FORMs in HTML

7.1 About Forms

- Forms are used to collect information from people viewing your web site.
- For example, you can use forms to find out details about your visitors through surveys and feedback, or engage in e-commerce by selling your goods and services to people.
- Forms are defined by the <FORM> </FORM> tags and are made up of different elements to collect data.
- Once the user inputs all of the information, they submit the form by using the "submit" button that you create.
- What happens with the data is a decision you will need to make.
- You can use a script to manage the data, sent the data to database, or even receive data via e-mail.

7.1 About Forms

- Forms can contain;
  - Text boxes
  - Password boxes
  - Check boxes
  - Radio buttons
  - Buttons
  - Select lists
  - Text areas
  - Labels
  - Fieldsets

7.2 Form Elements

- All form elements should be written in between the <form>..</form> tags.

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<label>	Defines a label for an <input> element
<textarea>	Defines a multiline input control
<select>	Defines a drop-down list
<option>	Defines an option in a drop-down list
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<button>	Defines a clickable button

7.2 Form Elements - <form>

- The <FORM> </FORM> element is used to create an HTML form and act as a container for form elements. Although the form element itself isn't usually a visible part of the page (like the body tag), it could be with appropriate CSS.

Most commonly used **FORM** element **Attributes**

Attribute	Description
method	Specifies the HTTP method used when submitting the form
action	Specifies an address (url) where to submit the form
autocomplete	Specifies if the browser should autocomplete the form
novalidate	Specifies that the browser should not validate the form.
name	Specifies a name used to identify the form

7.2 Form Elements - <form>

- The **METHOD** attribute specifies the HTTP method to be used when submitting the form data:
  - **GET**
  - **POST**
- **GET**;
  - The default method when submitting form data
  - Submitted form data will be visible in the page address field
  - The length of a URL is limited (about 3000 characters)
  - Never used to send sensitive data! Better for non-secure data
  - Useful for form submissions where a user want to bookmark the result
- **POST**;
  - The POST method does not display the submitted form data in the page address field.
  - Used for sensitive or personal information.
  - Has no size limitations, and can be used to send large amounts of data.

7.2 Form Elements - <form>

- The **ACTION** attribute defines the action to be performed when the form is submitted.
- Normally, the form data is sent to a web page on the server when the user clicks on the submit button.
- In the example below, the form data is sent to a page on the server called "action_page.php". This page contains a server-side script that handles the form data:
  - <form action="action_page.php">

7.2 Form Elements - <form>

- **AUTOCOMPLETE** attribute is used to provide an autocompletion option to user, when user visit the form page. Default value is "on"
- If autocompletion is on, it will autocomplete the form and if autocompletion is off, the user have to fill the form field mannual.
- It is possible to have autocomplete “on” and “off” for the form, and “off” and “on” for specific input fields.
- The autocomplete attribute works with <form> and the following <input> types:
  - text
  - search
  - url
  - tel
  - password
  - datepickers
  - color

```
<form action="demo_form.php" autocomplete="on">
```

7.2 Form Elements - <form>

- **NOVALIDATE** attribute is used to send the information for not validating the form field. It specifies that form data shouldn't be validated.
  - <form action="demo_form.php" novalidate>

- **NAME** attribute used to identify the form.
  - for DOM usage: (document.forms.name)

7.2 Form Elements - <input>

- The most important form element is the input element.
- An input element can vary in many ways, depending on the type attribute.
- An input element can be of type
  - text,
  - checkbox,
  - password,
  - radio,
  - submit,
  - reset
  - File
  - image

7.2 Form Elements - <input>

INPUT element Attributes:

- **TYPE** (required)
  - Defines the usage of the **INPUT** element.
  - Hidden inputs always have **TYPE = "hidden".**
- **NAME** provides a unique identification for **INPUT** element.
  - Each input field must have a **name** attribute to be submitted.
  - If the name attribute is omitted, the data of that input field will not be sent at all.
- **VALUE** indicates the value that the **INPUT** element sends to the server upon submission.
- **SIZE** attribute specifies the size for the input field. (in characters)
- **MAXLENGTH** attribute specifies the maximum number of characters that the input field will accept.

7.2 Form Elements - <input>

INPUT element Attributes:

- The **READONLY** attribute specifies that the input field is read only (cannot be changed)
  - <input type="text" name="firstname" value="Raygan" readonly>
- The **DISABLED** attribute specifies that the input field is disabled.
  - A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form
  - <input type="text" name="firstname" value="Raygan" disabled>

7.2 Form Elements - <input>

Text Box

- Text boxes allow the users to enter a single-line text.
- Default width of a text field is 20 characters.

Example

```
First name: <input type="text" name="fname" size="25"><br>
Last name:  <input type="text" name="lname" size="25">
```

Result

First name:

Last name:

7.2 Form Elements - <input>

Password Box

- Password boxes are like text boxes, except the characters in a password field are automatically masked. (shown as asterisks or circles)

Example

```
User Name:<br>
<input type="text" name="username"><br>
Password:<br>
<input type="password" name="pswd">
```

Result

User Name:

Password:

7.2 Form Elements - <input>

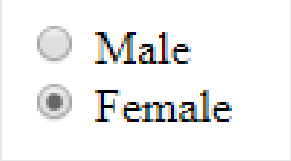
Radio Buttons

- Usually found in a group of options, only one radio button in a group can be selected at a time.
- Selecting one radio button deselects the others in its group.
- Each radio button within a group should have the same name and different values. (Otherwise, browsers cannot distinguish between them)
- **CHECKED** attribute indicates which radio button is selected initially

Example

```
<input type="radio" name="gender" value="male"> Male<br>
<input type="radio" name="gender" value="female" checked>Female<br>
```

Result



7.2 Form Elements - <input>

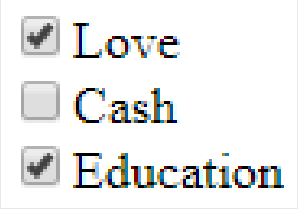
Check Boxes

- Check boxes let a user select NONE/ONE/MORE options of a limited number of choices.
- Each check box within a group should have the same name and different values. (Otherwise, browsers cannot distinguish between them)
- **CHECKED** attribute indicates initially selected checkboxe/s.

Example

```
<input type="checkbox" name="choice" value="cb1" checked>Love <br>
<input type="checkbox" name="choice" value="cb2">Cash <br>
<input type="checkbox" name="choice" value="cb3" checked>Education <br>
```

Result



7.2 Form Elements - <input>

Submit Button

- <input type="submit"> defines a submit button.
- A submit button is used to send form data to a server.
- The data is sent to the page specified in the form's action attribute.
- The file (form-handler) defined in the action attribute usually does something with the received input. (include script for processing input data).
- **VALUE** attribute changes the text displayed on the button (default is "**Submit**").

7.2 Form Elements - <input>

Submit Button

Example

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

Result

Username: Submit

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "html_form_action.asp".

7.2 Form Elements - <input>

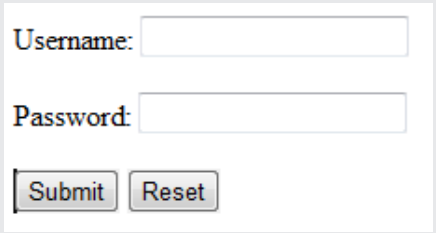
Reset Button

- A reset button is used to clear all the entries user entered into the form and reset the form-data to its default values.
- **VALUE** attribute changes the text displayed on the button (default is "**Reset**")

Example

```
<form name="input" action="html_form_action.asp" method="get">
<P>Username: <input type="text" name="user" size="25"></P>
<P>Password: <input type="password" name="pswd" size="25"></P>
<P><input type="submit" value="Submit">
<input type="reset" value="Reset"></P></form>
```

Result



7.2 Form Elements - <label>

Label

- The **<label>** tag defines a label for an **<input>** element.
- The <label> element does not render as anything special for the user. However, it provides a usability improvement for mouse users, because if the user clicks on the text within the <label> element, it toggles the control.
- The **for** attribute of the <label> tag should be equal to the **id** attribute of the related element to bind them together.
- A label can be bound to an element either by using the "for" attribute, or by placing the element inside the <label> element.

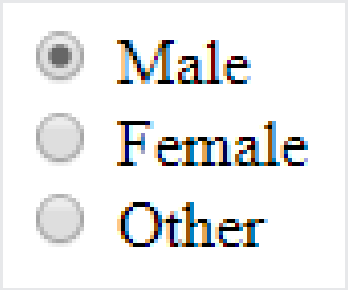
7.2 Form Elements - <label>

Label

Example

```
<input type="radio" name="gender" id="male" value="male" checked>
<label for="male">Male</label><br>

<input type="radio" name="gender" id="female" value="female">
<label for="female">Female</label><br>

<input type="radio" name="gender" id="other" value="other">
<label for="other">Other</label><br>
```

Result



7.2 Form Elements - <button>

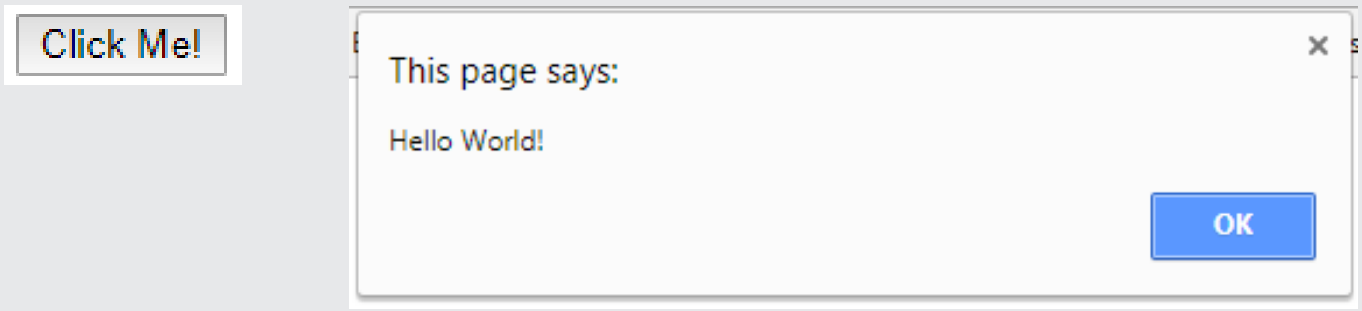
Button

- The <button> element defines a clickable button.

Example

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

Result



7.2 Form Elements - <textarea>

Text Area

- Inserts a scrollable text box into FORM for entering multi-line text.
- It is commonly used in situations where you ask for info that may require multiple sentences.
- You control the dimension of the text area by using the **ROWS** and **COLS** attributes.
- The **rows** attribute specifies the visible number of lines in a text area.
- The **cols** attribute specifies the visible width of a text area.

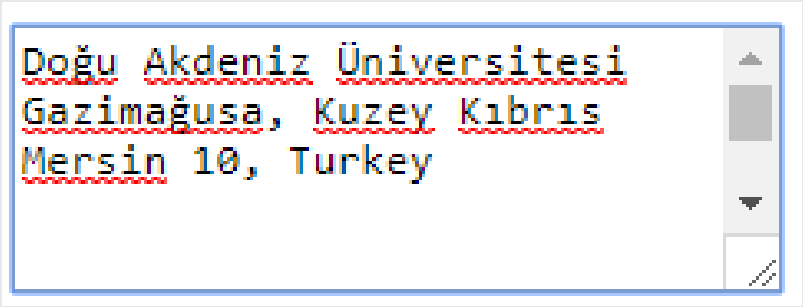
7.2 Form Elements - <textarea>

Text Area

Example

```
<textarea name="message" rows="5" cols="30">
Doğu Akdeniz Üniversitesi
Gazimağusa, Kuzey Kıbrıs
Mersin 10, Turkey
</textarea>
```

Result



7.2 Form Elements - <select>,<option>

List Box

- **<select>** tag presents a drop-down list with choices indicated by the **<option>** tags
  - Include **NAME** attribute
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the selected attribute to the option
- Change the number of list options visible by including the **SIZE** = "x" attribute inside the <SELECT> tag
  - *x* number of options visible
- Use the **multiple** attribute to allow the user to select more than one value:
  - if you use multiple attribute, you should also assign different values for each of the **value** attributes of option tags

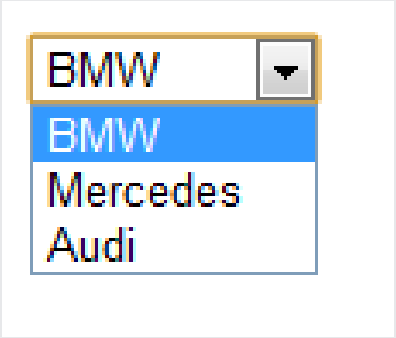
7.2 Form Elements - <select>,<option>

List Box

Example

```
<select name="cars">
<option selected>BMW</option>
<option>Mercedes</option>
<option>Audi</option>
</select>
```

Result



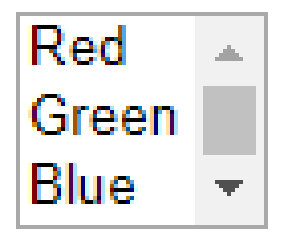
7.2 Form Elements - <select>,<option>

List Box

Example

```
<select name="colors" size="3" multiple>
    <option value="red">Red</option>
    <option value="green">Green</option>
    <option value="blue">Blue</option>
    <option value="purple">Purple</option>
 </select>
```

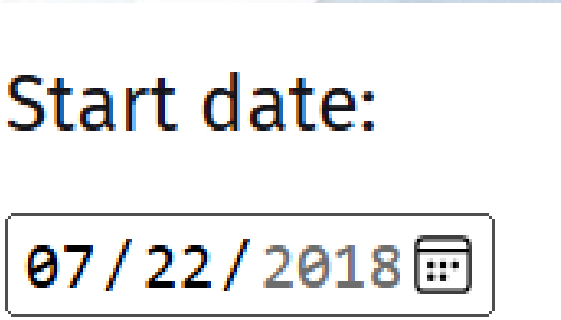
Result



<input type="date">

<input> elements of **type="date"** create input fields that let the user enter a date. The appearance of the date picker input UI varies based on the browser and operating system. The value is normalized to the format yyyy-mm-dd.

- <label for="start">Start date:</label>
- <input type="date" id="start" name="trip-start" value="2018-07-22" min="2018-01-01" max="2018-12-31" />



<input type="datetime-local">

<input> elements of type **datetime-local** create input controls that let the user easily enter both a date and a time, including the year, month, and day as well as the time in hours and minutes.

- <label for="meeting-time">Choose a time for your appointment:</label>
- <input type="datetime-local" id="meeting-time" name="meeting-time" value="2018-06-12T19:30" min="2018-06-07T00:00" max="2018-06-14T00:00" />

Choose a time for your appointment:

<input type="file">

`<input>` elements with `type="file"` let the user choose one or more files from their device storage. Once chosen, the files can be uploaded to a server using form submission, or manipulated using JavaScript code and the File API.

```
  <body>
<form method="post" enctype="multipart/form-data">
  <div>
    <label for="profile_pic">Choose file to upload</label>
    <input
      type="file"
      id="profile_pic"
      name="profile_pic"
      accept=".jpg, .jpeg, .png" />
  </div>
  <div>
    <button>Submit</button>
  </div>
</form>
  </body>
</html>
```

<input type="image">

`<input>` elements of type `image` are used to create graphical submit buttons, i.e. submit buttons that take the form of an image rather than text.

```
  <head>
    <title>form image</title>
  </head>
  <body>
<p>Sign in to your account:</p>

<div>
  <label for="userId">User ID</label>
  <input type="text" id="userId" name="userId">
</div>

<input type="image" id="image" alt="Login"
       src="jacket.JPG">
  </body>
</html>
```

<input type="range">

`<input>` elements of type `range` let the user specify a numeric value which must be no less than a given value, and no more than another given value. The precise value, however, is not considered important. This is typically represented using a slider or dial control rather than a text entry box like the number input type.

```
<p>Audio settings:</p>

<div>
  <input type="range" id="volume"
name="volume"
         min="0" max="11">
  <label for="volume">Volume</label>
</div>

<div>
  <input type="range" id="cowbell"
name="cowbell"
         min="0" max="100" value="90"
step="10">
  <label for="cowbell">Cowbell</label>
</div>
```

Audio settings:

Volume

Cowbell

7.2 Form Elements - <fieldset>,<legend>

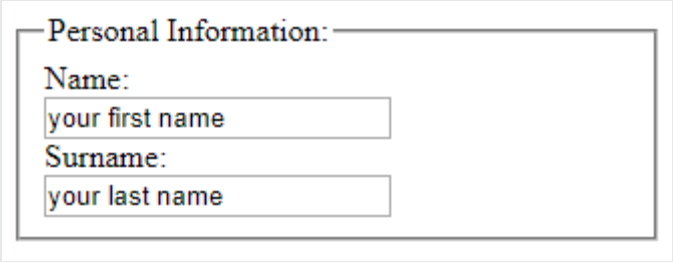
Grouping Form Data

- The **<fieldset>** element is used to group related data in a form.
- The **<legend>** element defines a caption for the **<fieldset>** element.

Example

```
<fieldset><legend>Personal Information:</legend>
    Name:<br>
    <input type="text" name="firstname" value="your first name"><br>
    Surname:<br>
    <input type="text" name="lastname" value="your last name">
</fieldset>
```

Result



7.2 Form Elements

Example

```
<!DOCTYPE html>

<!-- Form using a variety of components. -->
<html>
<head>
<meta charset = "utf-8">
<title>Form Example-1</title>
</head>
<body>
<h1>Feedback Form</h1>
<p>Please fill out this form to help us improve our site.</p>

<form method = "post" action = "">

<p><label><strong>Name:</strong>
<input name = "name" type = "text" size = "25">
</label></p>

<p><label><strong>Comments:</strong><br>
<textarea name = "comments" rows = "4" cols = "36"></textarea>
</label></p>

<p><label><strong>E-mail Address:</strong>
<input name = "email" type = "email" size = "25">
</label></p>
```

7.2 Form Elements

Example (cont..)

```
<p><strong>Things you liked:</strong><br>
<label>Site design
<input name = "thingsliked" type = "checkbox" value = "Design"></label>
<label>Links
<input name = "thingsliked" type = "checkbox" value = "Links"></label>
<label>Ease of use
<input name = "thingsliked" type = "checkbox" value = "Ease"></label>
<label>Images
<input name = "thingsliked" type = "checkbox" value = "Images"></label>
<label>Source code
<input name = "thingsliked" type = "checkbox" value = "Code"></label>
</p>

<p><strong>How did you get to our site?:</strong><br>
<label>Search engine
<input name = "howtosite" type = "radio" value = "search engine" checked></label>
<label>Links from another site
<input name = "howtosite" type = "radio" value = "link"></label>
<label>Reference in a book
<input name = "howtosite" type = "radio" value = "book"></label>
<label>Other
<input name = "howtosite" type = "radio" value = "other"></label>
</p>
```

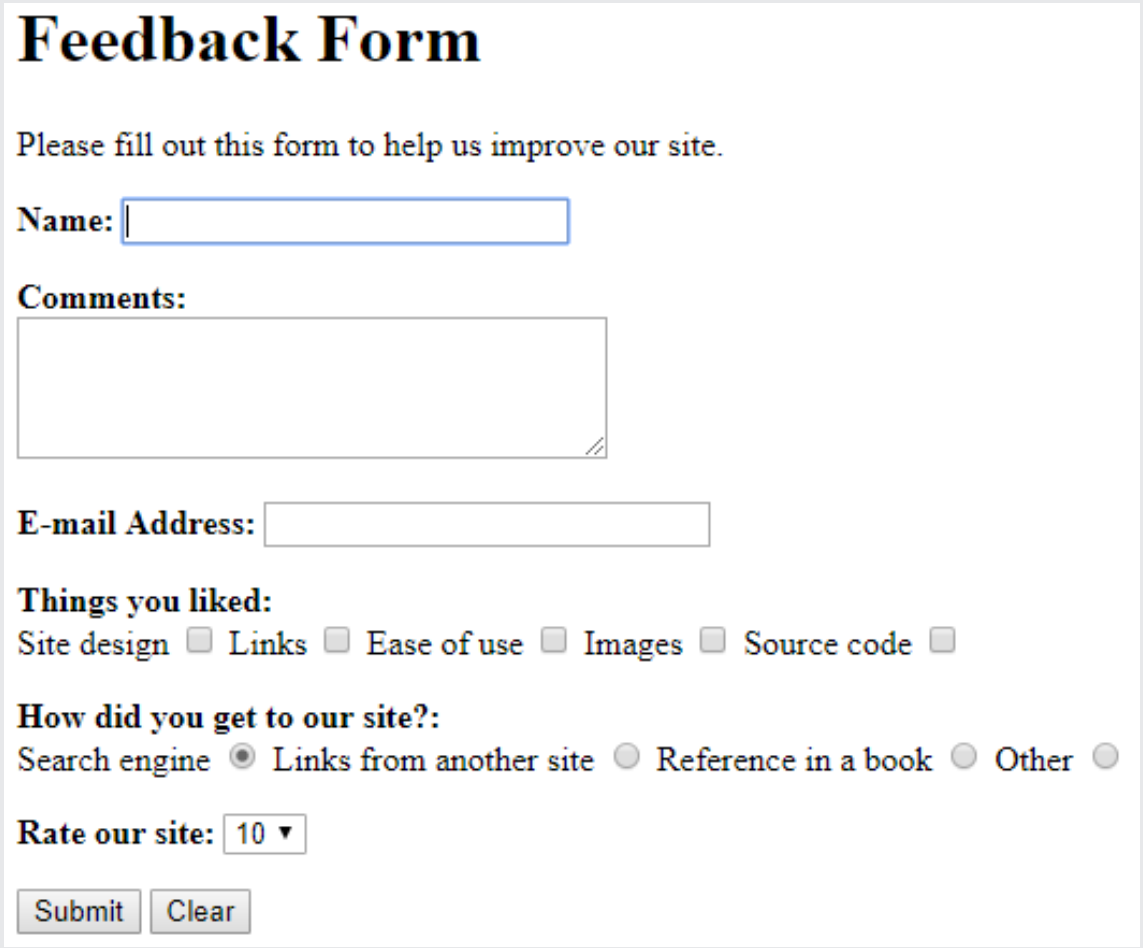
7.2 Form Elements

Example (cont..)

```
<p>
<b>Rate our site:<b>
<select name = "rating">
<option selected>10</option>
<option>9</option>
<option>8</option>
<option>7</option>
<option>6</option>
<option>5</option>
<option>4</option>
<option>3</option>
<option>2</option>
<option>1</option>
</select>
</p>

<p>
<input type = "submit" value = "Submit">
<input type = "reset" value = "Clear">
</p>
</form>
</body>
</html>
```

7.2 Form Elements

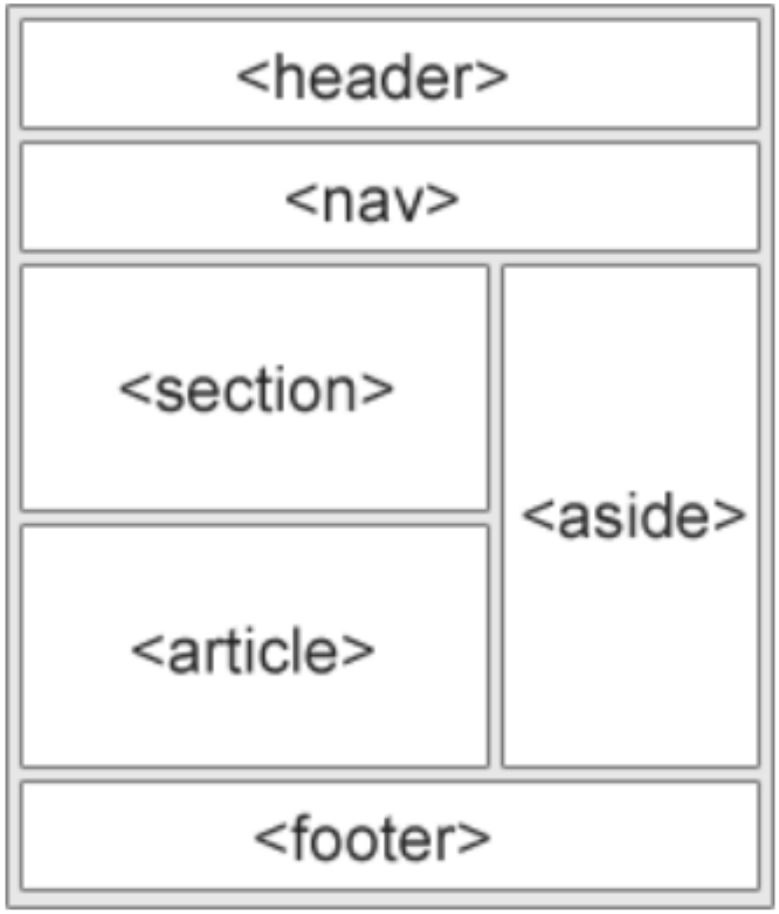
Output



What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.
Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.
Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

- Semantic Elements in HTML
- 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>
<aside>
<article>
<footer>

HTML <section> Element

The `<section>` element defines a section in a document.
According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."
Examples of where a `<section>` element can be used:

- Chapters
- Introduction
- News items
- Contact information

A web page could normally be split into sections for introduction, content, and contact information.

```
<html>
<body>

<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is an international
organization working on issues regarding the conservation, research and
restoration of the environment, formerly named the World Wildlife Fund.
WWF was founded in 1961.</p>
</section>

<section>
  <h1>WWF's Panda symbol</h1>
  <p>The Panda has become the symbol of WWF. The well-known panda logo
of WWF originated from a panda named Chi Chi that was transferred from
the Beijing Zoo to the London Zoo in the same year of the establishment
of WWF.</p>
</section>

</body>
</html>
```

WWF

The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.

WWF's Panda symbol

The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.

HTML <article> Element

The `<article>` element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the `<article>` element can be used:

- Forum posts
- Blog posts
- User comments
- Product cards
- Newspaper articles

```
<h1>The article element</h1>

<article>
  <h2>Google Chrome</h2>
  <p>Google Chrome is a web browser developed by Google, released in
2008. Chrome is the world's most popular web browser today!</p>
</article>

<article>
  <h2>Mozilla Firefox</h2>
  <p>Mozilla Firefox is an open-source web browser developed by
Mozilla. Firefox has been the second most popular web browser since
January, 2018.</p>
</article>

<article>
  <h2>Microsoft Edge</h2>
  <p>Microsoft Edge is a web browser developed by Microsoft, released
in 2015. Microsoft Edge replaced Internet Explorer.</p>
</article>

</body>
```

The article element

Google Chrome

Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!

Mozilla Firefox

Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.

Microsoft Edge

Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.

HTML <header> Element

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

```
<!DOCTYPE html>
<html>
<body>

<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment, and build a future in which humans live in harmony with
nature.</p>
</article>

</body>
</html>
```

What Does WWF Do?

WWF's mission:

WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.

HTML <footer> Element

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

```
<!DOCTYPE html>
<html>
<body>

<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>

</body>
</html>
```

Author: Hege Refsnes

hege@example.com

HTML <nav> Element

The <nav> element defines a set of navigation links.

```
<!DOCTYPE html>
<html>
<body>

<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>

</body>
</html>
```

HTML | CSS | JavaScript | jQuery

The `<time>` tag defines a specific time (or datetime).
The `datetime` attribute of this element is used translate the time into a machine-readable format so that browsers can offer to add date reminders through the user's calendar, and search engines can produce smarter search results.

```
<!DOCTYPE html>
<html>
<body>

<h1>The time element</h1>

<p>Open from <time>10:00</time> to <time>21:00</time> every weekday.</p>

<p>I have a date on <time datetime="2008-02-14 20:00">Valentines
day</time>.</p>

<p><b>Note:</b> The time element does not render as anything special in
any of the major browsers.</p>

</body>
</html>
```

The time element

Open from 10:00 to 21:00 every weekday.

I have a date on Valentines day.

Note: The time element does not render as anything special in any of the major browsers.

Definition and Usage

The <progress> tag represents the completion progress of a task.

```
<!DOCTYPE html>
<html>
<body>

<h1>The progress element</h1>

<label for="file">Downloading progress:</label>
<progress id="file" value="32" max="100"> 32% </progress>

</body>
</html>
```

The progress element

Downloading progress: 

Definition and Usage

The <audio> tag is used to embed sound content in a document, such as music or other audio streams.

The <audio> tag contains one or more <source> tags with different audio sources. The browser will choose the first source it supports.

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

There are three supported audio formats in HTML: MP3, WAV, and OGG.

```
<!DOCTYPE html>
<html>
<body>

<h1>The audio element</h1>

<p>Click on the play button to play a sound:</p>

<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

</body>
</html>
```

The audio element

Click on the play button to play a sound:

0:00 / 0:01

Definition and Usage

The <video> tag is used to embed video content in a document, such as a movie clip or other video streams.

The <video> tag contains one or more <source> tags with different video sources. The browser will choose the first source it supports.

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

There are three supported video formats in HTML: MP4, WebM, and OGG.

```
<!DOCTYPE html>
<html>
<body>

<h1>The video element</h1>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

The video element

Definition and Usage

The `<div>` tag defines a division or a section in an HTML document.

The `<div>` tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The `<div>` tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the `<div>` tag!

```
<html>
<head>
<style>
.myDiv {
  border: 5px outset red;
  background-color: lightblue;
  text-align: center;
}
</style>
</head>
<body>

<h1>The div element</h1>

<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>

<p>This is some text outside the div element.</p>

</body>
</html>
```

The div element

This is a heading in a div element

This is some text in a div element.

This is some text outside the div element.

Definition and Usage

The tag is an inline container used to mark up a part of a text, or a part of a document.
The tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
The tag is much like the element, but is a block-level element and is an inline element.

```
<!DOCTYPE html>
<html>
<body>

<h1>The span element</h1>

<p>My mother has <span style="color:blue;font-weight:bold">blue</span>
eyes and my father has <span style="color:darkolivegreen;font-
weight:bold">dark green</span> eyes.</p>

</body>
</html>
```

The span element

My mother has **blue** eyes and my father has **dark green** eyes.

Definition and Usage

The <aside> tag defines some content aside from the content it is placed in.

The aside content should be indirectly related to the surrounding content.

Tip: The <aside> content is often placed as a sidebar in a document.

```
<!DOCTYPE html>
<html>
<body>

<h1>The aside element</h1>

<p>My family and I visited The Epcot center this summer. The weather was
nice, and Epcot was amazing! I had a great summer together with my
family!</p>

<aside>
  <h4>Epcot Center</h4>
  <p>Epcot is a theme park at Walt Disney World Resort featuring exciting
attractions, international pavilions, award-winning fireworks and
seasonal special events.</p>
</aside>

</body>
</html>
```

The aside element

My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

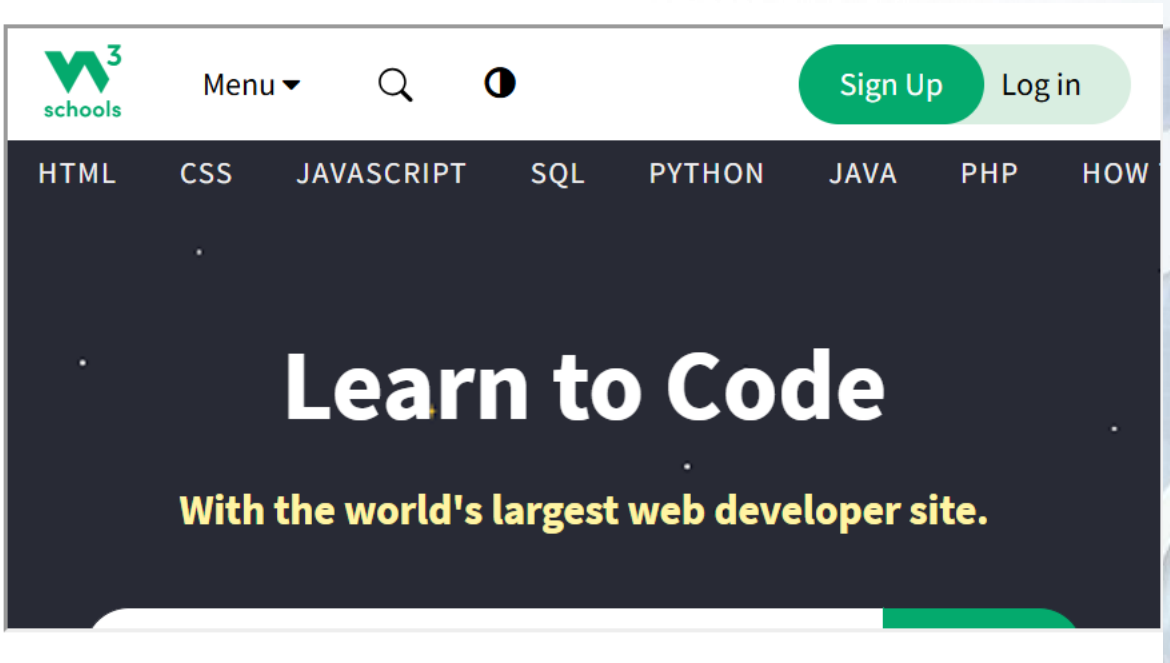
Epcot Center

Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.

Definition and Usage

The <iframe> tag specifies an inline frame.
An inline frame is used to embed another document within the current HTML document.

```
<!DOCTYPE html>
<html>
<body>

<h1>The iframe element + CSS</h1>

<p>An iframe with default borders:</p>
<iframe src="/default.asp" width="100%" height="300">
</iframe>

<p>An iframe with a thin black border:</p>
<iframe src="/default.asp" width="100%" height="300" style="border:1px
solid black;">
</iframe>

<p>An iframe with no borders:</p>
<iframe src="/default.asp" width="100%" height="300"
style="border:none;">
</iframe>

</body>
</html>
```


Menu
Sign Up
Log in
HTML
CSS
JAVASCRIPT
SQL
PYTHON
JAVA
PHP
Learn to Code
With the world's largest web developer site.

Definition and Usage

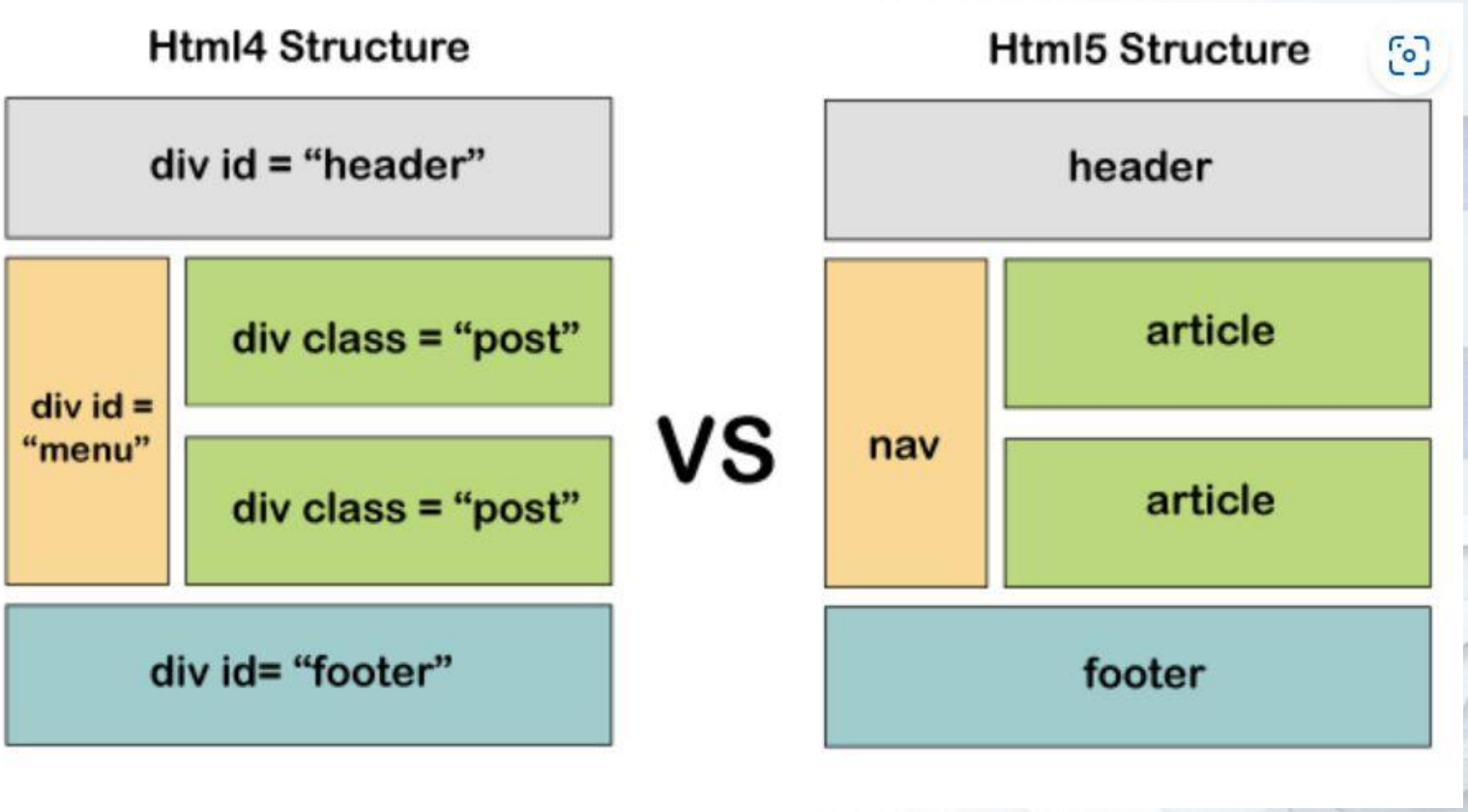
The <canvas> tag is used to draw graphics, on the fly, via scripting (usually JavaScript).
The <canvas> tag is transparent, and is only a container for graphics, you must use a script to actually draw the graphics.
Any text inside the <canvas> element will be displayed in browsers with JavaScript disabled and in browsers that do not support <canvas>

```
<!DOCTYPE html>
<html>
<body>
<h1>HTML5 Canvas</h1>

<canvas id="myCanvas" width="300" height="150" style="border:1px solid
grey"></canvas>

</body>
</html>
```

HTML5 Canvas

- HTML 5 and Features [5 Hrs.]
- 3.1. Introduction
- 3.2. Difference between HTML and HTML 5
- 3.3. HTML 5 New Semantics Elements (HEADER, FOOTER, SECTION)
- 3.4. HTML 5 New Elements
- 3.4.1. Tables, Images, Colors, Canvas, Forms
- 3.4.2. Interactive Elements
- 3.4.3. Graphics
- 3.4.4. Multimedia

- HTML5 is more complete and easier than **HTML4**, it has lots of new tags like **<header>**, **<footer>**, **<navy>**, **<Audio>**, **<video>**, **<main>** etc. It also supports graphics. In the following image, we have described all the essential terms related to HTML and HTML5.


Html4 Structure
div id = “header”
div id = “menu”
div class = “post”
div class = “post”
div id= “footer”
VS
Html5 Structure
header
nav
article
article
footer

- HTML is referred to as the **primary** language of the **World Wide Web**. **HTML** has many updates over time, and the latest **HTML** version is **HTML5**. There are some differences between the two versions:

1. HTML5 supports both **audio** and **video** while none of them were part of
2. HTML cannot allow JavaScript to run within the web browser, while **HTML5** provides full support for running JavaScript.
3. In **HTML5**, inline **mathML** and **SVG** can be used in a text, while in HTML it is not possible.
4. HTML5 supports new types of form controls, such as **date** and **time, email, number, category, title, Url, search, etc.**
5. Many elements have been introduced in HTML5. Some of the most important are **time, audio, description, embed, fig, shape, footer, article, canvas, navy, output, section, source, track, video**, etc.

HTML	HTML5
HTML does not provide native audio and video support.	HTML5 provides native audio and video support.
HTML only supports vector graphics if used in conjunction with different technologies like **Flash**, **VML**, or **Silverlight**.	HTML5 supports SVG (Scalable Vector Graphics), Canvas, and other virtual vector graphics.
HTML allows inline MathML and SVG in text with restricted use.	HTML5 allows inline **MathML** and **SVG** in text
HTML doesn't allow users to draw shapes such as circles, triangles, and rectangles.	HTML allows users to draw shapes such as circles, triangles, and rectangles.

HTML only uses browser cache and cookies to store data temporarily.	HTML5 uses web SQL databases, local storage, and application cache for storing data temporarily.
JavaScript and browser interface run in the same thread.	JavaScript and browser interface run in separate threads.
Longer **document type declaration**.	Shorter document type declaration.
Longer character encoding declaration. Uses the ASCII **character set**.	Shorter **character encoding** declaration. Uses the UTF-8 character set.
Compatible with almost all browsers.	Only compatible with newer browsers, considering there are many new tags and elements which only some browsers support.

- Built based on **Standard Generalized Markup Language** (SGML).

- HTML5 has improved parsing rules providing enhanced compatibility

What are Semantic Elements?

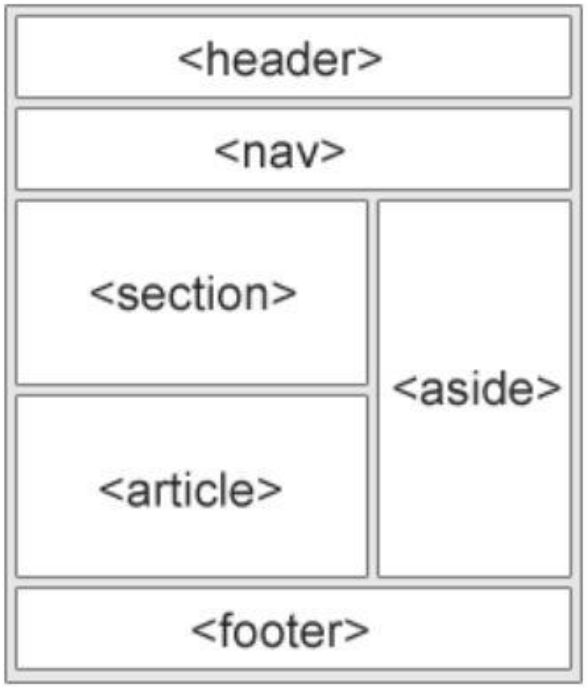
A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

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

- Semantic Elements in HTML
- 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:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



HTML 5 New Semantics Elements (HEADER, FOOTER, SECTION)

- HTML5 <header> Element
- The <header> element specifies a header for a document or section.
- The <header> element should be used as a container for introductory content.
- You can have several <header> elements in one document.

```
<!DOCTYPE html>
<html>
<body>

<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment, and build a future in which humans live in harmony with
nature.</p>
</article>

</body>
</html>
```

What Does WWF Do?

WWF's mission:

WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.

- HTML5 <footer> Element
- The <footer> element specifies a footer for a document or section.
- A <footer> element should contain information about its containing element.
- A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
- You can have several <footer> elements in one document.

```
<!DOCTYPE html>
<html>
<body>

<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">
  someone@example.com</a>.</p>
</footer>

</body>
</html>
```

Posted by: Hege Refsnes

Contact information: someone@example.com.

HTML <section> Element

The `<section>` element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a `<section>` element can be used:

- Chapters
- Introduction
- News items
- Contact information

```
<!DOCTYPE html>
<html>
<body>

<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is an international
organization working on issues regarding the conservation, research
and restoration of the environment, formerly named the World
Wildlife Fund. WWF was founded in 1961.</p>
</section>

<section>
  <h1>WWF's Panda symbol</h1>
  <p>The Panda has become the symbol of WWF. The well-known panda
logo of WWF originated from a panda named Chi Chi that was
transferred from the Beijing Zoo to the London Zoo in the same year
of the establishment of WWF.</p>
</section>

</body>
</html>
```

WWF

The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.

WWF's Panda symbol

The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.

7.3 HTML5 Input Types

- HTML5 added several new input types:
  - color
  - date
  - datetime-local
  - email
  - month
  - number
  - range
  - search
  - tel
  - time
  - url
  - week
- New input types that are not supported by older web browsers, will behave as <input type="text">.

7.3 HTML5 Input Types - color

- Is used for input fields that should contain a color.
- The color tool allows the user to choose a color using standard web formats.
- Depending on browser support, a color-picker can show up in the input field like the ones

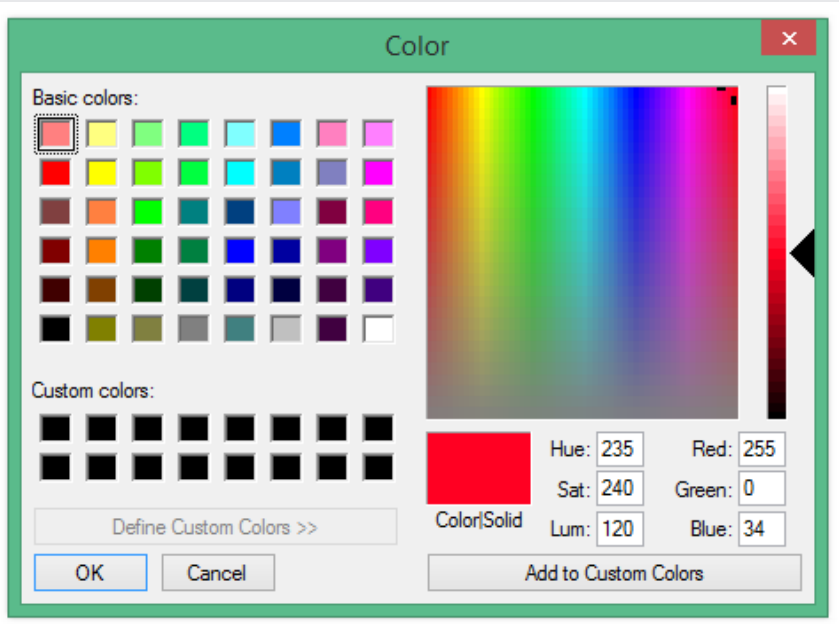
7.3 HTML5 Input Types - color

Example

```
Select your favorite color:
<input type="color" name="favcolor" value="#ff0022">
```

Result

Select your favorite color:



7.3 HTML5 Input Types - date

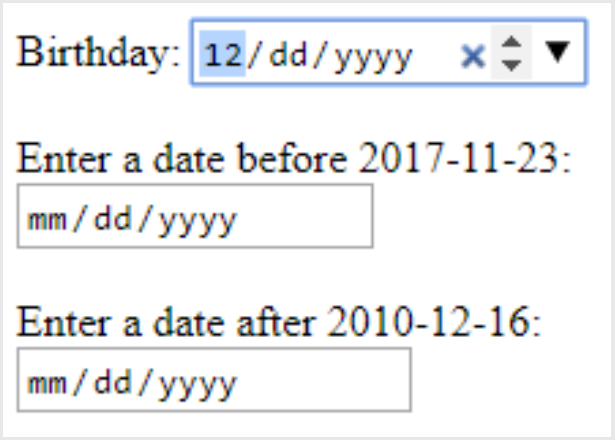
- Setting the input type to date indicates that you wish the user to enter a date.
- Depending on browser support, a date picker can show up in the input field.
- You can restrict the dates allowed to a specific range by applying the min and max attributes to the element.

7.3 HTML5 Input Types - date

Example

```
Birthday:
<input type="date" name="bday"><br><br>
Enter a date before 2017-11-23:<br>
<input type="date" name="bday" max="2017-11-22"><br><br>
Enter a date after 2010-12-16:<br>
<input type="date" name="bday" min="2010-12-17"><br><br>
```

Result



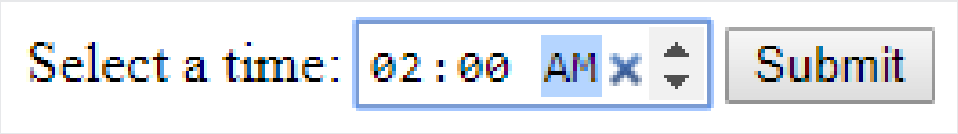
7.3 HTML5 Input Types - time

- The purpose of the time input type is to allow the user to enter a time.
- Depending on browser support a time picker might pop-up when you enter the input field.

Example

```
Select a time:
<input type="time" name="times">
```

Result



7.3 HTML5 Input Types – datetime-local

- The datetime-local element combines date and time in a single input field, with no time zone.
- Depending on browser support a time picker might pop-up when you enter the input field.

Example

```
Birthday (date and time):<br>
<input type="datetime-local" name="bdaytime">
```

Result

Birthday (date and time):

08/23/1977 09:10 AM

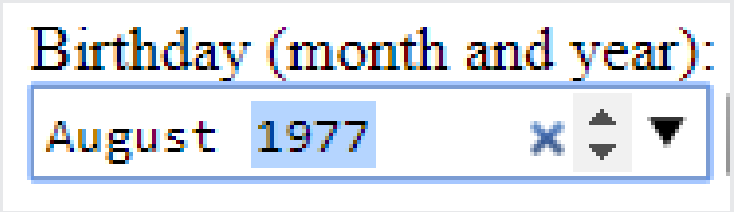
7.3 HTML5 Input Types - month

- Allows the user to select a month and year.
- Depending on browser support, a date picker can show up in the input field.

Example

```
Birthday (month and year):
<input type="month" name="bdaymonth">
```

Result



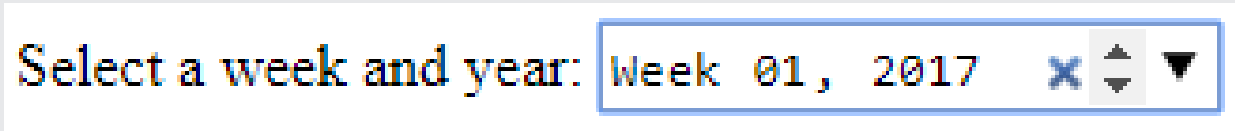
7.3 HTML5 Input Types - week

- Allows the user to select a week and year.
- Depending on browser support, a date picker can show up in the input field.

Example

```
Select a week and year:
<input type="week" name="week_year">
```

Result



7.3 HTML5 Input Types – email

- Used for input fields that should contain an e-mail address.
- Depending on browser support, the e-mail address can be automatically validated when submitted.

Example

```
E-mail:
<input type="email" name="mail">
```

Result

E-mail:

7.3 HTML5 Input Types - number

- Defines a **numeric** input field.
- You can also set restrictions on what numbers are accepted by using minand max attributes.

Example

```
Enter Your Office No. (between 101 and 112):
<input type="number" name="quantity" min="101" max="112">
```

Result

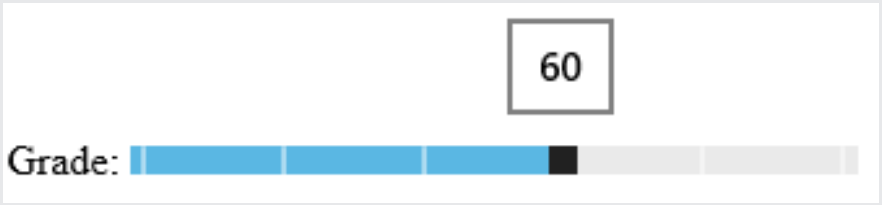
7.3 HTML5 Input Types - range

- Defines a control for entering a number whose exact value is not important.
- Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min and max attributes.
- Depending on browser support, the input type "range" can be displayed as a slider control.

Example

```
Grade:
<input type="range" name="points" min="0" max="100">
```

Result



7.3 HTML5 Input Types - tel

- Used for input fields that should contain a telephone number.
- The tel type is currently supported only in Safari 8.

Example

```
Telephone:
<input type="tel" name="telephone">
```

Result

Telephone:

7.3 HTML5 Input Types - search

- Used for search fields.
- A search field behaves like a regular text field.

Example

```
Search Google:
  <input type="search" name="googlesearch">
  <input type="submit" value="Search">
```

Result

Search Google: Search

7.3 HTML5 Input Types - url

- Used for input fields that should contain a URL address.
- Depending on browser support, the url field can be automatically validated when submitted.

Example

```
Add your homepage:
  <input type="url" name="homepage">
  <input type="submit" value="Submit">
```

Result

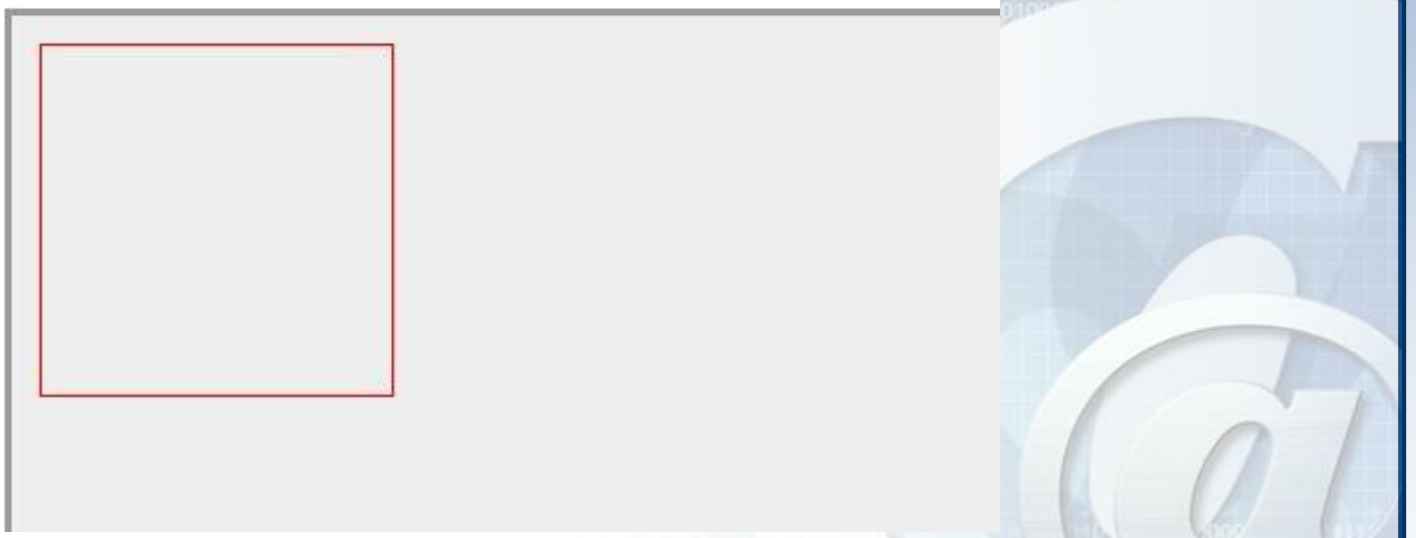
Add your homepage: Submit

The HTML `<canvas>` element is used to draw graphics on a web page.

The graphic to the left is created with `<canvas>`. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

```
<!DOCTYPE HTML>

<html>
   <head>

      <style>
         #mycanvas{border:1px solid red;}
      </style>
   </head>

   <body>
      <canvas id = "mycanvas" width = "100" height = "100"></canvas>
   </body>
</html>
```

This will produce the following result –



- Some elements have built-in interactivity, meaning they **do something** when you interact with them. A common example of this is links, which respond when you hover or click, but there are some new elements that offer other kinds of interactivity.

4.11 Interactive elements
- 4.11.1 The `details` element
- 4.11.2 The `summary` element
- 4.11.3 The `command` element
- 4.11.4 The `menu` element
  - 4.11.4.1 Introduction
  - 4.11.4.2 Building menus and toolbars
  - 4.11.4.3 Context menus
  - 4.11.4.4 Toolbars

Details & Summary

Created for what is traditionally referred to as an "accordion" widget, the paired `details` and `summary` elements can be used to show a snippet introductory or "summary" content that can be expanded / toggled open to show further "detail" content. These elements are useful for scenarios like a FAQ where users need to be able to quickly scan and parse a large amount of information.

```
<details>
    <summary>How much wood could a woodchuck chuck?
</summary>

    <p>A woodchuck would chuck as much wood as a
woodchuck could chuck</p>

</details>
```

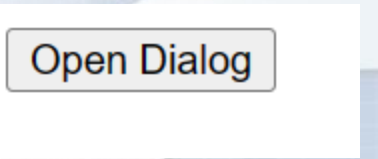
▶ How much wood could a woodchuck chuck?

Dialog

The new `dialog` element is for pop-up content in what would traditionally be referred to as a modal or lightbox. when open, it overlays the normal page content.

The `dialog` element is hidden by default, but can be set to display using the `open` attribute like so: `<dialog open>` . The typical way to toggle a `dialog` open and closed is to use a little javascript, attaching the open event to a button or other element.

```
<button id="open-dialog" type="button">Open
Dialog</button>

<dialog id="dialog-demo">
    Dialog content!
</dialog>

<!-- This should come just before the closing body
tag -->
<script>
    const modal = document.getElementById("dialog-
demo");
    const open = document.getElementById("open-
dialog");

    open.addEventListener("click", () => {
        modal.showModal();
```



The command element represents a command that the user can invoke.
A command can be part of a context menu or toolbar, using the menu element,
or can be put anywhere else in the page, to define a keyboard shortcut.
The type attribute indicates the kind of command:
either a normal command with an associated action,
or a state or option that can be toggled,
or a selection of one item from a list of items.

Here is an example of a toolbar with three buttons that let the user toggle between left, center, and right alignment. One could imagine such a toolbar as part of a text editor. The toolbar also has a separator followed by another button labeled "Publish", though that button is disabled.

```
<menu type="toolbar">
 <command type="radio" radiogroup="alignment" checked="checked"
          label="Left" icon="icons/alL.png" onclick="setAlign('left')">
 <command type="radio" radiogroup="alignment"
          label="Center" icon="icons/alC.png" onclick="setAlign('center')">
 <command type="radio" radiogroup="alignment"
          label="Right" icon="icons/alR.png" onclick="setAlign('right')">
 <hr>
 <command type="command" disabled
          label="Publish" icon="icons/pub.png" onclick="publish()">
</menu>
```

The `menu` element represents a list of commands.
The `type` attribute is an enumerated attribute indicating the kind of menu being declared.
The attribute has three states.
The `context` keyword maps to the **context menu** state, in which the element is declaring a context menu.
The `toolbar` keyword maps to the **toolbar** state, in which the element is declaring a toolbar.
The attribute may also be omitted.
The *missing value default* is the **list** state,
which indicates that the element is merely a list of commands that is neither
declaring a context menu nor defining a toolbar.

If a menu element's type attribute is in the context menu state, then the element represents
the commands of a context menu, and the user can only interact with the commands
if that context menu is activated.
If a menu element's type attribute is in the toolbar state, then the element represents
a list of active commands that the user can immediately interact with.
If a menu element's type attribute is in the list state, then the element either represents
an unordered list of items (each represented by an li element),
each of which represents a command that the user can perform or activate, or,
if the element has no li element children, flow content describing available commands.
The label attribute gives the label of the menu. It is used by user agents to display
nested menus in the UI. For example, a context menu containing another menu
would use the nested menu's label attribute for the submenu's menu label.
The type and label IDL attributes must reflect the respective content attributes of the

The menu element is used to define context menus and toolbars.

For example, the following represents a toolbar with three menu buttons on it, each of which has a dropdown menu with a series of options:

```
<menu type="toolbar">
 <li>
  <menu label="File">
   <button type="button" onclick="fnew()">New...</button>
   <button type="button" onclick="fopen()">Open...</button>
   <button type="button" onclick="fsave()">Save</button>
   <button type="button" onclick="fsaveas()">Save as...</button>
  </menu>
 </li>
 <li>
  <menu label="Edit">
   <button type="button" onclick="ecopy()">Copy</button>
   <button type="button" onclick="ecut()">Cut</button>
   <button type="button" onclick="epaste()">Paste</button>
  </menu>
 </li>
 <li>
  <menu label="Help">
   <li><a href="help.html">Help</a></li>
   <li><a href="about.html">About</a></li>
  </menu>
 </li>
</menu>
```

In a supporting user agent, this might look like this:

Thank You !

FORMs in HTML

END of CHAPTER 7

http://staff.emu.edu.tr/raygankansoy/en/teaching/itec229