport is the browser window size. 1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.

Media Queries

- In addition to resize text and images, it is also common to use media queries in responsive web pages.

- With media queries you can define completely different styles for different browser sizes.

- Example: resize the browser window to see that the three div elements will display horizontally on large screens and stack vertically on small screens

```
<style>
.left, .right {
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  float: left;
  width: 60%; /* The width is 60%, by default */
}

/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
  }
}
</style>
```

HTML Semantic Elements

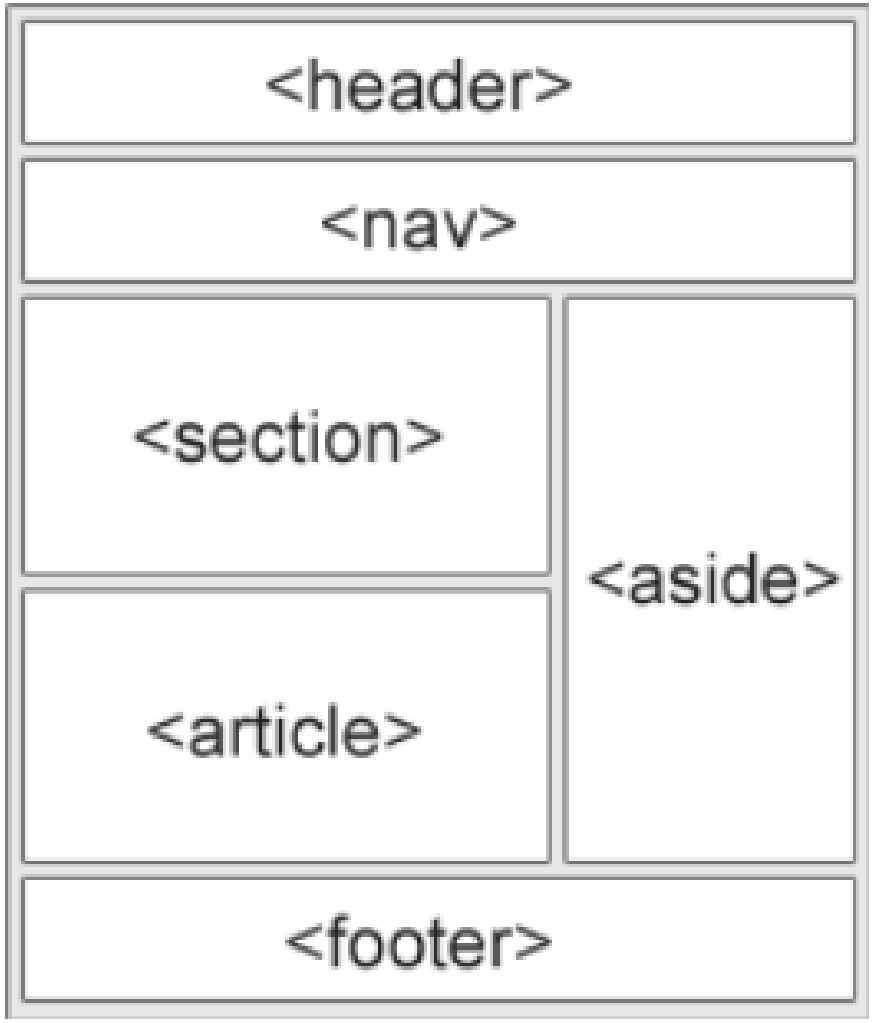
- A semantic element clearly describes its meaning to both the browser and the developer.

- Examples of non-semantic elements: and - Tells nothing about its content.

- Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.

- Many web sites contain HTML code like: to indicate navigation, header, and footer.

- In HTML there are some semantic elements that can be used to define different parts of a web page


<header>
<nav>
<section>
<article>
<aside>
<footer>

Semantic Elements in HTML

Tag	Description
<article>	Defines independent, self-contained content
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document

<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

HTML Entities

- Some characters are reserved in HTML.
- If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags.
- Character entities are used to display reserved characters in HTML.
- A character entity looks like this:

&*entity_name*;

- To display a less than sign (<) we must write: **&lt**

OR

&#*entity_number*;

- To display a less than sign (<) we must write **<**

- **Advantage of using an entity name:** An entity name is easy to remember.

- **Disadvantage of using an entity name:** Browsers may not support all entity names, but the support for entity numbers is good.

Getting Started with HTML5

- **HTML** stands for *Hyper Text Markup Language*. It is used to design web pages.
- HTML5 is the latest version of HTML.
- HTML5 comes with a lot of flexibility and it supports the following features −
- Uppercase tag names.
- Quotes are optional for attributes.
- Attribute values are optional.
- Closing empty elements are optional.