HTML Basics…

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML

- **markup language** that makes up most web pages and online applications.

- A **hypertext** is a text that is used to reference other pieces of text, while a **markup language** is a series of markings that tells web servers the style and structure of a document.

- Not considered a programming language as it can’t create dynamic functionality.

- With HTML, web users can create and structure sections, paragraphs, and links using elements, tags, and attributes.

How does HTML work?

- An average website includes several different HTML pages.
- HTML documents are files that end with a **.html** or **.htm** extension.
- A web browser reads the HTML file and renders its content so that internet users can view it.
- All HTML pages have a series of **HTML elements**, consisting of a set of **tags** and **attributes**.

- **HTML elements** are the building blocks of a web page.
- A tag tells the web browser where an element begins and ends, whereas an attribute describes the characteristics of an element.

The three main parts of an HTML element are:

- **Opening tag –** used to state where an element starts to take effect. The tag is wrapped with opening and closing angle brackets.

For example, use the start tag **<p>** to create a paragraph.

- **Content –** this is the output that other users see.

- **Closing tag** – the same as the opening tag, but with a forward slash before the element name.

For example, **</p>** to end a paragraph.

The combination of these three parts will create an HTML element:

<**p**>This is how you add a paragraph in HTML.</**p**>

- Another critical part of an HTML element is its **attribute**, which has two sections – a **name** and **attribute value**.

The name identifies the additional information that a user wants to add, while the attribute value gives further specifications.

For example, a style element adding the color purple and the font-family **verdana** will look like this:

<**p style**="color:purple; font-family:verdana">This is how you add a paragraph in HTML.</**p**>

- Another attribute, the **HTML class**, The class attribute adds style information that can work on different elements with the same class value.

For example,

we will use the same style for a heading **<h1>** and a paragraph **<p>**.

The style includes background color, text color, border, margin under the class **.important**.

To achieve the same style between **<h1>** and **<p>**, add **class=”important”** after each start tag:

```
<html>
<head>
<style>
.important {
background-color: blue;
color: white;
border: 2px solid black;
margin: 2px;
}
</style>
</head>
<body>
<h1 class="important">This is a heading</h1>
<p class="important">This is a paragraph.</p>
</body>
</html>
```

- Most elements have an opening and a closing tag, but some elements do not need closing tags to work, such as **empty elements**.

These elements do not use an end tag because they do not have content:

<**img src**="/" **alt**="Image">

This image tag has two attributes – an **src** attribute, the image path, and an **alt** attribute, the descriptive text. However, it does not have content nor an end tag.

Every HTML page uses these three tags:

<html> tag is the root element that defines the whole HTML document.

<head> tag holds meta information such as the page's title and charset.

<body> tag encloses all the content that appears on the page.

```
<html>
<head>
<!-- META INFORMATION -->
</head>
<body>
<!-- PAGE CONTENT -->
</body>
</html>
```

Currently, there are 142 HTML tags available that allow for the creation of various elements.

Even though modern browsers no longer support some of these tags, learning all the different elements available is still beneficial.

Examples:

- **Heading tags** – these range from **<h1>** to **<h6>**, where heading h1 is largest in size, getting smaller as they move up to h6.
- **Paragraph tags** – are all enclosed by using the **<p>** tag.
- **List tags** – have different variations. Use the **<ol>** tag for an ordered list, and use **<ul>** for an unordered list. Then, enclose individual list items using the **<li>** tag.

- a **<strong>** tag would render an element in bold, whereas the **<em>** tag would show it in italics.

- Hyperlinks are also inline elements that use an **<a>** tag and an **href** attribute to indicate the link’s destination:

```
<a href="https://example.com/">Click me!</a>
```

HTML examples

1.html

2.html

3.html

4.html

5.html

6.html

7.html

8.html

9.html

10.html

11.html

12.html

- What does HTML stand for?
- Who is making the Web standards?
- Choose the correct HTML element for the largest heading:

a. <heading>

b. <h1>

c. <h6>

d. <head>

- What is the correct HTML element for inserting a line break?

a. <br>

b. <break>

c. <lb>

- What is the correct HTML for adding a background color?

a. <body bg=“yellow”>

b. <body style=“background-color:yellow”>

c. <background>yellow</background>

- Correct HTML element for strong text?
- Correct HTML element to define emphasized text?
- What is the correct HTML for creating a hyperlink?

  a. <a href="http://www.w3schools.com">W3Schools</a>

  b. <a url="http://www.w3schools.com">W3Schools</a>

  c. <a> http://www.w3schools.com</a>

  d. <a name="http://www.w3schools.com">W3Schools</a>
- Which character is used to indicate an end tag?
- How can you make a numbered list?

  a. <dl>   b. <ol>   c. <list>   d. <ul>
- How can you make bulleted list?
- What is the correct HTML for making a text input field?

  a. <textfield>

  b. <input type="text">

  c. <textinput type="text">

  d. <input type="textfield">

HTML5…

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML stands for Hyper Text Markup Language and it is the standard markup language for creating **web pages** and **web applications**. It is used to describe the structure of Web pages using markup.

HTML5 is the 5th version of HTML and is a core technology markup language of internet that is used for **structuring** and **presenting** the content for **www.**

HTML Evolution – What Differs Between HTML and HTML5?

- The first version of HTML consisted of 18 tags. Since then, each new version came with new tags and attributes added to the markup.

- The most significant upgrade of the language so far was the introduction of **HTML5** in 2014.

- The main difference between HTML and HTML5 is that HTML5 **supports new kinds of form controls.**

- HTML5 also introduced several semantic tags that clearly describe the content, such as **<article>**, **<header>**, and **<footer>**.

- HTML5 is the latest standard for HTML
- It was specially designed to deliver rich content without the need for additional plugins.
- HTML5 is also cross-platform. It is designed to work whether you are using a PC, or a Tablet, a Smartphone, or a Smart TV

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Differences between HTML and HTML5

HTML	HTML5
It didn’t support audio and video without the use of flash player support.	It supports audio and video controls with the use of <audio> and <video> tags.
Not possible to draw shapes like circle, rectangle, triangle etc.	HTML5 allows to draw shapes like circle, rectangle, triangle etc.
It does not allow drag and drop effects.	It allows drag and drop effects.
It works with all old browsers.	It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc.
Older version of HTML are less mobile-friendly.	HTML5 language is more mobile-friendly.
Elements like nav, header were not present.	New element for web structure like nav, header, footer etc.
Doctype declaration and character encoding is too long and complicated.	Doctype declaration and character encoding is quite simple and easy.

A Minimum HTML5 Document…

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>
<body>
Content of the document......
</body>
</html>
```

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

HTML <!DOCTYPE> Declaration

- Every HTML document must start with a **<!DOCTYPE>** declaration to inform the web browser about the document type.
- All HTML documents must start with a <!DOCTYPE> declaration.
- The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

In HTML 5, the declaration is simple:

<!DOCTYPE html> (case insensitive)

- HTML5 uses a DOCTYPE declaration which is very short, due to its lack of references to a DTD in the form of a URL, All it contains is the tag name of the root element of the document, HTML.

- In older documents (HTML 4 or XHTML), the declaration is more complicated because the declaration must refer to a DTD (Document Type Definition – consists of list of validated elements and attributes and to conform to a particular syntax).

HTML 4.01:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

XHTML 1.1:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<meta charset="UTF-8">

- The charset attribute specifies the character encoding for the HTML document.
- The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world!
- <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

The HTML <head> Element

- The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.
- HTML metadata is data about the HTML document.
- Metadata is not displayed.
- Metadata typically define the document title, character set, styles, scripts, and other meta information.

The HTML <body> Element

- The <body> tag defines the document's body.

- The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

Note: There can only be one <body> element in an HTML document.

HTML5 - New Features…

- Some of the most interesting new features in HTML5 are:
  - New content-specific elements, like <article>, <footer>, <header>, <nav>, <section>.
  - The <canvas> element for 2D drawing.
- The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.
- The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

- The <video> and <audio> elements for media playback.

Used to show video and play audio files on web page.

- Support for local storage

HTML5 local storage is a component of the Web storage application programming interface. It is a **method by which Web pages locally store named key/value pairs inside a client's Web browser**.

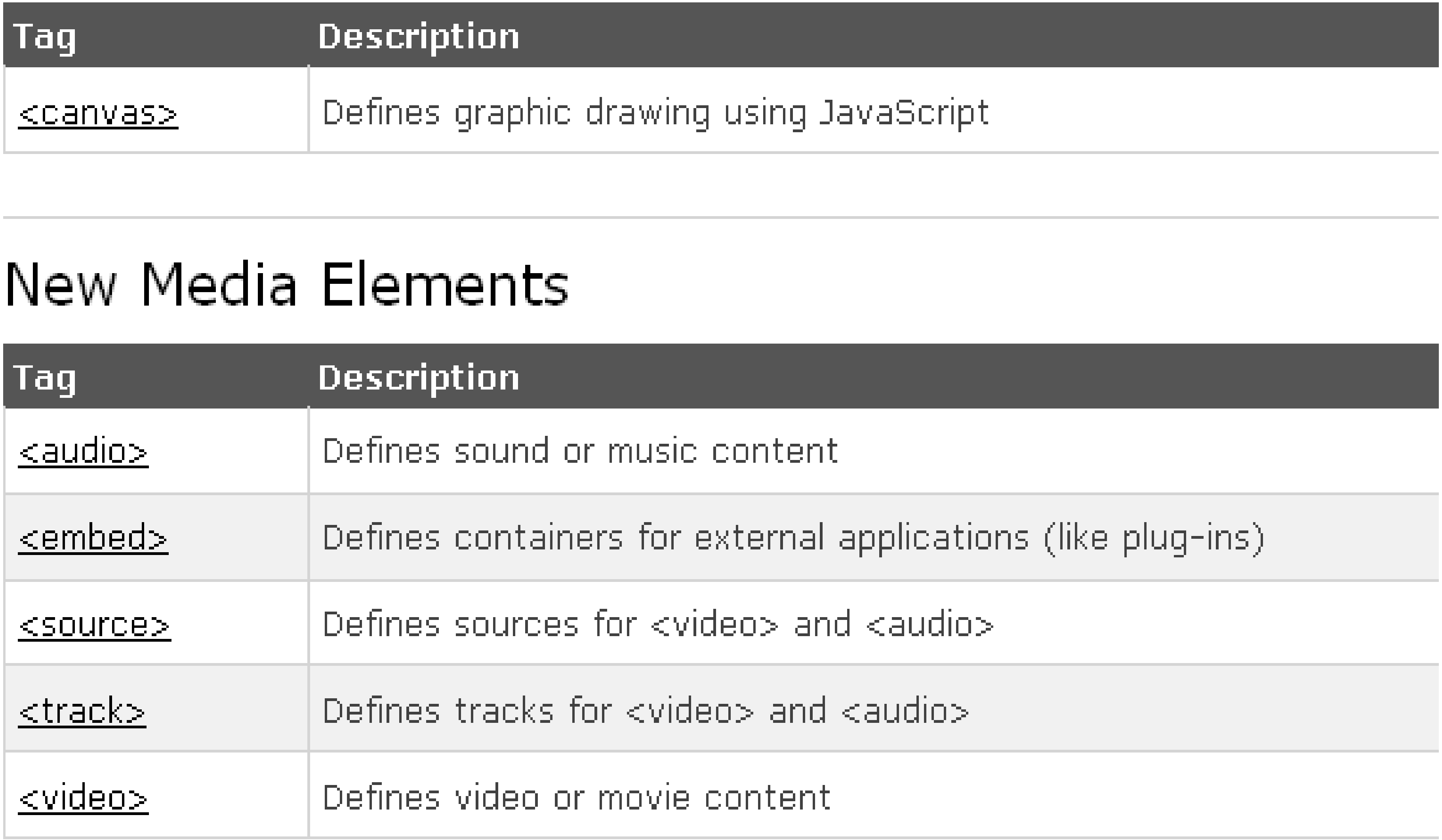
- New form controls, like calendar, date, time, email, url, search.

Browser Support for HTML5…

All major browsers (Chrome, Firefox, Internet Explorer, Safari, Opera) support the new HTML5 elements and APIs, and continue to add new HTML5 features to their latest versions

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

New Elements in HTML5…

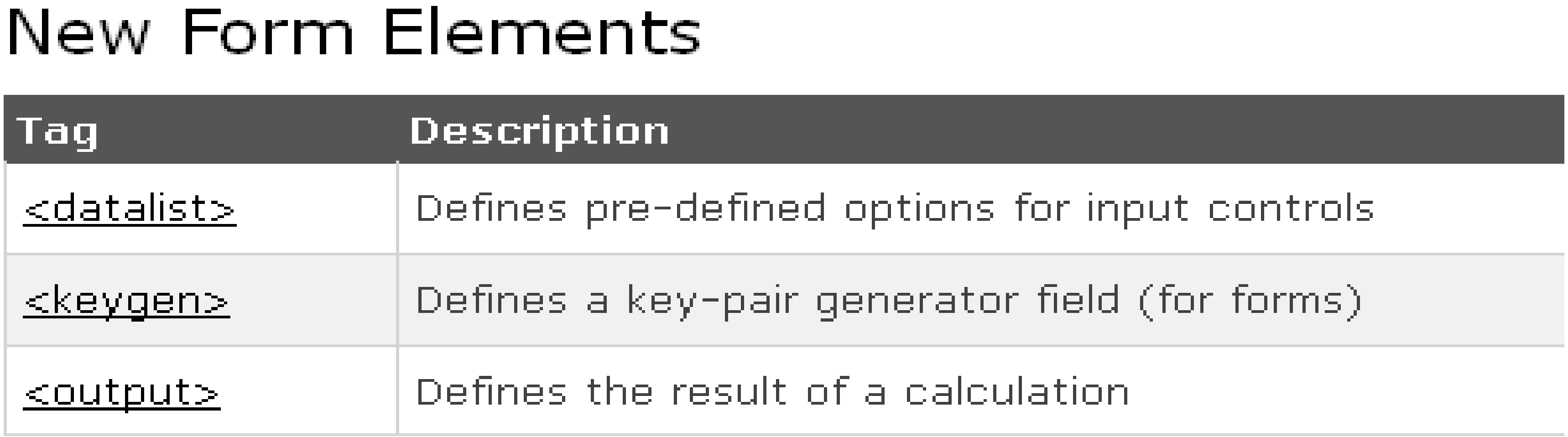


Tag	Description
<canvas>	Defines graphic drawing using JavaScript

New Media Elements

Tag	Description
<audio>	Defines sound or music content
<embed>	Defines containers for external applications (like plug-ins)
<source>	Defines sources for <video> and <audio>
<track>	Defines tracks for <video> and <audio>
<video>	Defines video or movie content

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014



New Form Elements

Tag	Description
<datalist>	Defines pre-defined options for input controls
<keygen>	Defines a key-pair generator field (for forms)
<output>	Defines the result of a calculation

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

New Semantic Elements in HTML5…

- Many of existing web sites today contains HTML code like this: <div id="nav">, <div class="header">, or <div id="footer">, to indicate navigation links, header, and footer

- HTML5 offers new semantic elements to clearly define different parts of a web page.

Note: The div tag is known as Division tag.

The div tag is **used in HTML to make divisions of content in the web page** like (text, images, header, footer, navigation bar, etc).

It is used to the group of various tags of HTML so that sections can be created and style can be applied to them.

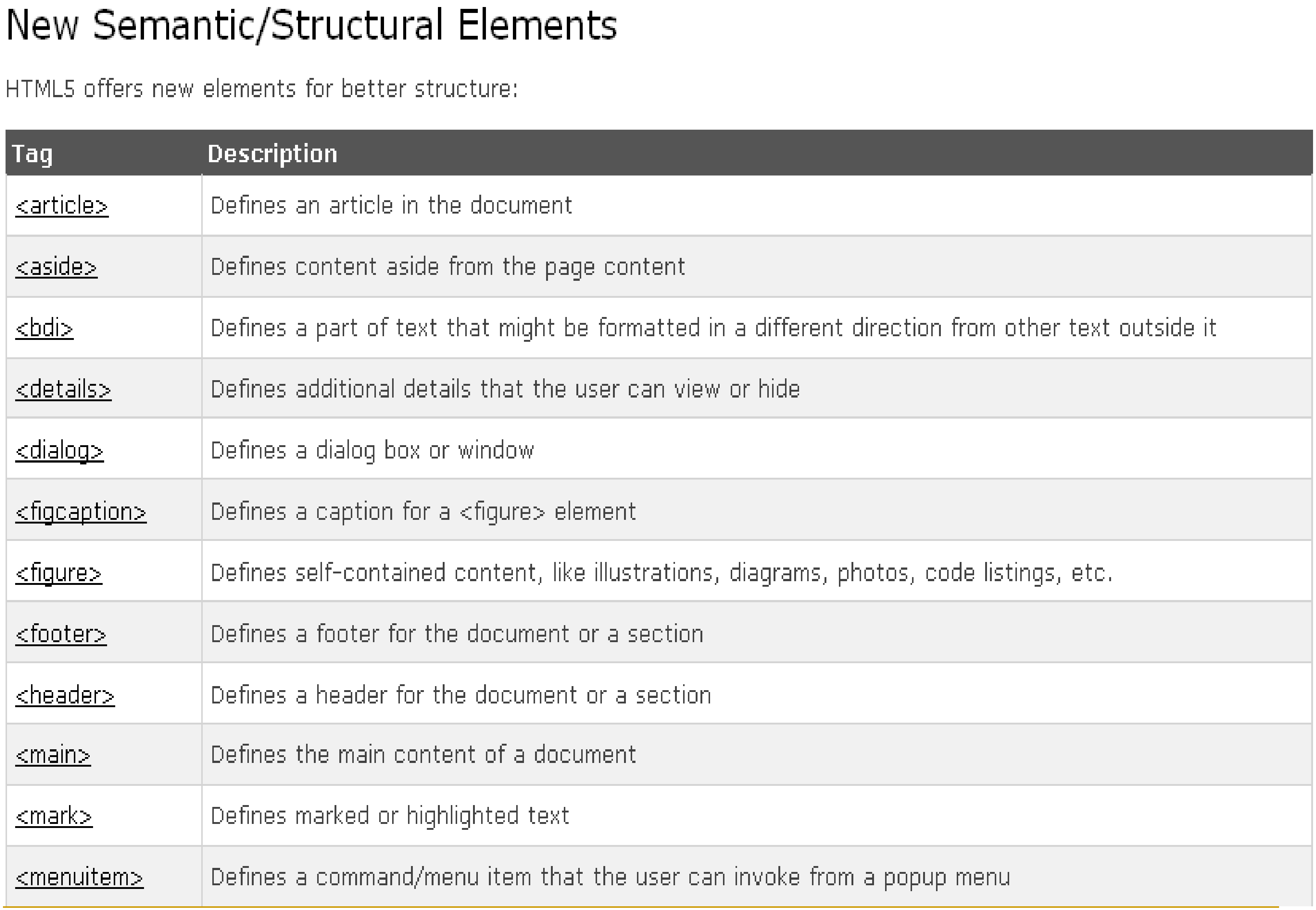
Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

What are semantic elements?

- Semantic elements clearly describes its meaning to both browser and the developer.
- Defines purpose of the element. They describe meaning of element to browser instead of just displaying it.
- By providing this extra level of clarity, HTML5 semantic elements also **help search engines to read the page and find** the required information faster.

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

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

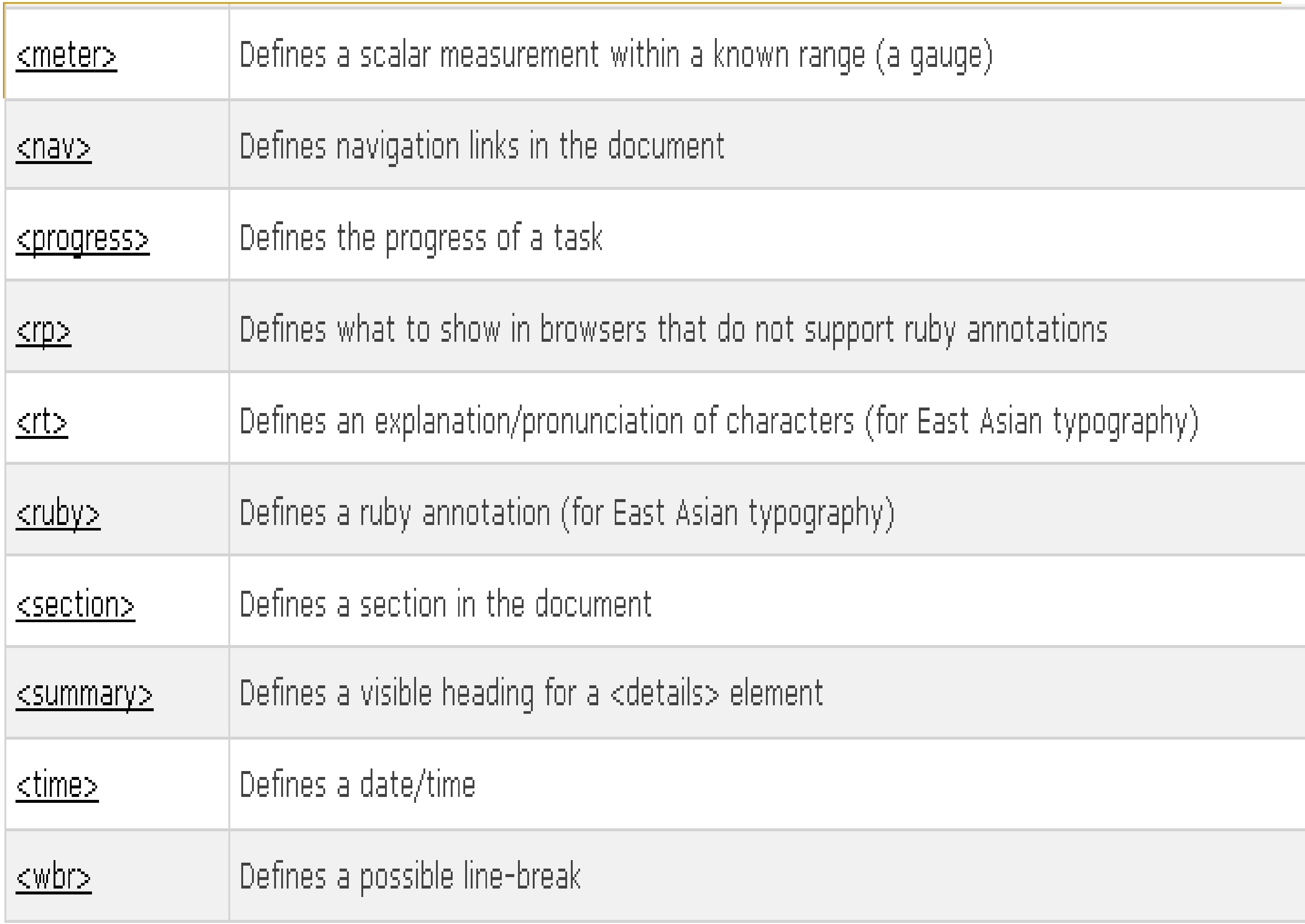


New Semantic/Structural Elements

HTML5 offers new elements for better structure:

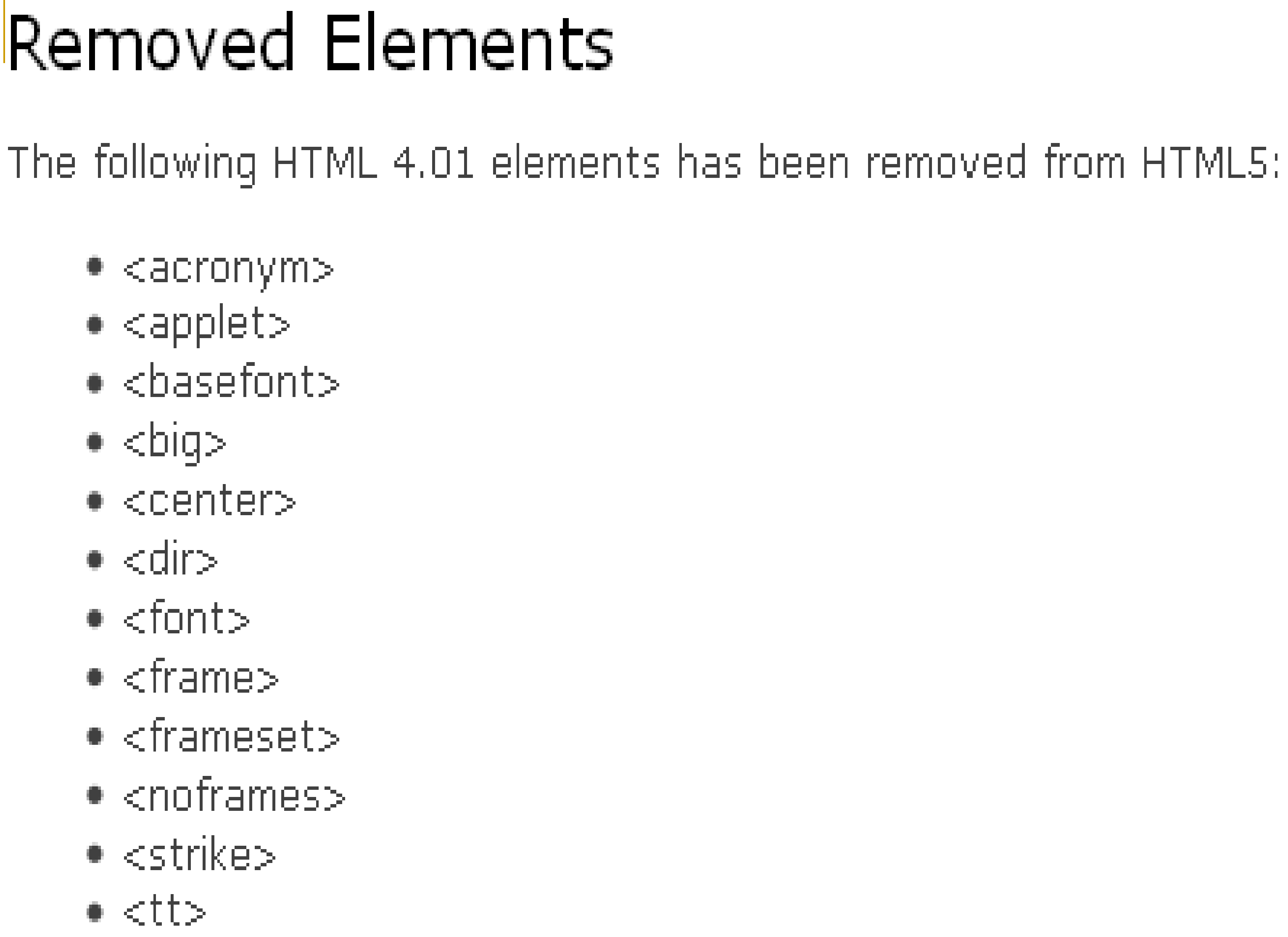
Tag	Description
<article>	Defines an article in the document
<aside>	Defines content aside from the page content
<bdi>	Defines a part of text that might be formatted in a different direction from other text outside it
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for the document or a section
<header>	Defines a header for the document or a section
<main>	Defines the main content of a document
<mark>	Defines marked or highlighted text
<menuitem>	Defines a command/menu item that the user can invoke from a popup menu

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014



<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links in the document
<progress>	Defines the progress of a task
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<section>	Defines a section in the document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

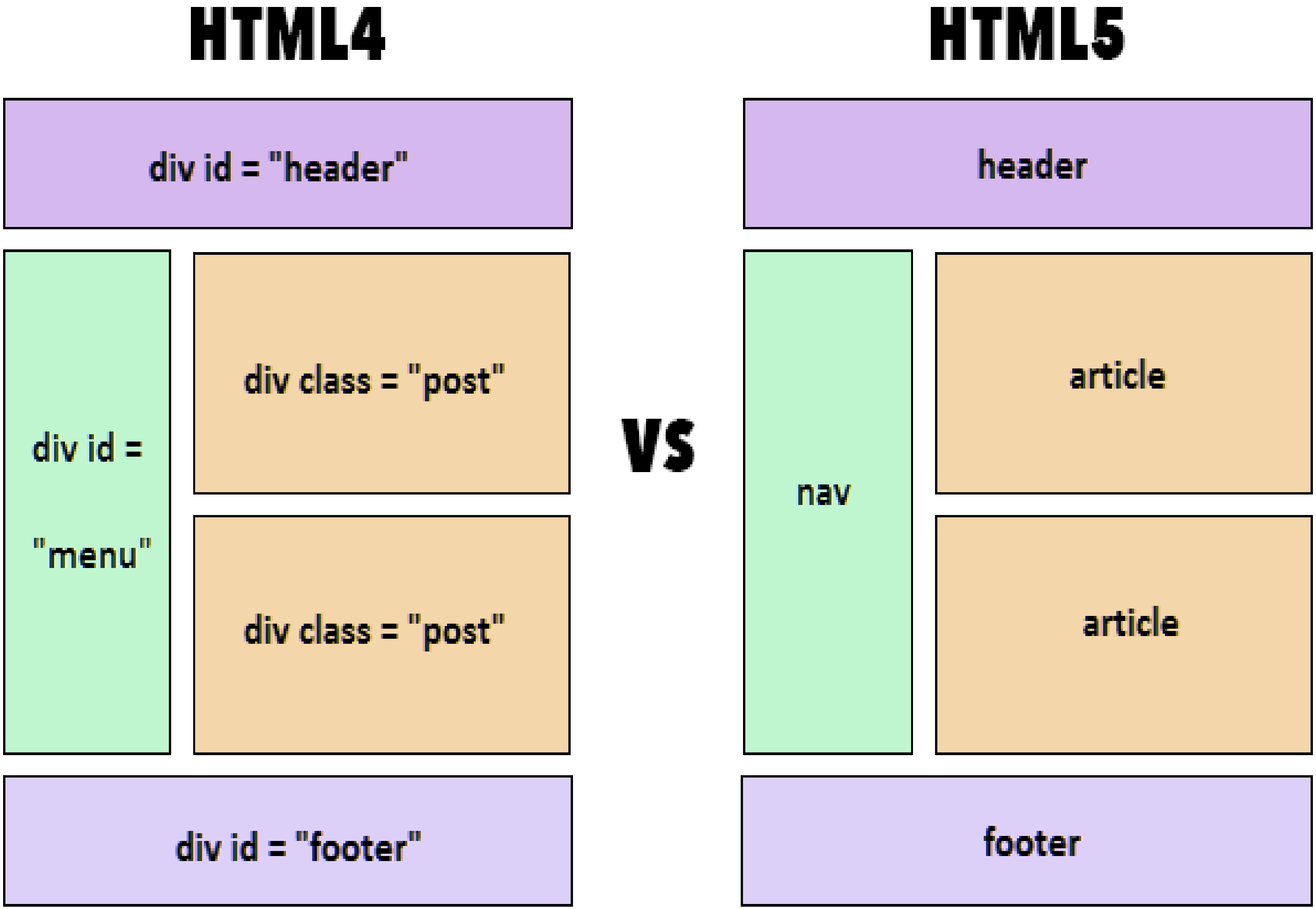


Removed Elements

The following HTML 4.01 elements has been removed from HTML5:

- <acronym>
- <applet>
- <basefont>
- <big>
- <center>
- <dir>
- <font>
- <frame>
- <frameset>
- <noframes>
- <strike>
- <tt>

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014



HTML4

div id = "header"

div id =

"menu"

div class = "post"

div class = "post"

div id = "footer"

VS

HTML5

header

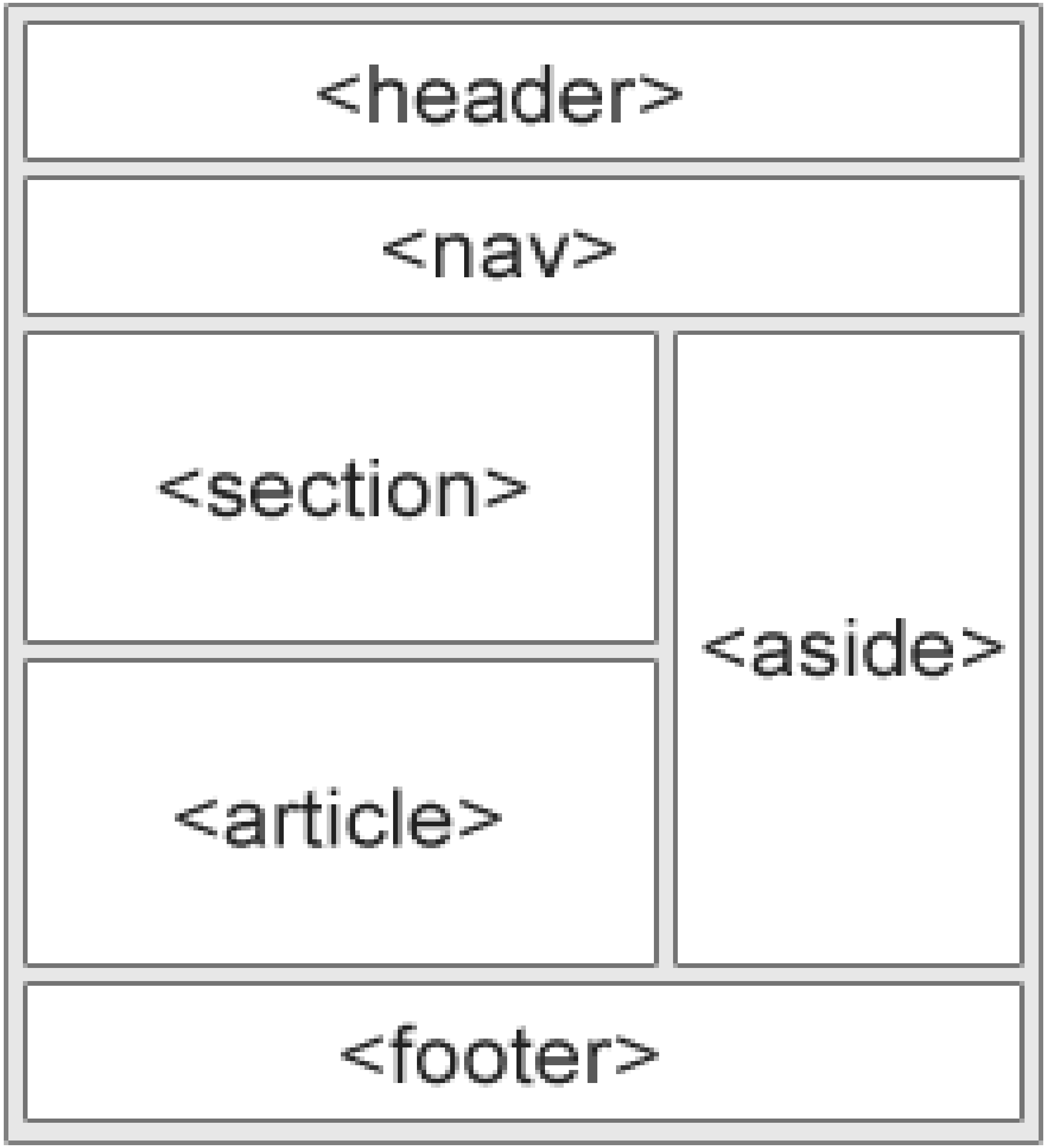
nav

article

article

footer

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



Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 <section> Element…

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

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Example: (Refer Ex1.html)

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

HTML5 <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 independently from rest of website.

- Examples of where an <article> element can be used:

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

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

- Example (Refer Ex2.html)

```
<article>

    <h1>Internet Explorer 9</h1>
     <p>Windows Internet Explorer 9 (abbreviated as IE9) was
 released to the  public on March 14, 2011 at 21:00 PDT.....
    </p>

</article>
```

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

HTML5 <header> Element…

- The <header> element specifies a header for a document or section

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

- A <header> element typically contains:
  - One or more heading elements (<h1> - <h6>)
  - Logo or icon
  - Authorship information

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

But <header> cannot be placed within a <footer>, <address> or another <header> element.

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

Refer Ex3.html

```
<article>

 <header>
    <h1>Internet Explorer 9</h1>
    <p><time pubdate datetime="2011-03-15"></time></p>
  </header>

  <p>Windows Internet Explorer 9 (abbreviated as IE9) was released
to the  public on March 14, 2011 at 21:00 PDT.....</p>
</article>
```

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

HTML5 <footer> Element…

- The <footer> element specifies a footer for a document or section

- A footer typically contains the
  - author of the document,
  - copyright information,
  - contact information,
  - back to top links
  - Related documents
  - sitemap

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

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

Refer Ex4.html

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p><time pubdate datetime="2012-03- 01"></time></p>
</footer>
```

//datetime attribute translates time to machine readable format

//pubdate attribute indicates publication date of document.

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 <nav> Element…

- The <nav> element defines a set of navigation links
- The <nav> element is intended for large blocks of navigation links.
- However, not all links in a document should be inside a <nav> element!

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Example:

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

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 <figure> and <figcaption> Elements…

- The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc

- The <figcaption> tag defines a caption for a <figure> element

- The <figcaption> element can be placed as the first or last child of the <figure> element

- The <img> element defines the actual image/illustration.

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Refer Ex5.html

```
<figure>
  <img src="img_pulpit.jpg" alt="The Pulpit Rock"     width="304"
height="228">
  <figcaption>Fig1. - The Pulpit Pock, Norway.</figcaption>
</figure>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML <aside> element

- The <aside> element defines some content aside from the content it is placed in (Like a sidebar).
- The <aside> content should be indirectly related to surrounding content.

HTML <summary> tag:

Defines a visible heading for the <details> element. The heading can be clicked to view/hide the details.

It should be first child element of <details>

```
<details>
 <summary> Epcot Center </summary>
 <p> ………………………………….. </p>
</details>
```

HTML <mark> tag:

Defines text that should be marked or highlighted.

<p> Do not forget to buy <mark> Choclolates </mark> today. </p>

HTML <time> tag:

Defines a specific time.

The datetime attribute of this element is used to translate the time into machine-readable format so that browsers can offer to add date reminders through users calendar and search engines and produce smart results.

<p> I have a date on<time datetime=“2021-02-14 20:00”>Valentines Day</time>.</p>

Q 1 - An article element contains?

A - Text or embedded content

B - Image

C - audio

D - none

Q2 - Which of the following is true about HTML 5?

A - HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1.

B - HTML5 is a standard for structuring and presenting content on the World Wide Web.

C - HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

D - All of the above.

Q 3: Which of the following tag is used to represent the result of different types of output in HTML5?

A - output

B - placeholder

C - autofocus

D - required

Q4: Which browser does the HTML5 supports?

A – Firefox

B – Safari

C – Internet Explorer

D – All the mentioned above

Q 5: The new form elements are introduced in HTML5 is:

A – 2

B – 3

C – 4

D – 5

Q 6: ------------------ is not a HTML5 element.

A - <footer>

B - <article>

C - <wrapper>

D – <figcaption>

Q 7: The most interesting new APIs in HTML5 are:

A – HTML Geolocation and HTML Application Cache

B – HTML Drag and Drop and HTML Web Workers

C – HTML Local Storage and HTML SSE

D – All of the above

Q 8: **Which of the following is true about 'audio' tag in HTML5?**

A - HTML5 supports <audio> tag which is used to embed sound content in an HTML or XHTML document.

B - The current HTML5 draft specification does not specify which audio formats browsers should support in the audio tag.

C - Both of the above.

D - None of the above.

Q 9: Which of the following attribute triggers event when the window loses focus?

A - onbeforeload

B - onblur

C - onlostfocus

D - lostfocus

Q 10: Which of the following attribute triggers event when the browser starts to load the media data?

A - onloadedmetadata

B - onloadstart

C - onmessage

D - onoffline

Q1: A

Q2: D

Q3: A

Q4: D

Q5: B

Q6: C

Q7: D

Q8: C

Q9: B

Q10: B

https://www.onlineinterviewquestions.com/html5-mcq/

https://www.interviewbit.com/html-interview-questions/

HTML forms

- An HTML form is used to collect user input. The user input is most often sent to a server for processing.
- All of the controls of a form appear in the content of a <form> tag.
- <form>, can have several different attributes, only one of which, action, is required.
- The action attribute specifies the URL of the application on the Web server that is to be called when the user clicks the *Submit* button.

The HTML <input> element

- The **<input>** HTML element is used to create interactive controls for web-based forms in order to accept data from the user;

- A wide variety of types of input data and control widgets are available.

- The <input> element is one of the most powerful and complex in all of HTML due to the sheer number of combinations of input types and attributes.

- Many of the commonly used controls are specified with the inline tag <input>, including those for text, passwords, checkboxes, radio buttons, and the action buttons *Reset*, *Submit*

Text Box

- It is a type of input which takes the text. Any type of input can be created using <input> tag.

- The ***type* attribute** indicates what type of input is needed for the text box, the value should be given as text. If type attribute is not specified, the default type is text.

- For any type of input, a name has to be provided which is done using ***name* attribute**.
- This is used to reference elements in a Javascript, or to reference form data after a form is submitted.

The ***value* attribute** is used differently for different input types:

- For "button", "reset", and "submit" - it defines the text on the button
- For "text", "password", and "hidden" - it defines the initial (default) value of the input field
- For "checkbox", "radio", "image" - it defines the value associated with the input (this is also the value that is sent on submit)

- The size of the text can be controlled using ***size*** **attribute**, indicates visible width, in characters, of an <input> element.

- Every browser has a limit on the number of characters it can collect. If this limit is exceeded, the extra characters are chopped off.

- To prevent this chopping, ***maxlength*** **attribute** can be used. When maxlength is used, users can enter only those many characters that is given as a value to the attribute.

```
<form action = " ">
 <p>
 <label>Enter your Name:
<input type = "text" name = "myname" size = "20" maxlength = "20" />
</label>
 </p>
</form>
```

Password Box

- Define a password field (characters are masked)
- If the contents of a text box should not be displayed when they are entered by the user, a password control can be used.
- In this case, regardless of what characters are typed into the password control, only bullets or asterisks are displayed by the browser.

```
<form action = " ">
 <p>
<label>Enter the email id:
<input type = "text" name = "myname" size = "24" maxlength = "25" />
</label> </p>
 <p>
<label>Enter the password:
 <input type = "password" name = "mypass" size = "20" maxlength = "20" />
 </label> </p>
</form>
```

Radio Button

- Radio buttons are special type of buttons which allows the user to select only individual option
- Radio buttons are created using the input tag with the *type* attribute having the value **radio.**
- When radio buttons are created, values must be provided with the help of *value* attribute.
- All the radio buttons which are created would have same name. This is because the radio buttons are group elements.
- If one of the radio buttons has to be selected as soon as the web page is loaded, checked attribute should be used. The value also would be checked.

```html
<form action = " ">
<p>
 <label>
<input type="radio" name="act" value="one"/>Puneeth Rajkumar</label>
<label><input type="radio" name="act" value="two"/>Sudeep</label>
<label><input type="radio" name="act" value="three"/>Darshan</label>
<label><input type="radio" name="act" value="four"/>ShivaRajkumar</label>
</p>
 </form>
```

Check Box

- Check box is a type of input using which multiple options can be selected.
- Check box can also be created using the <input> tag with the *type* having the value “checkbox”.
- During the creation of check box, the value should be provided using the *value* attribute.
- All the checkbox which are created would have the same name because they are group elements.
- If one of the check box have to be selected as soon as the page is loaded, checked attribute should be used with the value checked.

```
<form action = " ">
 <p>
 <label><input type="checkbox" name="act“
value="one"/>Ragini</label>
<label><input type="checkbox" name="act"
value="two"/>Ramya</label>
 <label><input type="checkbox" name="act"
value="three"/>Aindritha</label>
<label><input type="checkbox" name="act"
value="four"/>Radhika</label>
<label><input type="checkbox" name="act"
value="four"/>Rakshitha</label>
 </p>
 </form>
```

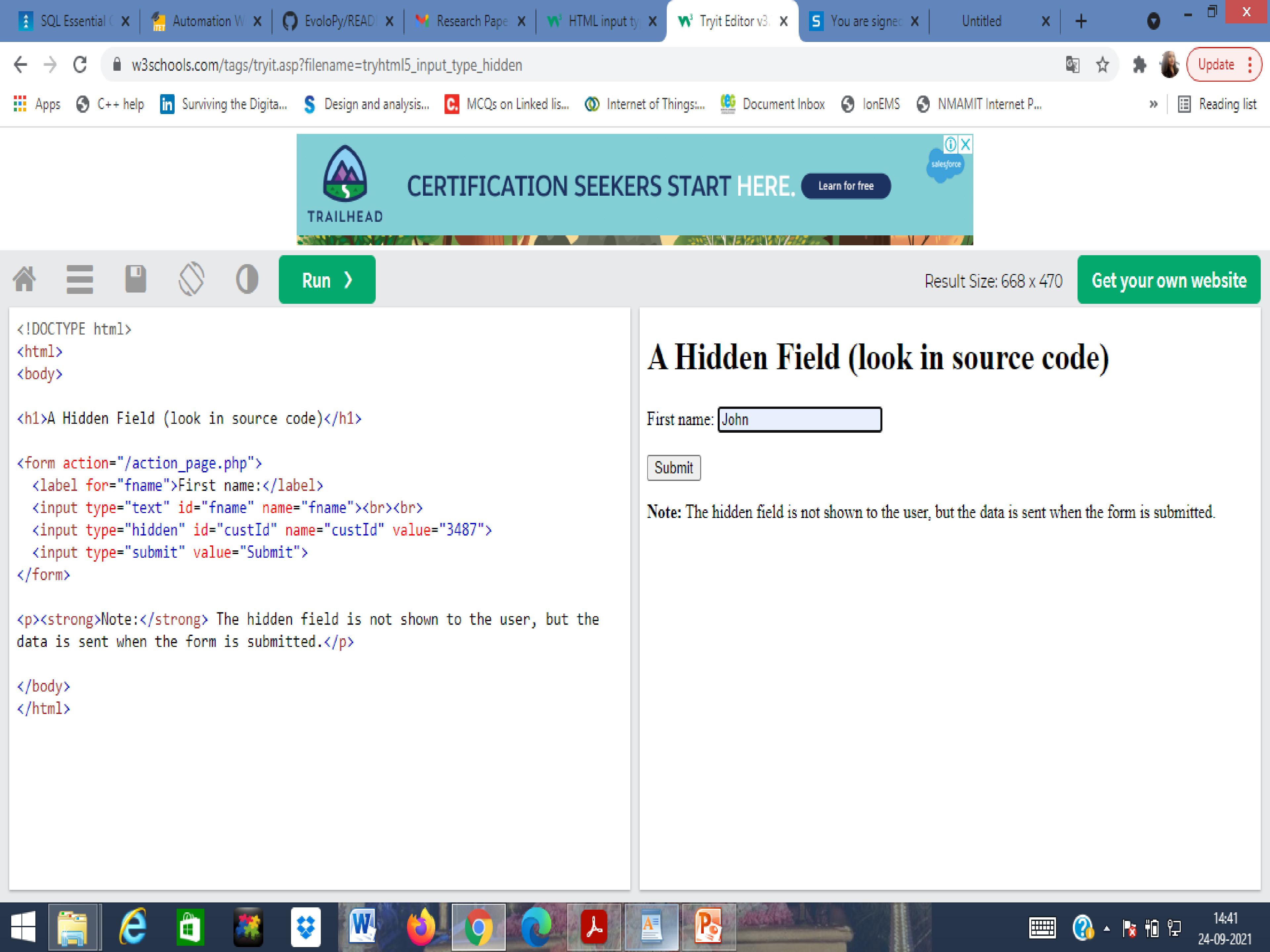
HTML <input type=“hidden”>

- The <input type="hidden"> defines a hidden input field.
- A hidden field let web developers include data that cannot be seen or modified by users when a form is submitted.
- A hidden field often stores what database record that needs to be updated when the form is submitted.

Note: While the value is not displayed to the user in the page's content, it is visible (and can be edited) using any browser's developer tools or "View Source" functionality. Do not use hidden inputs as a form of security!

Usage:

```
 <input type="hidden" id="custId" name="custId" value="3487">
```



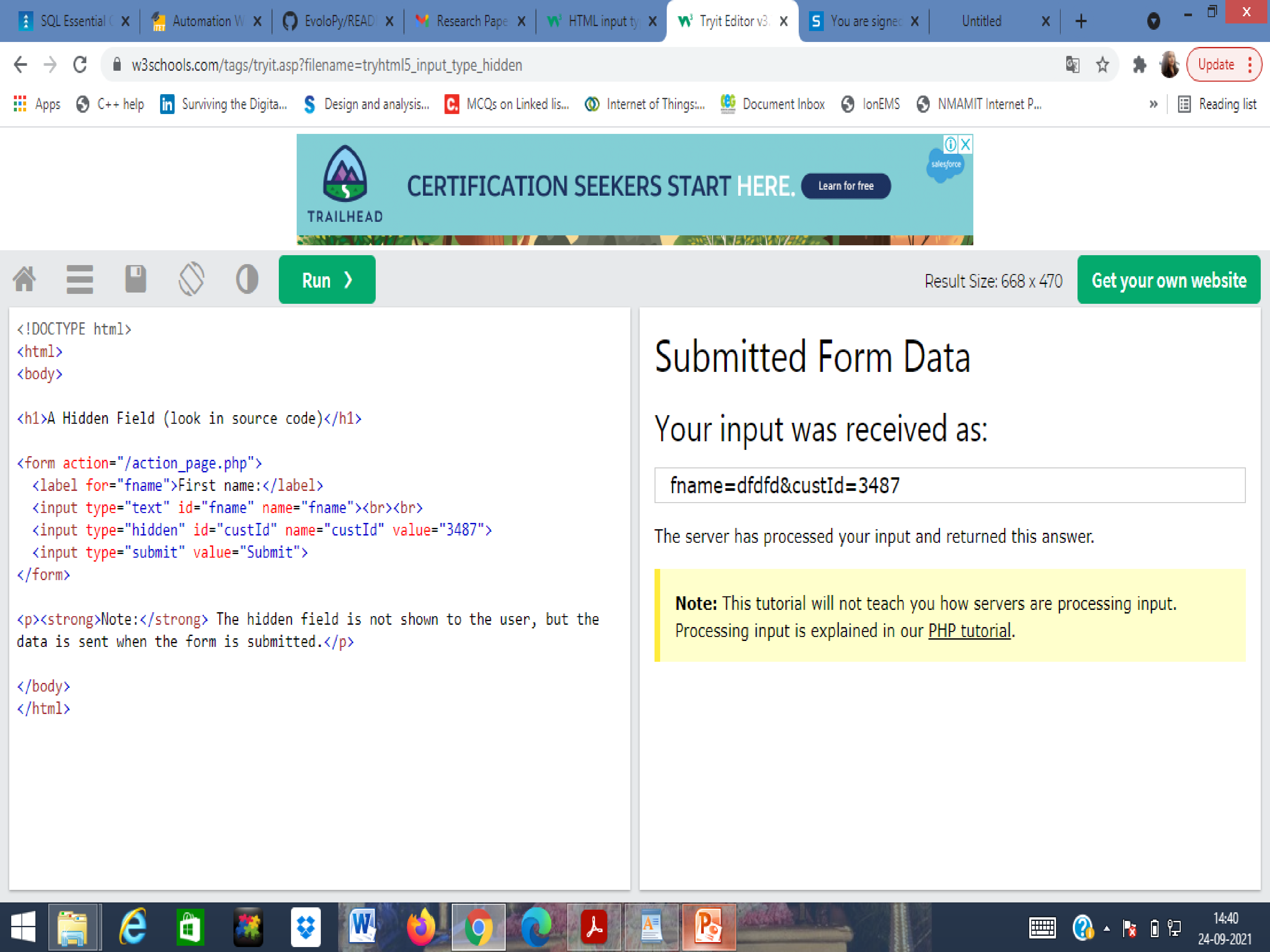
```
<!DOCTYPE html>
<html>
<body>

<h1>A Hidden Field (look in source code)</h1>

<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="custId" name="custId" value="3487">
  <input type="submit" value="Submit">
</form>

<p><strong>Note:</strong> The hidden field is not shown to the user, but the
data is sent when the form is submitted.</p>

</body>
</html>
```

A Hidden Field (look in source code)

First name: John

Submit

Note: The hidden field is not shown to the user, but the data is sent when the form is submitted.



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

<h1>A Hidden Field (look in source code)</h1>

<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="custId" name="custId" value="3487">
  <input type="submit" value="Submit">
</form>

<p><strong>Note:</strong> The hidden field is not shown to the user, but the
data is sent when the form is submitted.</p>

</body>
</html>
```

Submitted Form Data

Your input was received as:

```
fname=dfdfd&custId=3487
```

The server has processed your input and returned this answer.

Note: This tutorial will not teach you how servers are processing input. Processing input is explained in our PHP tutorial.

The <select> Tag:

- Menu items is another type of input that can be created on the page.
- To create the menu item, <select> tag is used.
- To insert the item in the menu, <option> tag is used.

```
<form action = "">
 <p>
 With size = 1 (the default)
<select name = "branches">
<option> Information Science </option>
<option> Computer Science </option>
<option> Electronics </option>
<option> Electrical </option>
<option> Mechanical </option>
 </select>
 </p>
</form>
```

The <textarea> Tag:

- Text area is a type of input using which multiple statements can be entered.
- Text area is created using <textarea> tag.
- Text area should have the name.
- During the creation of text area, it should be mentioned how many sentences can be entered. This is done using *rows* attribute.
- Similarly, it should also be mentioned how many characters can be entered in a line. This is done using *cols* attribute.
- If the value given to rows is exceeded i.e. if users enter sentences more than specified, the *scroll bar* automatically appears.

```
<form action=" ">
<h3> Enter your comments</h3>
 <p> <textarea name="feedback" rows="5" cols="100">
</textarea>
 </p>
 </form>
```

The Action Buttons:

- The *Reset* button clears all of the controls in the form to their initial states.

- The *Submit* button has two actions: First, the form data is encoded and sent to the server; second, the server is requested to execute the server-resident program specified in the action attribute of the <form> tag.

- The purpose of such a server-resident program is to process the form data and return some response to the user.

- Every form requires a *Submit* button. The *Submit* and *Reset* buttons are created with the <input> tag.

```
<form action=" ">
<p>
 <input type="SUBMIT" value="SUBMIT"/>
<input type="RESET" value="RESET"/>
</p>
 </form>
```

Complete HTML form:

HTMLform.html

HTML5 Input Types…

- HTML5 has several new input types for forms

- These new features allow better input control and validation
  - color
  - date
  - datetime
  - datetime-local
  - email
  - month
  - number
  - range
  - search
  - tel
  - time
  - url
  - week

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

HTML5 <input type="color">

- Defines a color picker.
- The default value is #000000 (black). The value must be in seven-character hexadecimal notation.

Usage:

```
<label>Select your favorite color:
<input type="color" name="favcolor“ value=“#ff0000”>
</label>
```

HTML5 <input type=“date">

- Defines a date picker.
- The resulting value includes the year, month, and day.

Usage:

```
<label> Birthday:
<input type="date" name="bday">
</label>
```

HTML5 <input type=“email">

- The <input type="email"> defines a field for an e-mail address.
- The input value is automatically validated to ensure it is a properly formatted e-mail address.
- To define an e-mail field that allows multiple e-mail addresses, add the "multiple" attribute.

Usage:

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

HTML5 <input type=“month”>

- Defines a month and year control.
- The format is "YYYY-MM".

Usage:

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

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 <input type=“number”>

- Define a field for entering a number (You can also set restrictions on what numbers are accepted)
- Use the following attributes to specify restrictions:
  - max - specifies the maximum value allowed
  - min - specifies the minimum value allowed
  - step - specifies the legal number intervals
  - value - Specifies the default value

Usage:

<label>Quantity (between 1 and 5): <input type="number" name="quantity" min="1" max="5"></label>

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 <input type=“range”>

- Defines a control for entering a number whose exact value is not important (like a slider control).
- Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the attributes below.

max - specifies the maximum value allowed

min - specifies the minimum value allowed

step - specifies the legal number intervals

value - Specifies the default value

Usage:

```
<label><input type="range" name="points" min="1"
max="10"></label>
```

HTML5 <input type=“search”>

- Define a text field for entering a search string (like a site search, or Google search).
- Remember to set a name for the search field, otherwise nothing will be submitted.

Usage:

```
<label>Search Google: <input type="search"
name="googlesearch"></label>
```

HTML5 <input type=“tel”>

- Defines a field for entering a telephone number.

Usage:

```
<label>Telephone: <input type="tel" name="usrtel"></label>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 <input type=“time”>

- Defines a control for entering a time (no time zone).

Usage:

<label>Select a time: <input type="time" name="usr_time"></label>

HTML5 <input type=“url”>

- Defines a field for entering a URL.
- The input value is automatically validated before the form can be submitted.

Usage:

```
<label>Add your homepage: <input type="url"
name="homepage"></label>
```

HTML5 <input type=“week”>

- Defines a week and year control (no time zone).

Usage:

```
<label> Select a week: <input type="week" name="week_year">
</label>
```

For all the new HTML5 input types illustration is as below:

Refer Ex6.html

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 New Form Elements…

HTML5 has the following new form elements:

- <datalist>
- <output>

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

- The <datalist> element specifies a list of pre-defined options for an <input> element
- Use the <input> element's list attribute to bind it together with a <datalist> 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.

Usage: Refer Ex7.html

```
<input list="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

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

Refer Ex9.html

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  <input type="range"  id="a" name="a" value="50">
  +
  <input type="number" id="b" name="b" value=“25">
  =
  <output name="x" for="a b"></output>
  <br><br>
</form>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

- The for attribute specifies the relationship between the result of the calculation, and the elements used in the calculation. It contains a list of elements on which the calculation has been performed for which the output element is rendering the result.

- The oninput attribute fires when the value of an **<input> or <textarea>** element is changed.

- The value property sets or returns the value of the value attribute of a text field.

HTML5 New Form Attributes…

- HTML5 has several new attributes for <form> and <input>

New attributes for <form>:

- autocomplete
- novalidate

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

New attributes for <input>:

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

<form> / <input> autocomplete Attribute…

- The ***autocomplete*** attribute specifies whether a form or input field should have autocomplete on or off

- When ***autocomplete*** is on, the browser automatically complete values based on values that the user has entered before

- It is possible to have ***autocomplete*** "on" for the form, and "off" for specific input fields, or vice versa

- The ***autocomplete*** attribute works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Refer Ex10.html

```
<form action="demo_form.asp"        autocomplete="on">
  First name:<input type="text"        name="fname"><br>
  Last name: <input type="text"        name="lname"><br>
  E-mail: <input type="email" name="email"
        autocomplete="off"><br>
  <input type="submit">
</form>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<form> novalidate Attribute…

- The novalidate attribute is a boolean attribute
- When present, it specifies that the form-data (input) should not be validated when submitted

```
<form action="demo_form.asp" novalidate>
        E-mail: <input type="email"      name="user_email">
  <input type="submit">
</form>
```

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

<input> autofocus Attribute…

- The autofocus attribute is a Boolean attribute

- When present, it specifies that an <input> element should automatically get focus when the page loads.

First name:<input type="text" name="fname" autofocus>

refer Ex11.html

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<input> form Attribute…

- The form attribute specifies one or more forms an <input> element belongs to.

- To refer to more than one form, use a space-separated list of form id’s

Example:

```
<form action="demo_form.asp" id="form1">
  First name: <input type="text" name="fname"><br>
  <input type="submit" value="Submit">
</form>

Last name: <input type="text" name="lname" form="form1">
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<input> formaction Attribute…

- The ***formaction*** attribute specifies the URL of a file that will process the input control when the form is submitted

- The ***formaction*** attribute overrides the action attribute of the <form> element

- The ***formaction*** attribute is used with type="submit" and type="image"

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Example:

```
<form action="demo_form.asp">
     First name: <input type="text"        name="fname"><br>
     Last name: <input type="text"         name="lname"><br>
     <input type="submit" value="Submit"><br>
     <input type="submit"          formaction="demo_admin.asp"
          value="Submit as admin">
   </form>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<input> formmethod Attribute…

- The ***formmethod*** attribute defines the HTTP method for sending form-data to the action URL
- The ***formmethod*** attribute overrides the method attribute of the <form> element
- The ***formmethod*** attribute can be used with type="submit" and type="image"

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Example:

```
<form action="demo_form.asp" method="get">
     First name: <input type="text"          name="fname"><br>
     Last name: <input type="text"           name="lname"><br>
     <input type="submit" value="Submit">
     <input type="submit" formmethod="post"
    formaction="demo_post.asp"
     value="Submit using POST">
    </form>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<input> formtarget Attribute…

- The formtarget attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form

- The formtarget attribute overrides the target attribute of the <form> element

- The formtarget attribute can be used with type="submit" and type="image"

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Refer Ex12.html

```
<form action="demo_form.asp">
  First name: <input type="text"        name="fname"><br>
  Last name: <input type="text"         name="lname"><br>
  <input type="submit" value="Submit as       normal">
  <input type="submit" formtarget="_blank"
        value="Submit to a new window">
</form>
```

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<form target="_blank|_self|_parent|_top">

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the same frame (this is default)
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window

<input> min and max Attributes…

- The min and max attributes specify the minimum and maximum value for an <input> element

- The min and max attributes works with the following input types: number, range, date, datetime, datetime-local, month, time and week

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

Examples:

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">
```

Quantity (between 1 and 5):

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

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<input> multiple Attribute…

- The multiple attribute is a boolean attribute
- When present, it specifies that the user is allowed to enter more than one value in the <input> element
- The multiple attribute works with the following input types: email, and file
- Select images: <input type="file" name="img" multiple>

Refer Ex13.html

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

<input> pattern Attribute…

- The pattern attribute specifies a regular expression that the <input> element's value is checked against

- The pattern attribute works with the following input types: text, search, url, tel, email, and password

  Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">

  Refer Ex14.html

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

<input> placeholder Attribute…

- The placeholder attribute specifies a short hint that describes the expected value of an input field (e.g. a sample value or a short description of the expected format)

- The short hint is displayed in the input field before the user enters a value

- The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

- <input type="text" name="fname" placeholder="First name">

  Refer Ex15.html

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<input> required Attribute

- The required attribute is a boolean attribute

- When present, it specifies that an input field must be filled out before submitting the form.

- The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file

Refer Ex16.html

Username: <input type="text" name="usrname" required>

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

<input> step Attribute…

- The step attribute specifies the legal number intervals for an <input> element

- Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

- The step attribute can be used together with the max and min attributes to create a range of legal values

- The step attribute works with the following input types: number, range, date, datetime, datetime-local, month, time and week.

Source: W3Schools and Robert W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson, 2014

- <input type="number" name="points" step="3">

  Refer Ex16.html

Source: W3Schools and Robert W. Sebesta, “Programming the World Wide Web”, Fourth Edition, Pearson, 2014

HTML5 Canvas…

- The HTML5 <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript)

- The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics

- Canvas has several methods for drawing paths, boxes, circles, text, and adding images

- You can have multiple <canvas> elements on one HTML page

```
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #000000;">
</canvas>
```

HTML5 Video…

- Before HTML5, videos could only be played with a plug-in (like flash). However, different browsers supported different plug-ins

- HTML5 defines a new element which specifies a standard way to embed a video or movie on a web page: the <video> element

Refer Ex18.html

```
<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>
```

- The control attribute adds video controls, like play, pause, and volume.

- You should also insert text content between the <video> and </video> tags for browsers that do not support the <video> element

- The <video> element allows multiple <source> elements. <source> elements can link to different video files

HTML5 Audio…

- HTML5 provides a standard for playing audio files

Refer Ex19.html

```
<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>
```

Sitesbay

Difference Between Html and Html5

Html	Html5
Doctype declaration in Html is too longer <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	DOCTYPE declaration in Html5 is very simple "<!DOCTYPE html>
character encoding in Html is also longer <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">	character encoding (charset) declaration is also very simple <meta charset="UTF-8">
Audio and Video are not part of HTML4	Audio and Videos are integral part of HTML5 e.g. <audio> and <video> tags.
Vector Graphics is possible with the help of technologies such as VML, Silverlight, Flash etc	Vector graphics is integral part of HTML5 e.g. SVG and canvas
It is almost impossible to get true GeoLocation of user browsing any website especially if it comes to mobile devices.	JS GeoLocation API in HTML5 helps identify location of user browsing any website (provided user allows it)
Html5 use cookies.	It provides local storage in place of cookies.
Not possible to draw shapes like circle, rectangle, triangle.	Using Html5 you can draw shapes like circle, rectangle, triangle.
Does not allow JavaScript to run in browser. JS runs in same thread as browser interface.	Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5
Works with all old browsers	Supported by all new browser.

Sitesbay.com

Sitesbay.com

- Further readings…
  - HTML5 Geolocation…
  - Drag and drop…
  - Web storage…

How Are HTML, CSS, and Javascript Related?

- HTML is used to add text elements and create the structure of content. However, it is not enough to build a professional and fully responsive website.

- So, HTML needs the help of Cascading Style Sheets (CSS) and JavaScript to create the vast majority of website content.

- CSS is responsible for stylings such as background, colors, layouts, spacing, and animations.

- On the other hand, JavaScript adds dynamic functionality such as sliders, pop-ups, and photo galleries.
- These three languages are the fundamentals of front-end development.

Thank You …