|  |  |  |
| --- | --- | --- |
| **1.** | The <strong> tag is used to define text with strong importance | The bold (<b>) tag specifies bold text without any extra importance. |

**This text is bold**

*This text is italic*

This issubscript and superscript

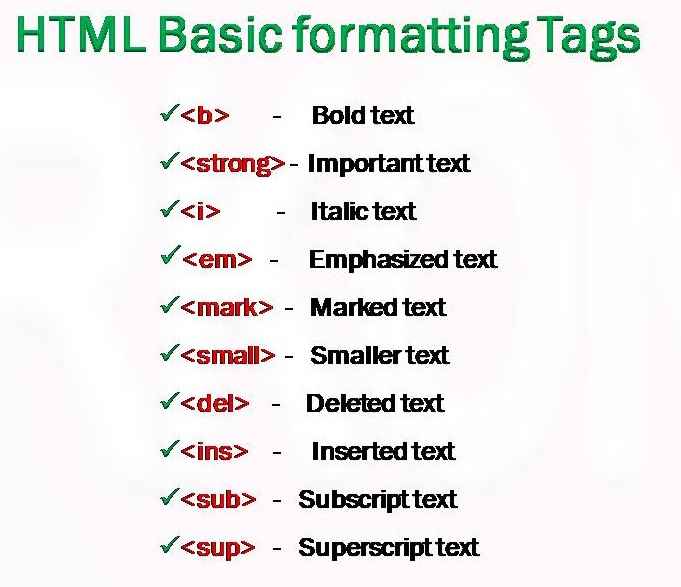
The <em> element represents stress emphasis of its contents, while the <i> element represents text that is set off from the normal prose, such as a foreign word,

Mark:

Do not forget to buy milk today.

My favorite color is  red.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<b>](https://www.w3schools.com/tags/tag_b.asp) | Defines bold text |
| [<em>](https://www.w3schools.com/tags/tag_em.asp) | Defines emphasized text |
| [<i>](https://www.w3schools.com/tags/tag_i.asp) | Defines a part of text in an alternate voice or mood |
| [<small>](https://www.w3schools.com/tags/tag_small.asp) | Defines smaller text |
| [<strong>](https://www.w3schools.com/tags/tag_strong.asp) | Defines important text |
| [<sub>](https://www.w3schools.com/tags/tag_sub.asp) | Defines subscripted text |
| [<sup>](https://www.w3schools.com/tags/tag_sup.asp) | Defines superscripted text |
| [<ins>](https://www.w3schools.com/tags/tag_ins.asp) | Defines inserted text |
| [<del>](https://www.w3schools.com/tags/tag_del.asp) | Defines deleted text |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |



**The HTML <blockquote> element defines a section that is quoted from another source.**

Aasdsadsadsa dsad sa sa d sadsa dsasad sadsasadsadsaa

**The HTML <q> tag defines a short quotation.**

**Browsers normally insert quotation marks around the quotation.**

This text is “qas dsadsa dsa” as like this.

The HTML <abbr> tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

<abbr title="World Health Organization">WHO</abbr>

The WHO was founded in 1948.

The text in the <address> element usually renders in *italic,* and browsers will always add a line break before and after the <address> element.

The text in the <cite> element usually renders in *italic*.

<bdo dir="rtl">This text will be written from right to left</bdo>

[Try it Yourself »](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_formatting_bdo)

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<abbr>](https://www.w3schools.com/tags/tag_abbr.asp) | Defines an abbreviation or acronym |
| [<address>](https://www.w3schools.com/tags/tag_address.asp) | Defines contact information for the author/owner of a document |
| [<bdo>](https://www.w3schools.com/tags/tag_bdo.asp) | Defines the text direction |
| [<blockquote>](https://www.w3schools.com/tags/tag_blockquote.asp) | Defines a section that is quoted from another source |
| [<cite>](https://www.w3schools.com/tags/tag_cite.asp) | Defines the title of a work |
| [<q>](https://www.w3schools.com/tags/tag_q.asp) | Defines a short inline quotation |

* \_self - Default. Opens the document in the same window/tab as it was clicked
* \_blank - Opens the document in a new window or tab
* \_parent - Opens the document in the parent frame
* \_top - Opens the document in the full body of the window

<a href="mailto:someone@example.com">Send email</a>

Use mailto: inside the href attribute to create a link that opens the user's email program (to let them send a new email):

<img src="workplace.jpg" alt="Workplace" usemap="#workmap">  
  
<map name="workmap">  
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">  
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">  
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">  
</map>

<picture>

<source media="(min-width: 650px)" srcset="img\_food.jpg">

<source media="(min-width: 465px)" srcset="img\_car.jpg">

<img src="img\_girl.jpg" style="width:auto;">

</picture>

<picture>  
  <source srcset="img\_avatar.png">  
  <source srcset="img\_girl.jpg">  
  <img src="img\_beatles.gif" alt="Beatles" style="width:auto;">  
</picture>

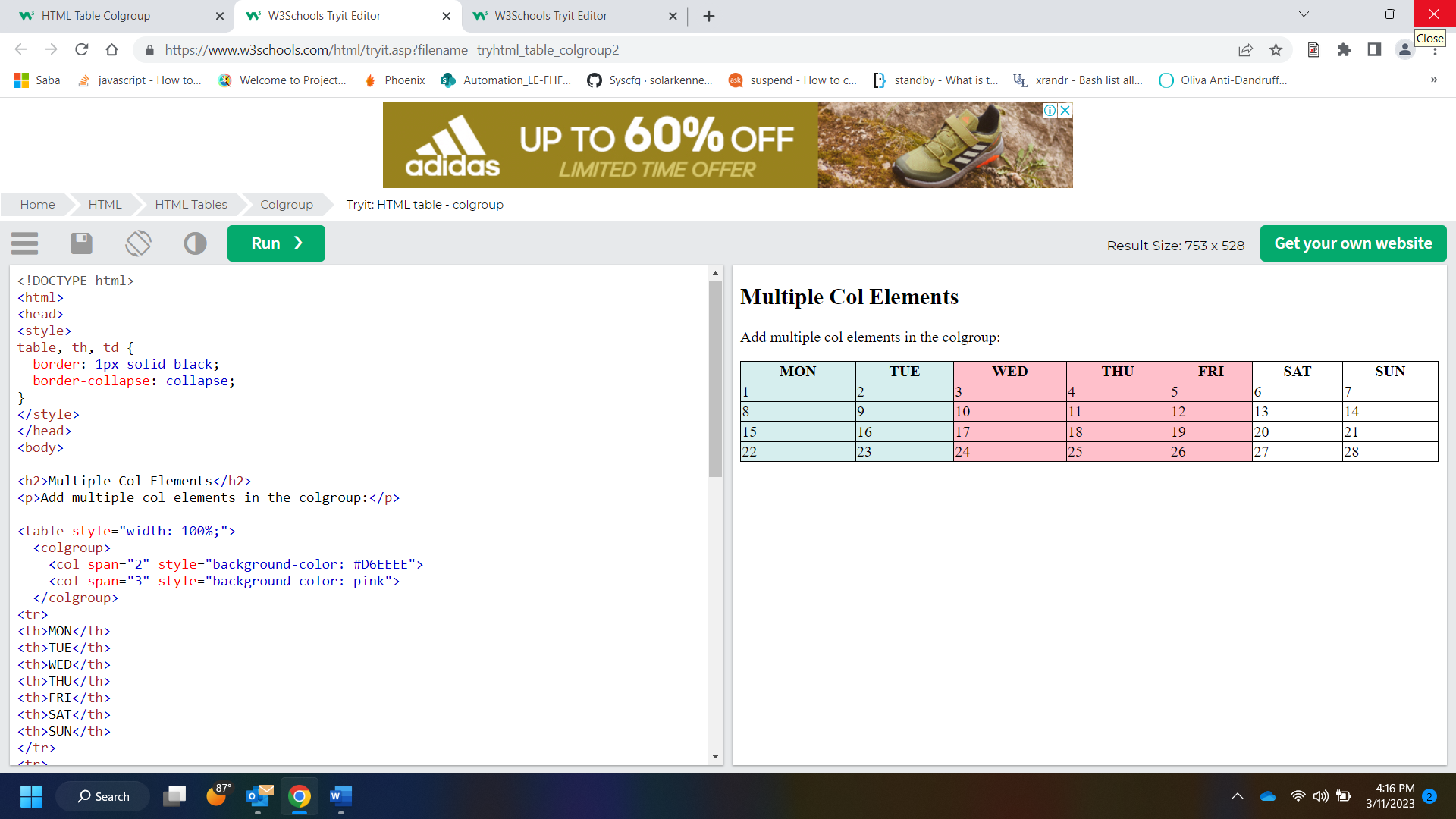
 <link rel="icon" type="image/x-icon" href="/images/favicon.ico">

Table:

* dotted
* dashed
* solid
* double
* groove
* ridge
* inset
* outset

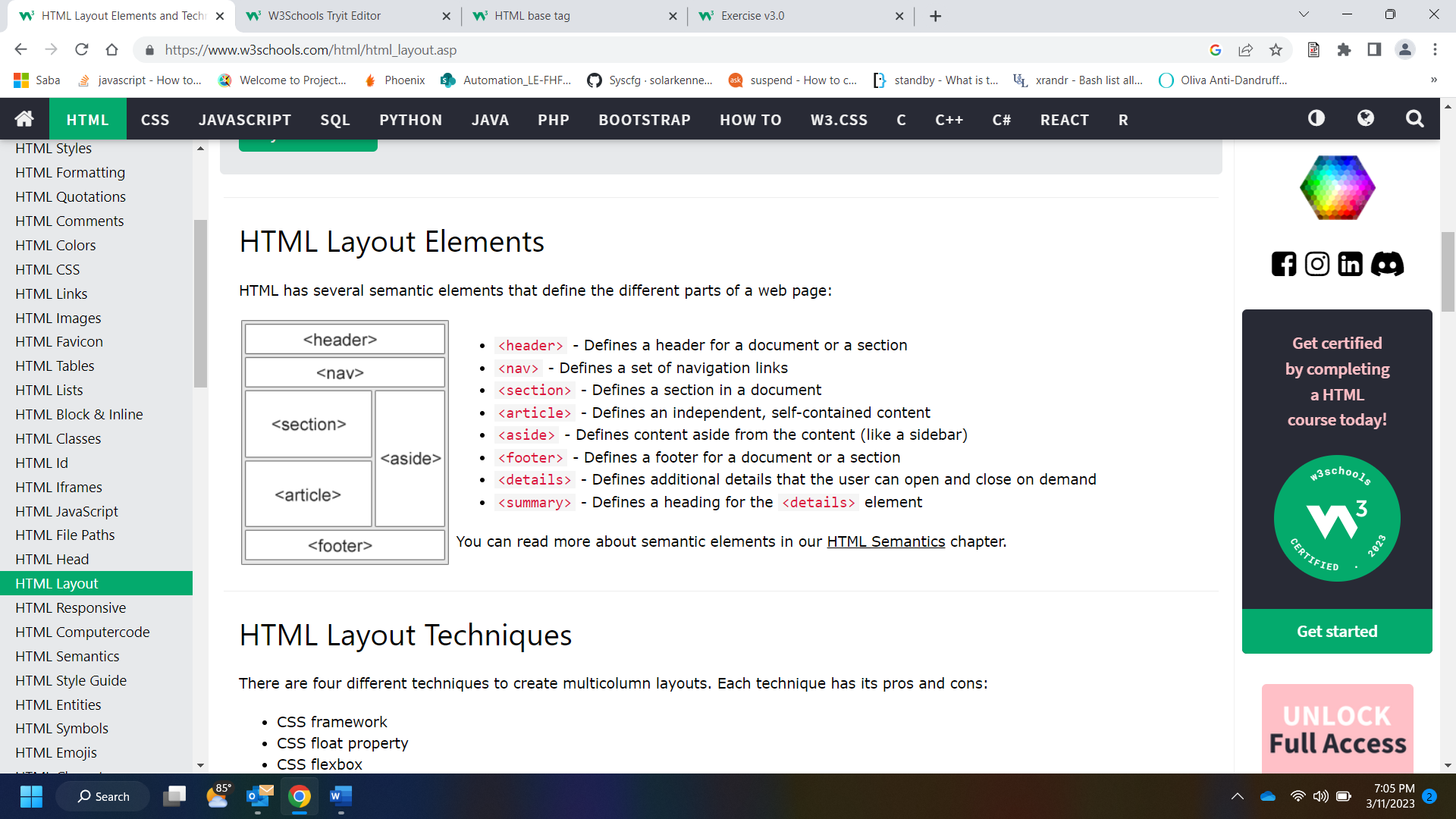
none

* hidden



Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.



<meta name=”viewport” content=”width=device-width, initial-scale=1.0”>

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<code>](https://www.w3schools.com/tags/tag_code.asp) | Defines programming code |
| [<kbd>](https://www.w3schools.com/tags/tag_kbd.asp) | Defines keyboard input |
| [<samp>](https://www.w3schools.com/tags/tag_samp.asp) | Defines computer output |
| [<var>](https://www.w3schools.com/tags/tag_var.asp) | Defines a variable |
| [<pre>](https://www.w3schools.com/tags/tag_pre.asp) | Defines preformatted text |

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

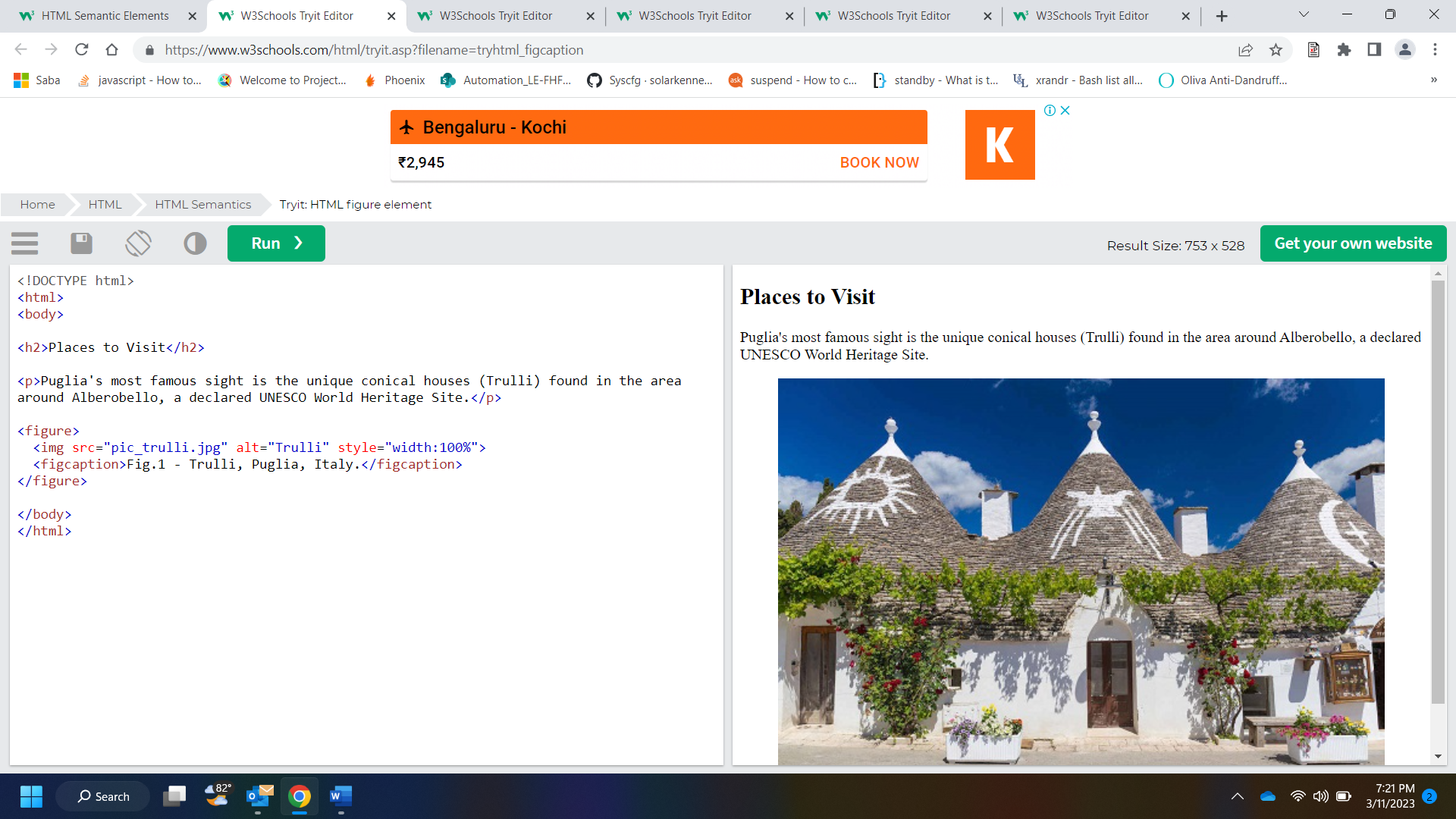
Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.

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

**Note:** You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.



According to the W3C: "A semantic Web allows data to be shared and reused across applications, enterprises, and communities."

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<article>](https://www.w3schools.com/tags/tag_article.asp) | Defines independent, self-contained content |
| [<aside>](https://www.w3schools.com/tags/tag_aside.asp) | Defines content aside from the page content |
| [<details>](https://www.w3schools.com/tags/tag_details.asp) | Defines additional details that the user can view or hide |
| [<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp) | Defines a caption for a <figure> element |
| [<figure>](https://www.w3schools.com/tags/tag_figure.asp) | Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. |
| [<footer>](https://www.w3schools.com/tags/tag_footer.asp) | Defines a footer for a document or section |
| [<header>](https://www.w3schools.com/tags/tag_header.asp) | Specifies a header for a document or section |
| [<main>](https://www.w3schools.com/tags/tag_main.asp) | Specifies the main content of a document |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |
| [<nav>](https://www.w3schools.com/tags/tag_nav.asp) | Defines navigation links |
| [<section>](https://www.w3schools.com/tags/tag_section.asp) | Defines a section in a document |
| [<summary>](https://www.w3schools.com/tags/tag_summary.asp) | Defines a visible heading for a <details> element |
| [<time>](https://www.w3schools.com/tags/tag_time.asp) | Defines a date/time |

## Always Declare Document Type

## Use Lowercase Element Names

## Close All HTML Elements

## Use Lowercase Attribute Names

## Always Quote Attribute Values

## Spaces and Equal Signs

## Always Specify alt, width, and height for Images

## Avoid Long Code Lines

## Blank Lines and Indentation

## Add the lang Attribute

Meta Data

To ensure proper interpretation and correct search engine indexing, both the language and the character encoding <meta charset="*charset*"> should be defined as early as possible in an HTML document:

<!DOCTYPE html>  
<html lang="en-us">  
<head>  
  <meta charset="UTF-8">  
  <title>Page Title</title>  
</head>

## Default Filenames

When a URL does not specify a filename at the end (like "https://www.w3schools.com/"), the server just adds a default filename, such as "index.html", "index.htm", "default.html", or "default.htm".

To display a less than sign (<) we must write: **&lt;** or **&#60;**

## Non-breaking Space

A commonly used entity in HTML is the non-breaking space: **&nbsp;**

A non-breaking space is a space that will not break into a new line.

Two words separated by a non-breaking space will stick together (not break into a new line). This is handy when breaking the words might be disruptive.

Examples:

* § 10
* 10 km/h
* 10 PM
* UTF-8 covers almost all of the characters and symbols in the world.
* The HTML charset Attribute
* To display an HTML page correctly, a web browser must know the character set used in the page.
* This is specified in the <meta> tag:
* <meta charset="UTF-8">

## From ASCII to UTF-8

ASCII was the first character encoding standard. ASCII defined 128 different characters that could be used on the internet: numbers (0-9), English letters (A-Z), and some special characters like ! $ + - ( ) @ < > .

ISO-8859-1 was the default character set for HTML 4. This character set supported 256 different character codes. HTML 4 also supported UTF-8.

ANSI (Windows-1252) was the original Windows character set. ANSI is identical to ISO-8859-1, except that ANSI has 32 extra characters.

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!

scheme://prefix.domain:port/path/filename

Explanation:

* **scheme** - defines the **type** of Internet service (most common is **http or https**)
* **prefix** - defines a domain **prefix** (default for http is **www**)
* **domain** - defines the Internet **domain name**(like w3schools.com)
* **port** - defines the **port number**at the host (default for http is **80**)
* **path** - defines a **path** at the server (If omitted: the root directory of the site)
* **filename** - defines the name of a document or resource

## The Most Important Differences from HTML for xhtml

* <!DOCTYPE> is **mandatory**
* The xmlns attribute in <html> is **mandatory**
* <html>, <head>, <title>, and <body> are **mandatory**
* Elements must always be **properly nested**
* Elements must always be **closed**
* Elements must always be in **lowercase**
* Attribute names must always be in **lowercase**
* Attribute values must always be **quoted**
* Attribute minimization is **forbidden**

<form action attribute>

**Tip:** If the action attribute is omitted, the action is set to the current page.

The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:

|  |  |
| --- | --- |
| **Value** | **Description** |
| \_blank | The response is displayed in a new window or tab |
| \_self | The response is displayed in the current window |
| \_parent | The response is displayed in the parent frame |
| \_top | The response is displayed in the full body of the window |
| *Framename* | The response is displayed in a named iframe |

The default value is \_self which means that the response will open

**Notes on GET:**

* Appends the form data to the URL, in name/value pairs
* NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
* The length of a URL is limited (2048 characters)
* Useful for form submissions where a user wants to bookmark the result
* GET is good for non-secure data, like query strings in Google

**Notes on POST:**

* Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
* POST has no size limitations, and can be used to send large amounts of data.
* Form submissions with POST cannot be bookmarked

## The Novalidate Attribute

The novalidate attribute is a boolean attribute.

<form action="/action\_page.php" novalidate>

## The HTML <form> Elements

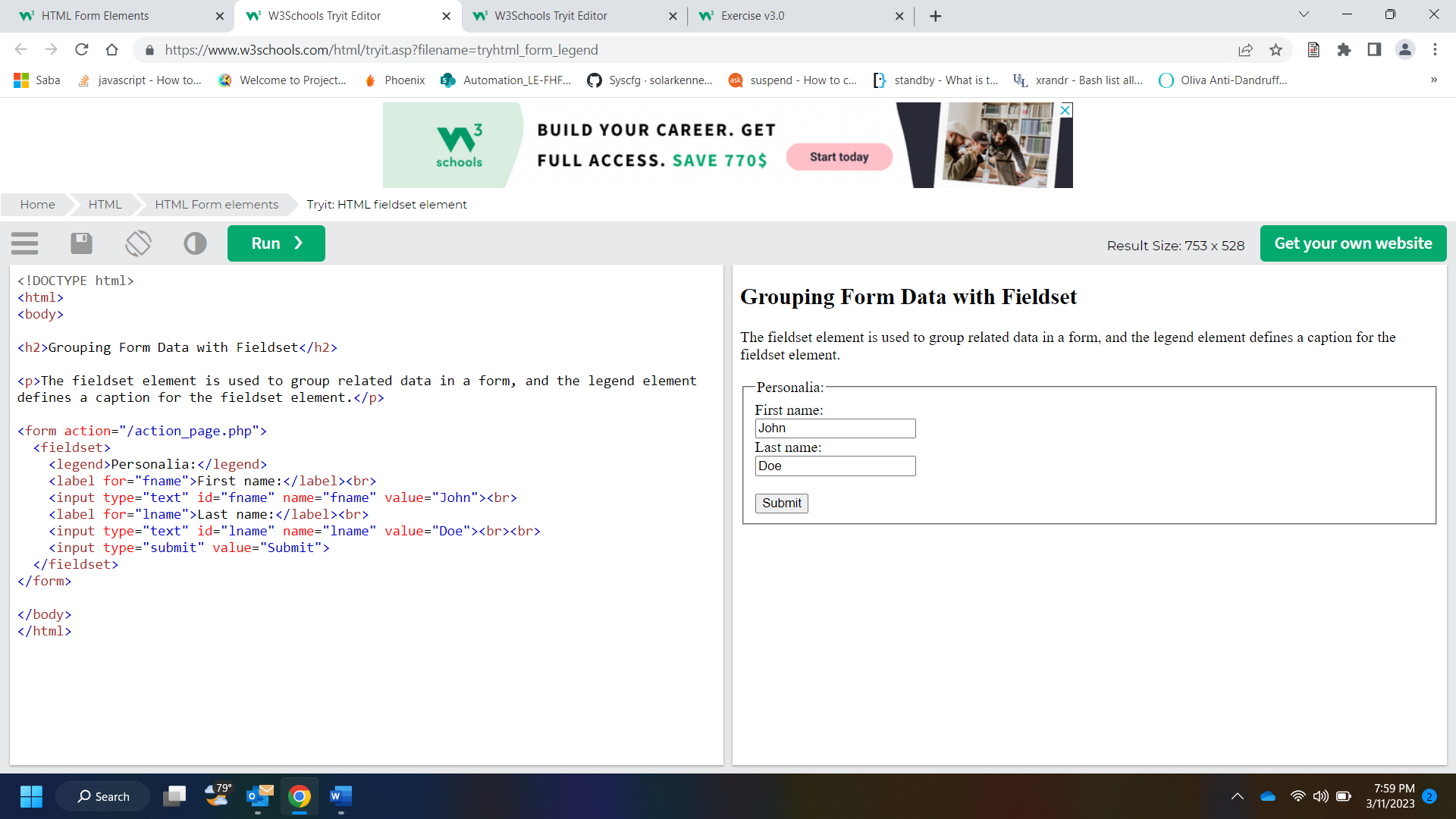
The HTML <form> element can contain one or more of the following form elements:

* <input>
* <label>
* <select>
* <textarea>
* <button>
* <fieldset>
* <legend>
* <datalist>
* <output>
* <option>
* <optgroup>

<select id="cars" name="cars" size="3">

<textarea name="message" rows="10" cols="30">

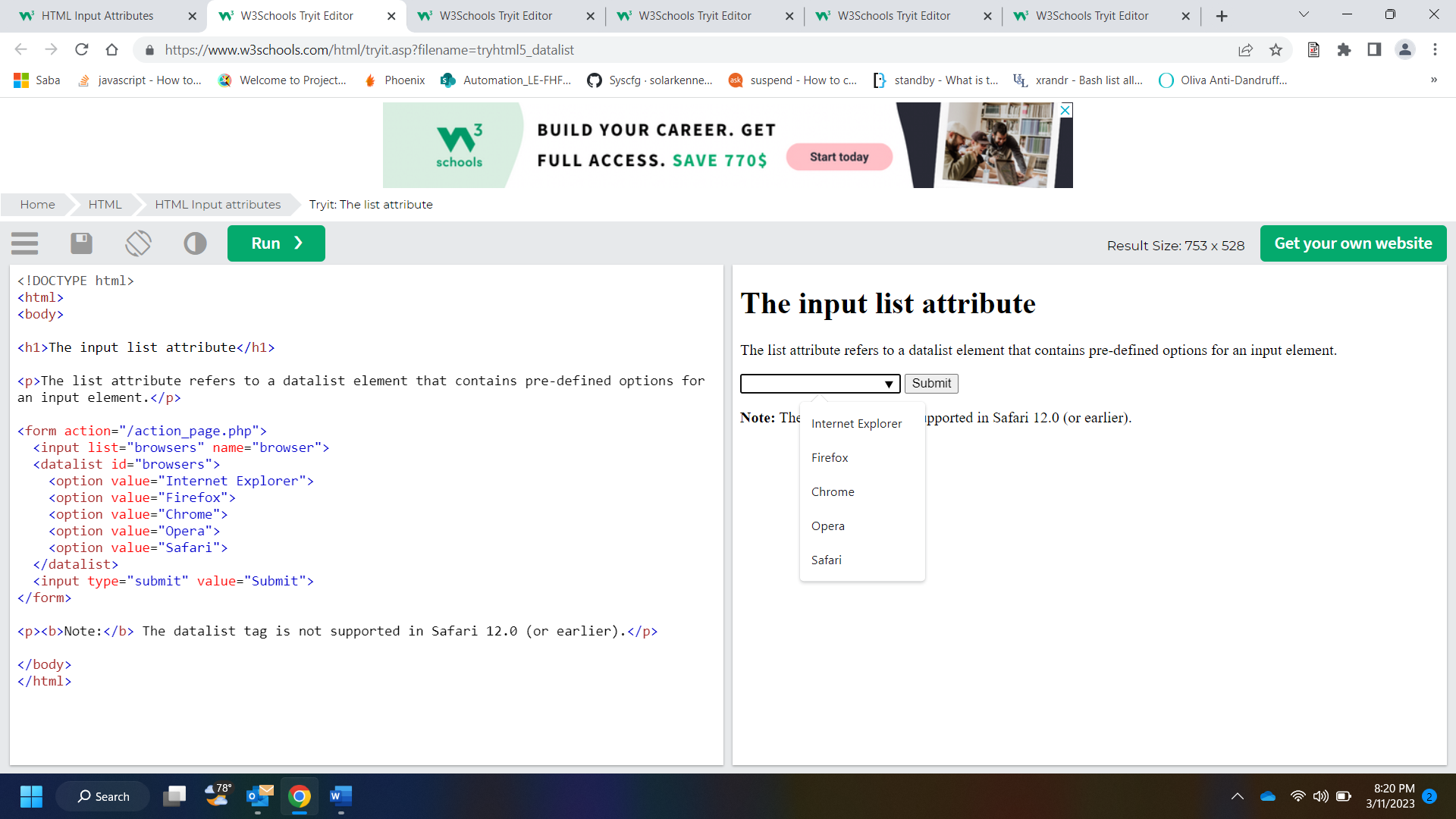
<textarea name="message" style="width:200px; height:600px;">



Here are the different input types you can use in HTML:

* <input type="button">
* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="hidden">
* <input type="image">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="reset">
* <input type="search">
* <input type="submit">
* <input type="tel">
* <input type="text">
* <input type="time">
* <input type="url">
* <input type="week">
* <input type="reset"> defines a **reset button** that will reset all form values to their default values:
* The <input type="image"> defines an image as a submit button.

  <input type="text" id="fname" name="fname" autofocus><br>



<form action="/action\_page.php">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formaction="/action\_page2.php" value="Submit as Admin">  
</form>

<button formenctype="multipart/form-data"

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.

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.moveTo(0, 0);  
ctx.lineTo(200, 100);  
ctx.stroke();  
</script>

SVG is a language for describing 2D graphics in XML.

Canvas draws 2D graphics, on the fly (with a JavaScript).

|  |  |
| --- | --- |
| **Canvas** | **SVG** |
| * Resolution dependent * No support for event handlers * Poor text rendering capabilities * You can save the resulting image as .png or .jpg * Well suited for graphic-intensive games | * Resolution independent * Support for event handlers * Best suited for applications with large rendering areas (Google Maps) * Slow rendering if complex (anything that uses the DOM a lot will be slow) * Not suited for game applications |

|  |  |  |
| --- | --- | --- |
| MPEG | .mpg .mpeg | MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML. |
| AVI | .avi | AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| WMV | .wmv | WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| QuickTime | .mov | QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. |
| RealVideo | .rm .ram | RealVideo. Developed by Real Media to allow video streaming with low bandwidths. Does not play in web browsers. |
| Flash | .swf .flv | Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers. |
| Ogg | .ogg | Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML. |
| WebM | .webm | WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML. |
| MPEG-4 or MP4 | .mp4 | MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and  recommended by YouTube. |

## HTML <video> Autoplay

To start a video automatically, use the autoplay attribute:

### **Example**

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

<video width="320" height="240" autoplay muted>

<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>

The <embed> element also defines an embedded object within an HTML document. – only in html5

<iframe width="420" height="345" src="https://www.youtube.com/embed/tgbNymZ7vqY">

</iframe>

<iframe width="420" height="315"  
src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1&mute=1">  
</iframe>

The getCurrentPosition() method is used to return the user's position.

    navigator.geolocation.getCurrentPosition(showPosition);

<!DOCTYPE HTML>

<html>

<head>

<style>

#div1 {

width: 350px;

height: 70px;

padding: 10px;

border: 1px solid #aaaaaa;

}

</style>

<script>

function allowDrop(ev) {

ev.preventDefault();

}

function drag(ev) {

ev.dataTransfer.setData("text", ev.target.id);

}

function drop(ev) {

ev.preventDefault();

var data = ev.dataTransfer.getData("text");

ev.target.appendChild(document.getElementById(data));

}

</script>

</head>

<body>

<p>Drag the W3Schools image into the rectangle:</p>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<br>

<img id="drag1" src="img\_logo.gif" draggable="true" ondragstart="drag(event)" width="336" height="69">

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<p>Count numbers: <output id="result"></output></p>

<button onclick="startWorker()">Start Worker</button>

<button onclick="stopWorker()">Stop Worker</button>

<p><strong>Note:</strong> Internet Explorer 9 and earlier versions do not support Web Workers.</p>

<script>

var w;

function startWorker() {

if(typeof(Worker) !== "undefined") {

if(typeof(w) == "undefined") {

w = new Worker("demo\_workers.js");

}

w.onmessage = function(event) {

document.getElementById("result").innerHTML = event.data;

};

} else {

document.getElementById("result").innerHTML = "Sorry, your browser does not support Web Workers...";

}

}

function stopWorker() {

w.terminate();

w = undefined;

}

</script>

</body>

</html>

Server-Sent Events (SSE) allow a web page to get updates from a server.

var source = new EventSource("demo\_sse.php");  
source.onmessage = function(event) {  
  document.getElementById("result").innerHTML += event.data + "<br>";  
};