<LESSON 01>

HTML

<Prepared By> Henry F. Bruckman Vargas

<Last Revised> March 11, 2019

Objectives

- Learn basic HTML terminology.
- Learn the basic structure of an HTML document.
- Learn some basic HTML elements and their most important attributes.
- Learn how and where to find more information about HTML.

What is HTML?

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage.
- Web pages are HTML documents.
- HTML documents are made of HTML elements.
- HTML elements are represented by HTML tags.

What is HTML?

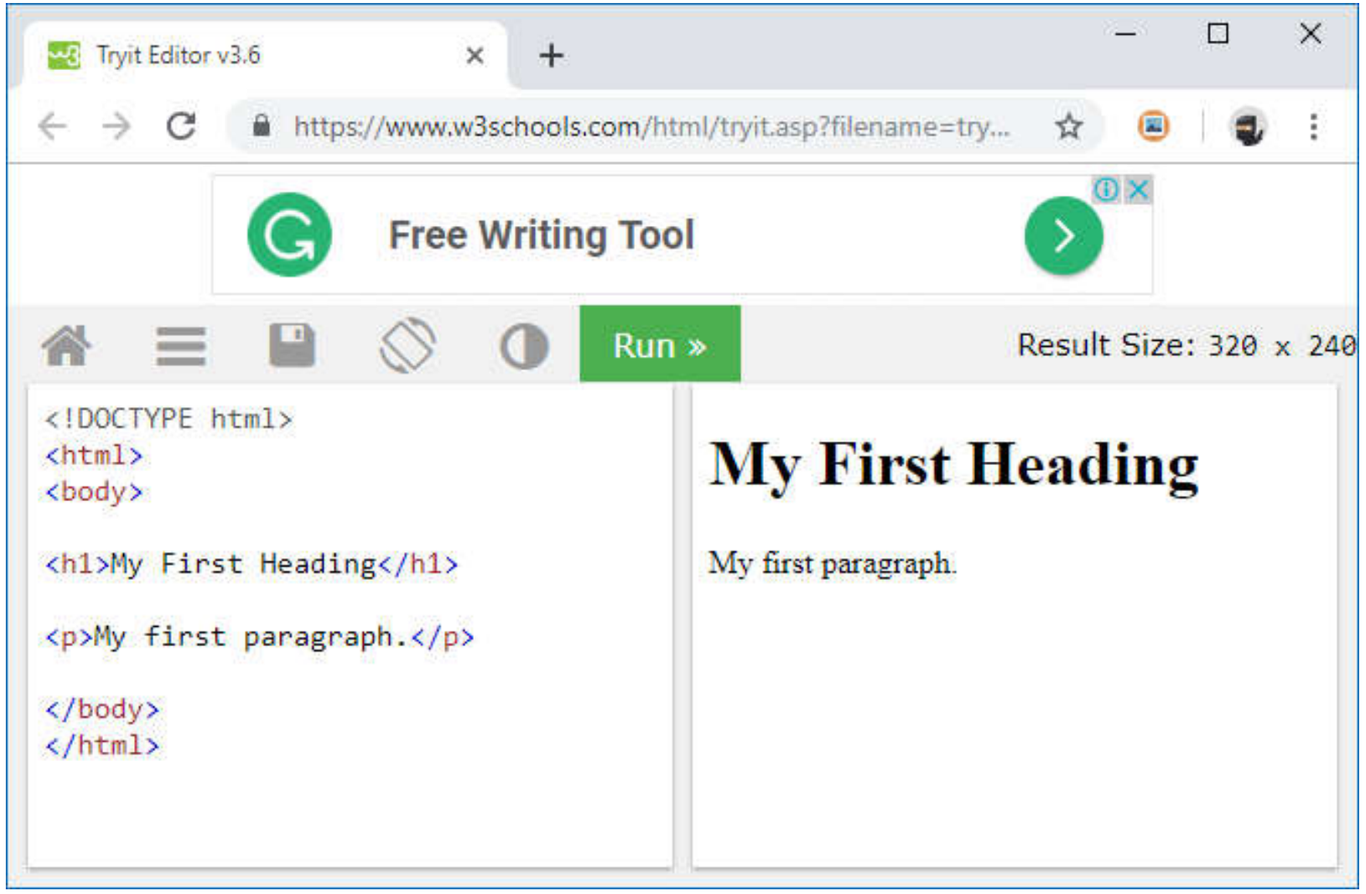
- HTML tags are names surrounded by angle brackets (`<>`).
- HTML tags normally come in pairs like `<p>` and `</p>`.
- The first tag in a pair is the start tag (or opening tag), the second tag is the end tag (or closing tag).

Web Browsers

- The purpose of a web browser (e.g., Chrome, Internet Explorer, Edge, Opera, Konqueror, Firefox, Safari) is to read HTML documents and display them.

- The browser does not display the HTML tags. Rather, it uses them to determine how to display the document.

Example


Tryit Editor v3.6
https://www.w3schools.com/html/tryit.asp?filename=try...
Free Writing Tool
Run »
Result Size: 320 x 240
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
My first paragraph.
</body>
</html>
My First Heading
My first paragraph.

HTML Editors

- Web pages can be created and modified by using professional HTML editors.
- However, for learning HTML we recommend a ~~simple~~ text editor ~~like Notepad (Windows OS) or TextEdit (Mac OS)~~ with syntax highlighting, auto-format, and auto-complete like Visual Studio Code or Netbeans.
- We believe using a ~~simple~~ text editor is a good way to learn HTML.

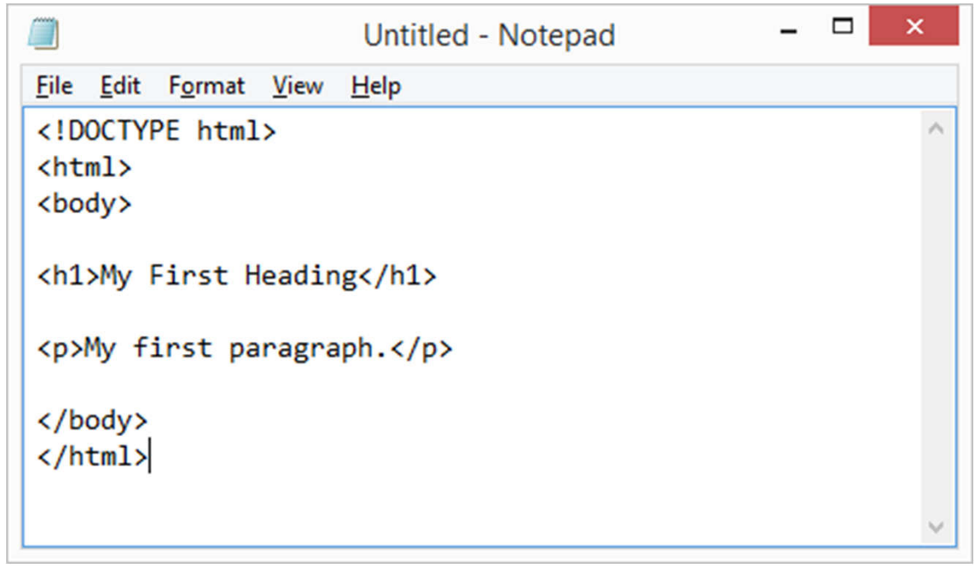
Write or Copy Some HTML

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

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```



Spaces Do Not Matter! But They Do!

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

- To the web browser, the number of spaces (or lines) between element tags do not matter!
- But for human readability purposes they do matter!

Spaces Do Not Matter! But They Do!

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

- So, you should ALWAYS indent (and format) your source code consistently.
- You can either use spaces or tabs to indent your code.

Spaces Do Not Matter! But They Do!

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

- Select the text and press:

TAB
to indent

and

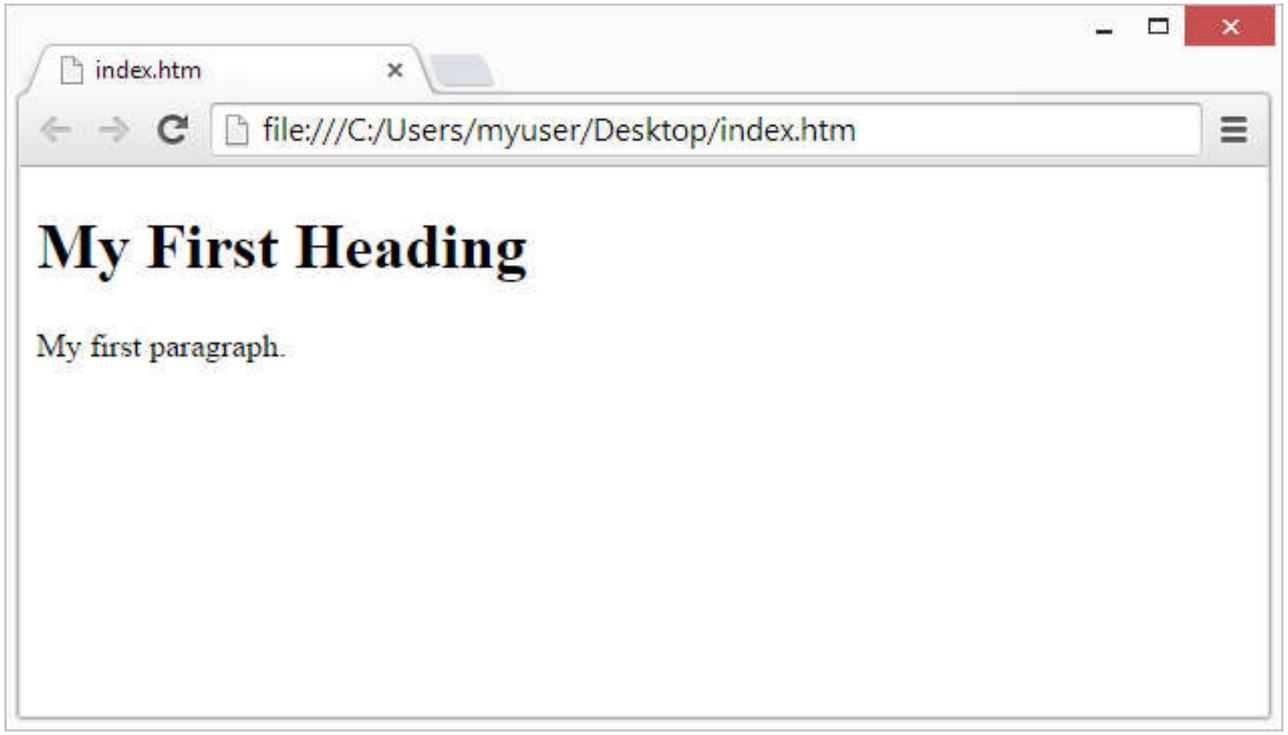
SHIFT + TAB
to unindent.

Save the HTML Page

- Write the file name in-between double quotes (").
- Include the *.htm or *.html extension to the file name.
- Use only letters ([a-z][A-Z]), digits ([0-9]) and underscores (_). No spaces!
- Set the encoding to UTF-8 (which is the preferred encoding for HTML files).
- Examples:

  "index.htm"
  "index.html"
  "support.html"
  "contact_us.html"
  "page42.html"
  "page_42.html"

Example


index.htm
file:///C:/Users/myuser/Desktop/index.htm
My First Heading
My first paragraph.

HTML Documents

- All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.

- The HTML document itself begins with `<html>` and ends with `</html>`.

- The visible part of the HTML document is between `<body>` and `</body>`.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

HTML Elements

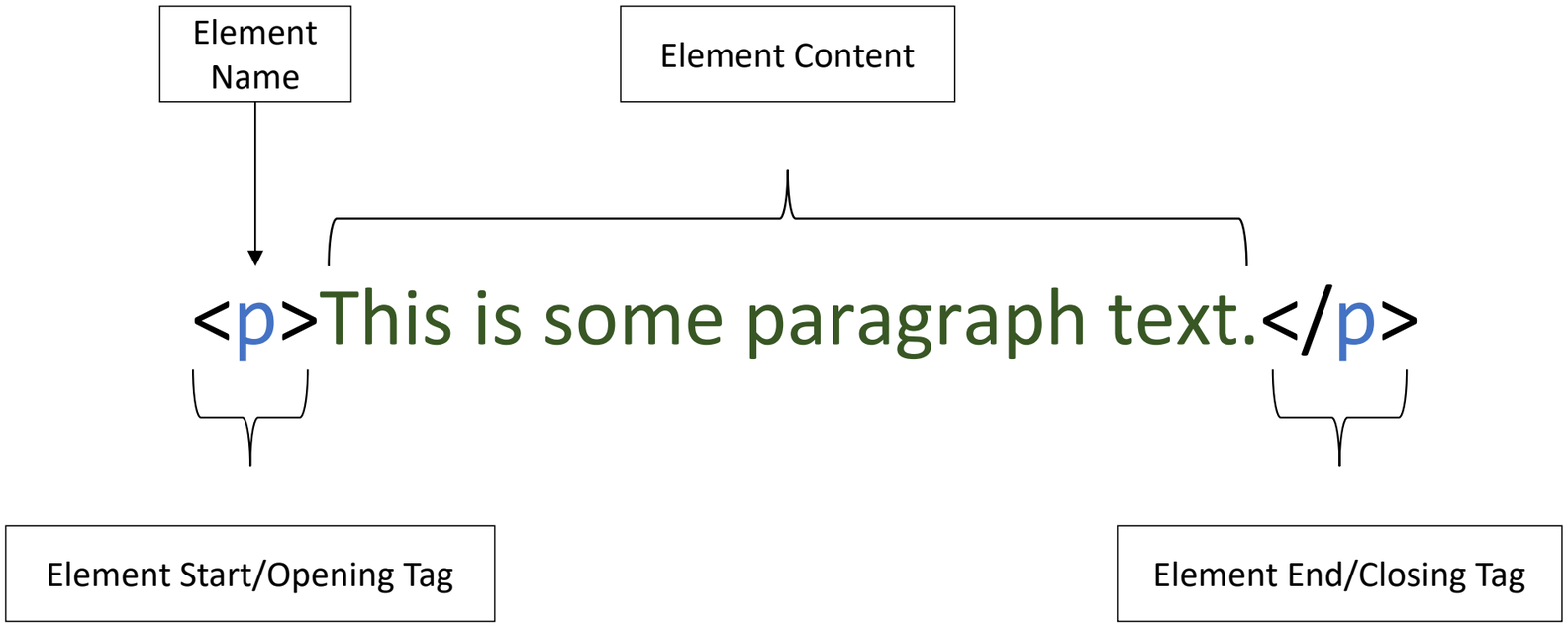
- An HTML element usually consists of a **start tag** and **end tag**, with the content inserted in between.

- The HTML element is everything from the start tag to the end tag.

- Example:

```
<p>My first paragraph.</p>
```

HTML Elements



HTML Elements

- Some elements do not have an end tag, or the end tag is optional (e.g., `<img>` or `<br>` or `<li>` or `<hr>` tags)

- Some HTML elements will display correctly, even if you forget the end tag.

- **<u>NOTE</u>: Never rely on this. It might produce unexpected results and/or errors if you forget the end tag.**

HTML Elements

- HTML elements with no content are called empty elements.
- Empty elements can be "closed" in the opening tag like this: `<br />`.
- **NOTE:** HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.

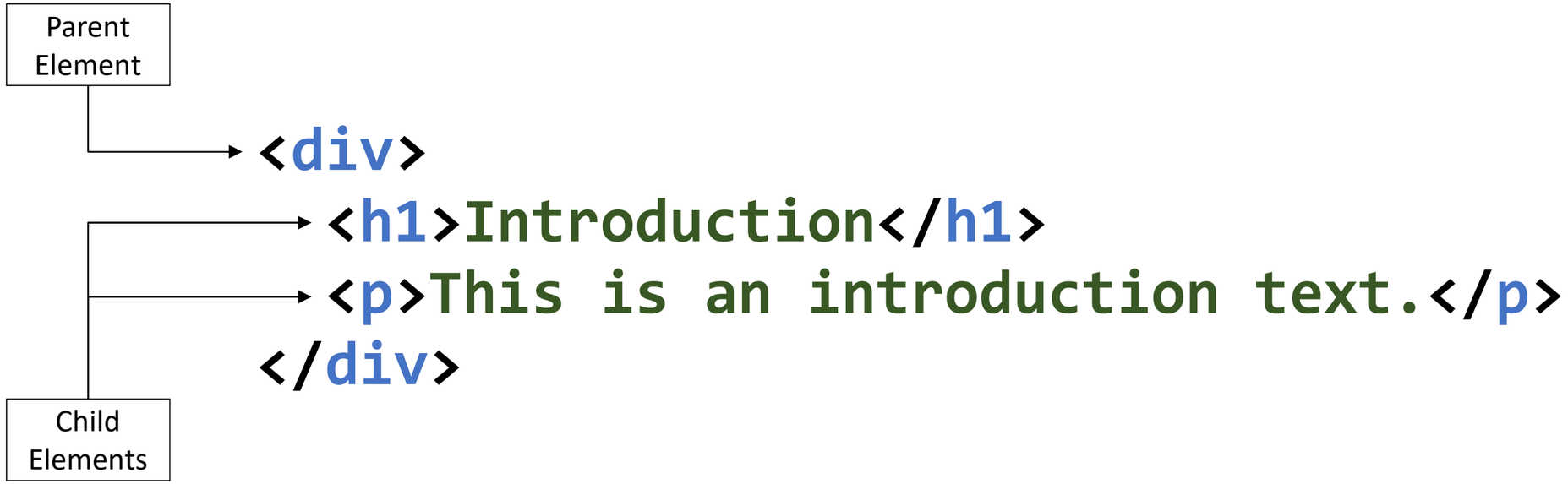
HTML Elements

- HTML elements can be nested, that is:
  - HTML elements can contain other HTML elements
  - HTML elements can be composed of other HTML elements.
- All HTML documents consist of nested HTML elements.
  - All HTML elements are nested inside the `<html>` element.

HTML Elements

- HTML elements contained within an HTML element are called *children*.
- HTML elements that contain other HTML elements are called *parents*.
- HTML elements that share the same parent element are called *siblings*.
- Child elements of child elements are called *descendants*.
- Parent elements of parent elements are called *ancestors*.
  - The `<html>` element is ancestor of all HTML elements in a HTML document.

HTML Elements



HTML Elements

- HTML tags are not case sensitive:

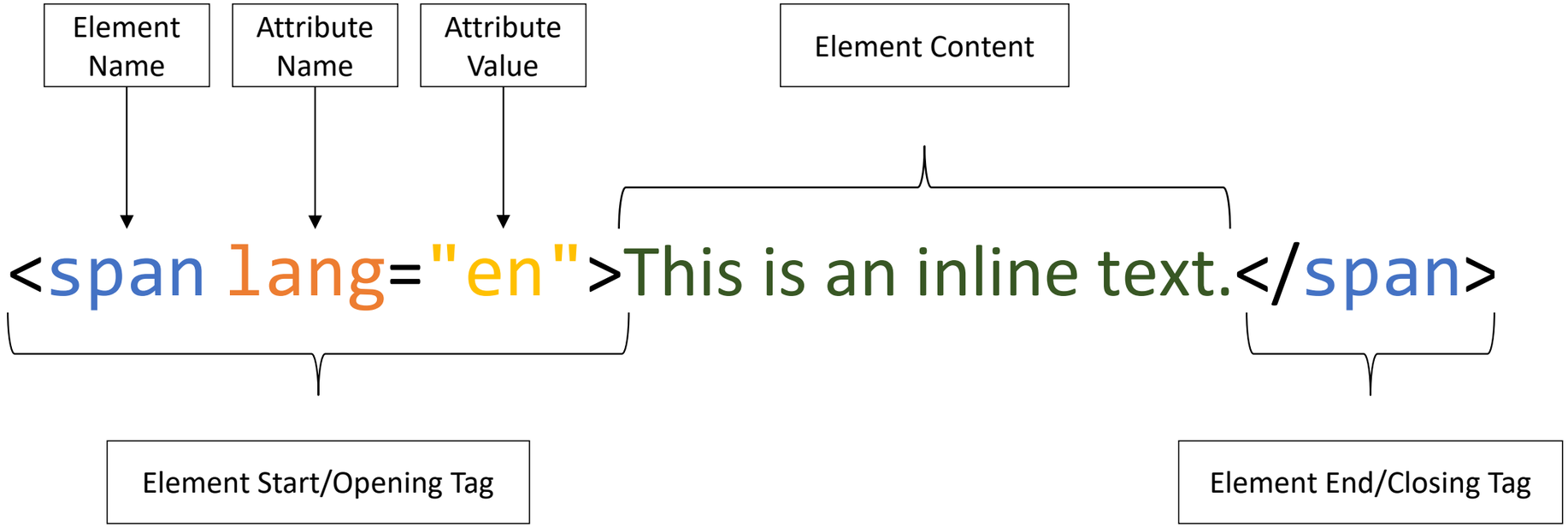
  `<P>` means the same as `<p>`

- The HTML5 standard does not require lowercase tags.
  - W3C **recommends** lowercase for HTML.
  - W3C **demands** lowercase for XHTML.

HTML Attributes

- All HTML elements can have **attributes**.
- Attributes provide **additional information** about an HTML element.
- Attributes are always specified in the **start tag.**
- Attributes usually come in name/value pairs (e.g., **name="value"**).

HTML Elements



HTML Attributes

- HTML attributes are not case sensitive:

  `HREF` means the same as `href`

- The HTML5 standard does not require lowercase attributes.
  - W3C **recommends** lowercase for HTML.
  - W3C **demands** lowercase for XHTML.

HTML Attributes

- Some HTML attributes are special and only work for certain HTML elements (e.g., `href`, `src`, `alt`, `colspan`, `type`)

- Some other HTML attributes work for all HTML elements:
  - `id` is used to assign a unique name to the element.
  - `class` is used to assign a classification to the element.
  - `style` is used to set the CSS format of the element inline.
  - `title` is used to show a tooltip text on over the element.

HTML Global Attributes

Attribute	Description
accesskey	Specifies a shortcut key to activate/focus an element
class	Specifies one or more class names for an element
contenteditable	Specifies whether the content of an element is editable or not
data-*	Used to store custom data private to the page or application
dir	Specifies the text direction for the content in an element
draggable	Specifies whether an element is draggable or not
dropzone	Specifies if the dragged data is going to copied, moved, or linked
hidden	Specifies that an element is not yet, or is no longer, relevant

HTML Global Attributes

Attribute	Description
id	Specifies a unique id for an element
lang	Specifies the language of the element's content (*for screen readers)
spellcheck	Specifies whether the element is to have its spelling checked or not
style	Specifies an inline CSS style for an element
tabindex	Specifies the tabbing order of an element
title	Specifies extra information about an element as a tooltip
translate	Specifies whether the content of an element should be translated or not

HTML Attributes Values

- The HTML5 standard does not require quotes around attribute values, **unless the values have spaces**.

```
<a href=https://www.w3schools.com>

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

<img src=cake.jpg alt="A delicious cake." width=320>

<img src="cake.jpg" alt="A delicious cake." width="320">
```

HTML Attributes Values

- Double quotes around attribute values are the most common in HTML, but **single quotes can also be used**.

```
<img src='cake.jpg' alt='A delicious cake.' width='320'>
```

- When the attribute value itself contains double quotes, it is necessary to use single quotes or vice versa:

```
<p title='John "ShotGun" Nelson'>
```

```
<p title="John 'ShotGun' Nelson">
```

HTML Attributes Values

- **NOTE**: Omitting quotes can produce errors.

- The HTML5 standard does not require quotes around attribute values.
  - W3C **recommends** quotes for HTML.
  - W3C **demands** quotes for XHTML.

The <!DOCTYPE> Declaration

- Represents the document type.
- Helps browsers to display web pages correctly.
- It must only appear once, at the top of the page (before any HTML tags).
- It is not case sensitive.

The <!DOCTYPE> Declaration

- **DOCTYPEs** are required for legacy reasons. When omitted, browsers tend to use a different rendering mode that is incompatible with some specifications.

- Including the **DOCTYPE** in a document ensures that the browser makes a best-effort attempt at following the relevant specifications.

- Example:

```
<!DOCTYPE html>
```

HTML Tag

- All HTML elements must be contained (nested) within the `<html>` and `</html>` tags.
- The `<html>` element represents the root of an HTML document.
- Authors are encouraged to specify a `lang` attribute on the root `<html>` element, giving the document's language (e.g., `"en"` for English, `"es"` for Spanish). This aids speech synthesis tools to determine what pronunciations to use, translation tools to determine what rules to use, and so forth.
- May contain a `<head>` element followed by a `<body>` element.

ISO 639-1 Language Codes

Language	ISO Code	Language	ISO Code	Language	ISO Code	Language	ISO Code
Abkhazian	ab	Basque	eu	Chinese (Simplified)	zh-Hans	Estonian	et
Afar	aa	Belarusian	be	Chinese (Traditional)	zh-Hant	Ewe	ee
Afrikaans	af	Bengali (Bangla)	bn	Chuvash	cv	Faroese	fo
Akan	ak	Bihari	bh	Cornish	kw	Fijian	fj
Albanian	sq	Bislama	bi	Corsican	co	Finnish	fi
Amharic	am	Bosnian	bs	Cree	cr	French	fr
Arabic	ar	Breton	br	Croatian	hr	Fula	ff
Aragonese	an	Bulgarian	bg	Czech	cs	Fulah	ff
Armenian	hy	Burmese	my	Danish	da	Pulaar	ff
Assamese	as	Catalan	ca	Divehi	dv	Pular	ff
Avaric	av	Chamorro	ch	Dhivehi	dv	Galician	gl
Avestan	ae	Chechen	ce	Maldivian	dv	Gaelic (Scottish)	gd
Aymara	ay	Chichewa	ny	Dutch	nl	Gaelic (Manx)	gv
Azerbaijani	az	Chewa	ny	Dzongkha	dz	Georgian	ka
Bambara	bm	Nyanja	ny	**English**	**en**	German	de
Bashkir	ba	Chinese	zh	Esperanto	eo	Greek	el

ISO 639-1 Language Codes

Language	ISO Code	Language	ISO Code	Language	ISO Code	Language	ISO Code
Greenlandic	kl	Inuktitut	iu	Kyrgyz	ky	Macedonian	mk
Guarani	gn	Inupiak	ik	Komi	kv	Malagasy	mg
Gujarati	gu	Irish	ga	Kongo	kg	Malay	ms
Haitian Creole	ht	Italian	it	Korean	ko	Malayalam	ml
Hausa	ha	Japanese	ja	Kurdish	ku	Maltese	mt
Hebrew	he	Javanese	jv	Kwanyama	kj	Maori	mi
Herero	hz	Kalaallisut	kl	Lao	lo	Marathi	mr
Hindi	hi	Greenlandic	kl	Latin	la	Marshallese	mh
Hiri Motu	ho	Kannada	kn	Latvian (Lettish)	lv	Moldavian	mo
Hungarian	hu	Kanuri	kr	Limburgish (Limburger)	li	Mongolian	mn
Icelandic	is	Kashmiri	ks	Lingala	ln	Nauru	na
Ido	io	Kazakh	kk	Lithuanian	lt	Navajo	nv
Igbo	ig	Khmer	km	Luga-Katanga	lu	Ndonga	ng
Indonesian	id, in	Kikuyu	ki	Luganda, Ganda	lg	Northern Ndebele	nd
Interlingua	ia	Kinyarwanda (Rwanda)	rw	Luxembourgish	lb	Nepali	ne
Interlingue	ie	Kirundi	rn	Manx	gv	Norwegian	no

ISO 639-1 Language Codes

Language	ISO Code	Language	ISO Code	Language	ISO Code	Language	ISO Code
Norwegian bokmål	nb	Quechua	qu	Siswati	ss	Thai	th
Norwegian nynorsk	nn	Romansh	rm	Slovak	sk	Tibetan	bo
Nuosu	ii	Romanian	ro	Slovenian	sl	Tigrinya	ti
Occitan	oc	Russian	ru	Somali	so	Tonga	to
Ojibwe	oj	Sami	se	Southern Ndebele	nr	Tsonga	ts
Old Church Slavonic	cu	Samoan	sm	**Spanish**	**es**	Turkish	tr
Old Bulgarian	cu	Sango	sg	Sundanese	su	Turkmen	tk
Oriya	or	Sanskrit	sa	Swahili (Kiswahili)	sw	Twi	tw
Oromo (Afaan Oromo)	om	Serbian	sr	Swati	ss	Uyghur	ug
Ossetian	os	Serbo-Croatian	sh	Swedish	sv	Ukrainian	uk
Pāli	pi	Sesotho	st	Tagalog	tl	Urdu	ur
Pashto, Pushto	ps	Setswana	tn	Tahitian	ty	Uzbek	uz
Persian (Farsi)	fa	Shona	sn	Tajik	tg	Venda	ve
Polish	pl	Sichuan Yi	ii	Tamil	ta	Vietnamese	vi
Portuguese	pt	Sindhi	sd	Tatar	tt	Volapük	vo
Punjabi (Eastern)	pa	Sinhalese	si	Telugu	te	Wallon	wa

ISO 639-1 Language Codes

Language	ISO Code
Welsh	cy
Wolof	wo
Western Frisian	fy
Xhosa	xh
Yiddish	yi, ji
Yoruba	yo
Zhuang	za
Chuang	za
Zulu	zu

Example

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Swapping Songs</title>
  </head>
  <body>
    <h1>Swapping Songs</h1>
    <p>Tonight I swapped some of the songs I wrote with some friends, who
    gave me some of the songs they wrote. I love sharing my music.</p>
  </body>
</html>
```

HTML Head

- Contained in: **`<html>`**
- May contain: *Metadata Content* (except **`<template>`**)

Metadata Content

- **Metadata content** is content that sets up the presentation or behavior of the rest of the content, or that sets up the relationship of the document with other documents, or that conveys other "out of band" information.

- Elements from other namespaces whose semantics are primarily metadata-related (e.g., RDF) are also metadata content.

  `<base>`, **`<link>`**, **`<meta>`**, **`<noscript>`**, **`<script>`**, **`<style>`**, **`<template>`**, **`<title>`**

HTML Body

- Contained in: `<html>`
- Contains: *Flow Content*

Flow Content

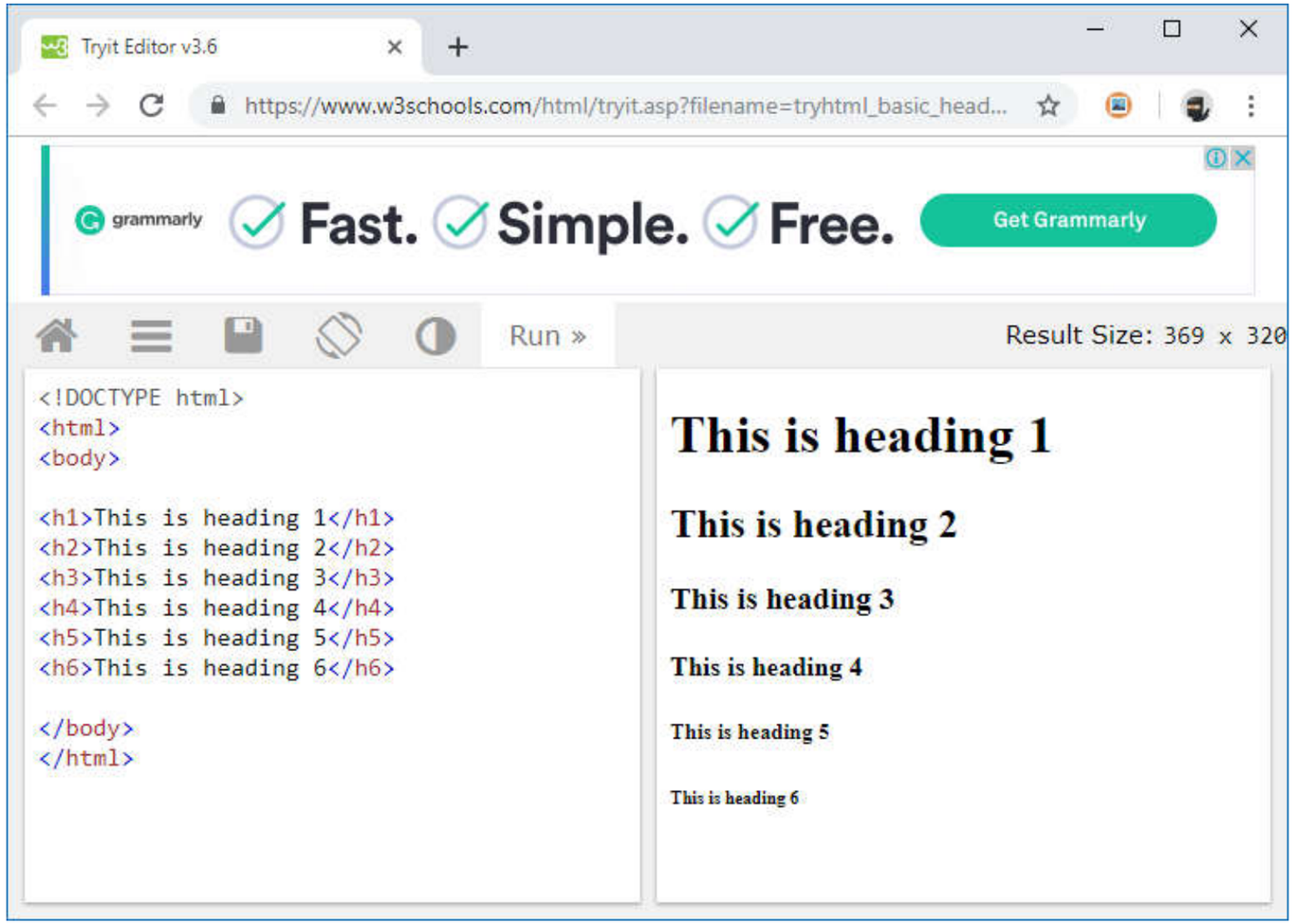
- <a>, <abbr>, <address>, <area> (if it is a descendant of a <map> element), <article>, <aside>, <audio>, <b>, <bdi>, <bdo>, <blockquote>,
, <button>, <canvas>, <cite>, <code>, <data>, <datalist>, <del>, <details>, <dfn>, <dialog>, , <dl>, , <embed>, <fieldset>, <figure>, <footer>, <form>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <header>, <hr>, <i>, <iframe>, <img>, <input>, <ins>, <kbd>, <label>, <link> (if it is allowed in the <body>), <main>, <map>, <mark>, <math>, <meter>, <nav>, <noscript>, <object>, <ol>, <output>, <p>, <picture>, <pre>, <progress>, <q>, <ruby>, <s>, <samp>, <script>, <section>, <select>, <small>, , , <style>, <sub>, <sup>, <svg>, <table>, <template>, <textarea>, <time>, <u>, <ul>, <var>, <video>, <wbr>, text

HTML Headings

- HTML headings are defined with the **<h1>** to **<h6>** tags.
- **<h1>** defines the most important heading.
- **<h6>** defines the least important heading.
- Examples:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Example


Tryit Editor v3.6
https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic_head...
grammarly
Fast.
Simple.
Free.
Get Grammarly
Run »
Result Size: 369 x 320
<!DOCTYPE html>
<html>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>
</html>
This is heading 1
This is heading 2
This is heading 3
This is heading 4
This is heading 5
This is heading 6

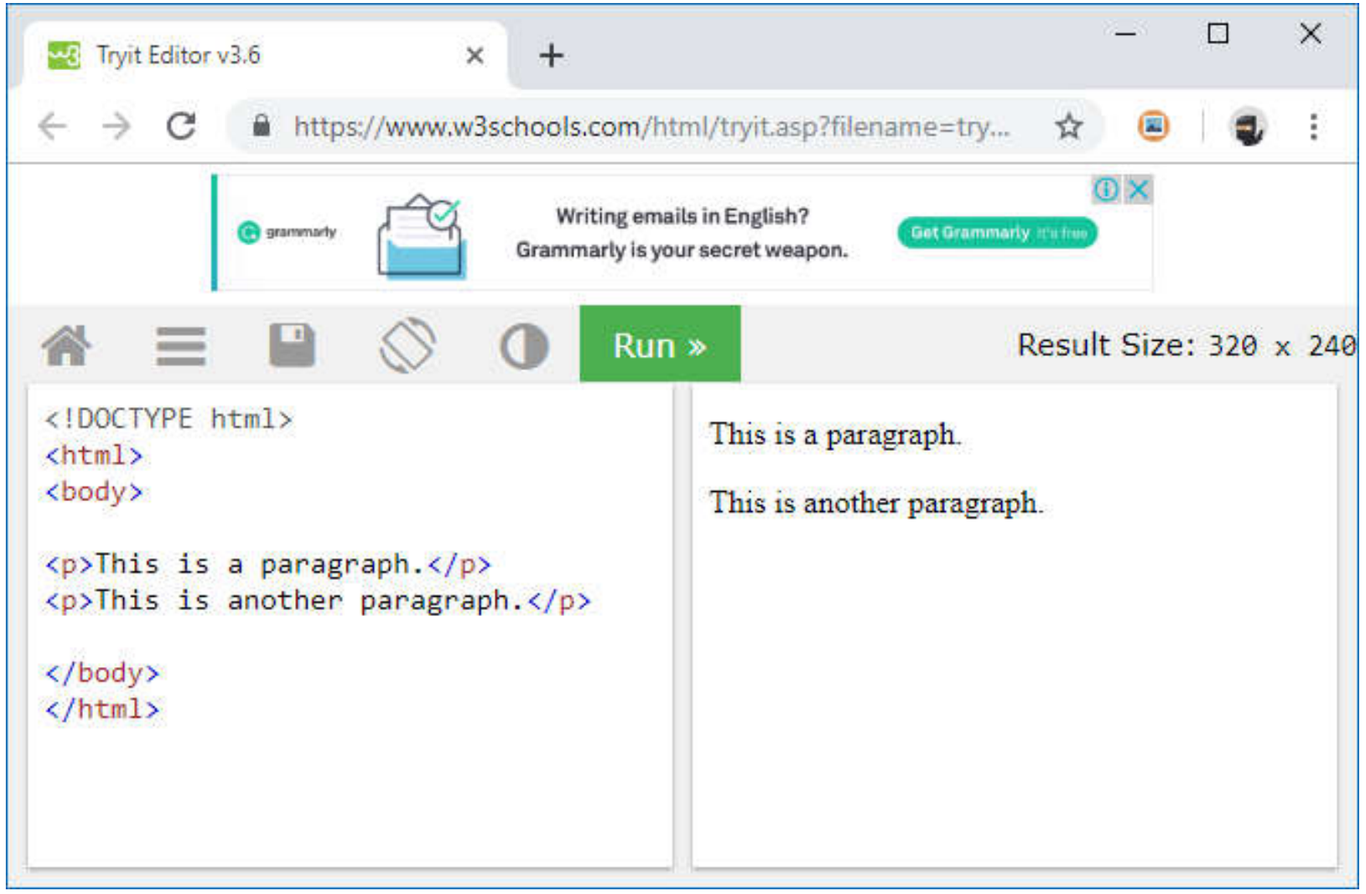
HTML Paragraphs

- HTML paragraphs are defined with the **<p>** tag.

- Examples:

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Example

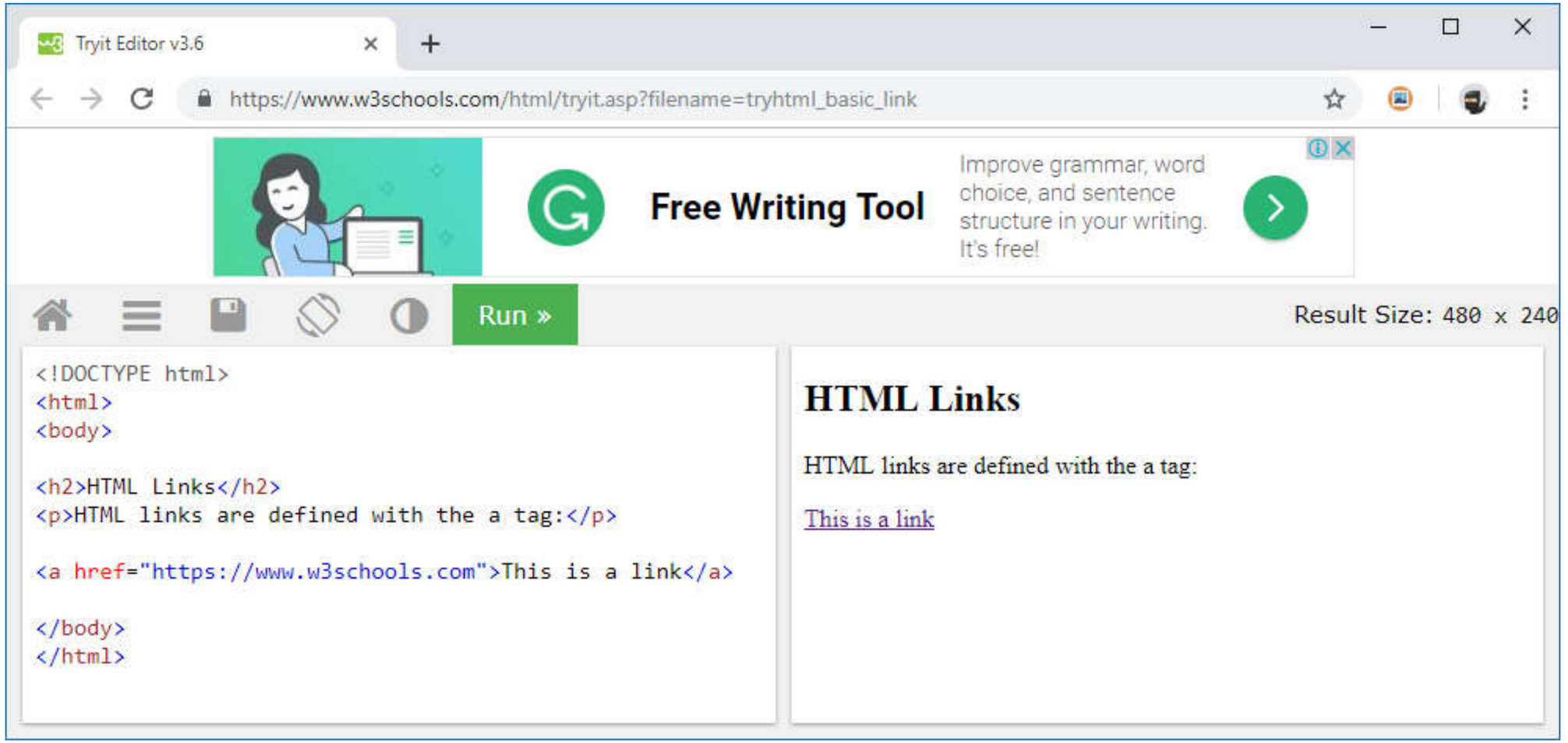

Tryit Editor v3.6
https://www.w3schools.com/html/tryit.asp?filename=try...
Writing emails in English?
Grammarly is your secret weapon.
Get Grammarly
Run »
Result Size: 320 x 240
<!DOCTYPE html>
<html>
<body>
This is a paragraph.
This is another paragraph.
</body>
</html>
This is a paragraph.
This is another paragraph.

HTML Links

- HTML links are defined with the `<a>` tag.
- The **link's destination** is specified in the `href` attribute.
- Examples:

```
<a href="https://www.w3schools.com">This is a link</a>
```

Example


Tryit Editor v3.6
https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic_link
Free Writing Tool
Improve grammar, word choice, and sentence structure in your writing. It's free!
Run »
Result Size: 480 x 240
<!DOCTYPE html>
<html>
<body>
<h2>HTML Links</h2>
HTML links are defined with the a tag:
This is a link
</body>
</html>
HTML Links
HTML links are defined with the a tag:
This is a link

HTML Links

- The value of `href` attribute can be the URL of a website, a webpage, or a file within a website:

- Examples:
  - `https://awebsite.com`
  - `https://awebsite.com/about.html`
  - `https://awebsite.com/files/bundle.zip`

- The value of `href` can also be an *absolute* or *relative* file path.

HTML Links

- To use an **absolute file path,** you need to include the URL of the website and the file path. Absolute file paths are typically used to link to external websites or files in other websites.

- Examples:

```
<a href="https://exsite.com">ExSite.Com</a>

<a href="https://exsite.com/info.html">ExSite Info</a>

<a href="https://exsite.com/pics/logo.jpg">ExSite</a>
```

HTML Links

- To use a **relative file path,** you need to include only the file path. Relative file paths are typically used to link to pages and files within the same website.

- Examples:

```
<a href="about.html">About</a>

<a href="pics/photo.jpg">Photo</a>

<a href="sounds/sfx/beep.wav">Beep</a>
```

HTML Links

- Relative file paths may include the following short hands:
  - `./` = current directory (redundant)
  - `../` = parent directory

- Examples:

```
<a href="./about.html">About</a>

<a href="../pics/photo.jpg">Photo</a>

<a href="../../sounds/sfx/beep.wav">Beep</a>
```

HTML Links

- To understand how the short hands work follow the examples in the next slides.

- Imagine that the file structure of your website is as shown in the right panel.

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - **products.html**
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - classic.mp3
  - **index.html**

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **products.html** from **index.html** you may write in **href** any of the following values:

  - Using an absolute file path:

    http://mysite.com/pages/products.html

  - Using a relative file path:

    pages/products.html

    ./pages/products.html

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - **products.html**
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - classic.mp3
  - **index.html**

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **`logo24.png`** from **`index.html`** you may write in `href` any of the following values:

  - Using an absolute file path:

    `http://mysite.com/images/icons/logo24.png`

  - Using a relative file path:

    `images/icons/logo24.png`

    `./images/icons/logo24.png`

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - **logo24.png**
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - products.html
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - classic.mp3
  - **index.html**

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **`classic.mp3`** from **`index.html`** you may write in `href` any of the following values:

  - Using an absolute file path:

    `http://mysite.com/sounds/music/classic.mp3`

  - Using a relative file path:

    `sounds/music/classic.mp3`

    `./sounds/music/classic.mp3`

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - products.html
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - **classic.mp3**
  - **index.html**

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **index.html** from **about.html** you may write in **href** any of the following values:

  - Using an absolute file path:

    http://mysite.com/index.html

  - Using a relative file path:

    ../index.html

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - **about.html**
    - products.html
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - classic.mp3
  - **index.html**

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **products.html** from **about.html** you may write in **href** any of the following values:

  - Using an absolute file path:

    http://mysite.com/pages/products.html

  - Using a relative file path:

    products.html

    ./products.html

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - **about.html**
    - **products.html**
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - classic.mp3
  - index.html

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **logo24.png** from **about.html** you may write in **href** any of the following values:

  - Using an absolute file path:

    http://mysite.com/images/icons/logo24.png

  - Using a relative file path:

    ../images/icons/logo24.png

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - **logo24.png**
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - **about.html**
    - products.html
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - classic.mp3
  - index.html

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **`classic.mp3`** from **`about.html`** you may write in `href` any of the following values:

  - Using an absolute file path:

    `http://mysite.com/sounds/music/classic.mp3`

  - Using a relative file path:

    `../sounds/music/classic.mp3`

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - **about.html**
    - products.html
  - sounds/
    - music/
      - bgm.html
      - rock.mp3
      - **classic.mp3**
  - index.html

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **index.html** from **bgm.html** you may write in **href** any of the following values:

  - Using an absolute file path:

    http://mysite.com/index.html

  - Using a relative file path:

    ../../index.html

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - products.html
  - sounds/
    - music/
      - **bgm.html**
      - rock.mp3
      - classic.mp3
  - **index.html**

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **products.html** from **bgm.html** you may write in **href** any of the following values:

  - Using an absolute file path:

    http://mysite.com/pages/products.html

  - Using a relative file path:

    ../../pages/products.html

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - **products.html**
  - sounds/
    - music/
      - **bgm.html**
      - rock.mp3
      - classic.mp3
  - index.html

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **logo24.png** from **bgm.html** you may write in **href** any of the following values:

  - Using an absolute file path:

    http://mysite.com/images/icons/logo24.png

  - Using a relative file path:

    ../../images/icons/logo24.png

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - **logo24.png**
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - products.html
  - sounds/
    - music/
      - **bgm.html**
      - rock.mp3
      - classic.mp3
  - index.html

HTML Links

- The way the short hands work is as follows. Imagine that the file structure of your website is as shown in the right panel.

- To access **`classic.mp3`** from **`bgm.html`** you may write in `href` any of the following values:

  - Using an absolute file path:

    `http://mysite.com/sounds/music/classic.mp3`

  - Using a relative file path:

    `classic.mp3`

    `./classic.mp3`

- mysite.com/
  - images/
    - icons/
      - favicon.ico
      - logo24.png
      - logo48.png
    - photos/
      - profilepic01.jpg
      - profilepic02.jpg
  - pages/
    - about.html
    - products.html
  - sounds/
    - music/
      - **bgm.html**
      - rock.mp3
      - **classic.mp3**
  - index.html

HTML Links

- You may use the **target** attribute to specify where do you want the link to open:
  - **"_blank"**: in a new window or tab
  - **"_self"**: in the place it was clicked (this is default)
  - **"_parent"**: in the parent frame
  - **"_top"**: as the new content of the current window or tab
  - **"fname"**: in the frame or internal frame of name fname

- Examples:

```
<a href="support.html" target="_top">Click here.</a>
```

HTML Images

- HTML images are defined with the `<img>` tag.
- The **source file** is specified by the `src` attribute.
  - The `src` attribute behaves in the same way as `href`. That is, you can use an absolute or relative path to the image file, it may be an internal file (within the current website) or an external file (from another website).
- The **alternative text** for when the image cannot be loaded and for accessibility is specified by the `alt` attribute.

HTML Images

- The **preferred dimensions** for the image are specified by the `width` and `height` attributes.

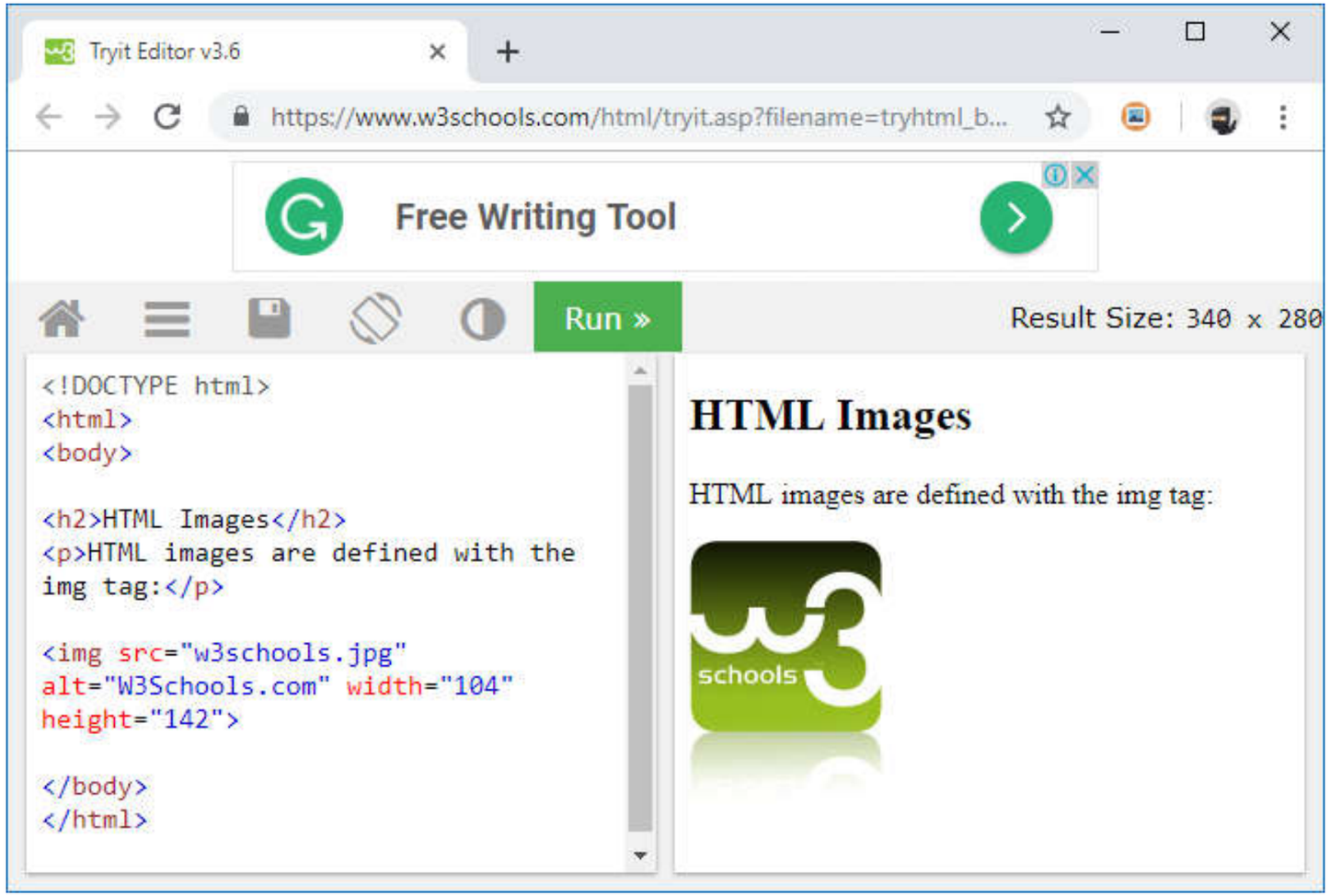
- If you specify only the preferred width (or the preferred height) the image will conserve the aspect ratio.

- If you specify both, then it may distort the image unless the values are precise multiples of the aspect ratio of the image.

- Example:

```
<img src="w3schools.jpg" alt="W3Schools Logo" width="104" height="142">
```

Example


Tryit Editor v3.6
https://www.w3schools.com/html/tryit.asp?filename=tryhtml_b...
Free Writing Tool
Run »
Result Size: 340 x 280
<!DOCTYPE html>
<html>
<body>
<h2>HTML Images</h2>
HTML images are defined with the
img tag:
<img src="w3schools.jpg"
alt="W3Schools.com" width="104"
height="142">
</body>
</html>
HTML Images
HTML images are defined with the img tag:
schools

HTML Lists

- HTML list are defined with the **<ol>** or **<ul>** tags.
- The **<ol>** tag is used for **o**rdered **l**ists (or *numbered lists*).
- The **<ul>** tag is used for **u**nordered **l**ists (or *bulleted lists*).
- The list items themselves are contained within **<li>** tags or within nested **<ol>** or **<ul>**.

HTML Lists

- Ordered lists `<ol>` may have the `type` attribute with the following values:
  - `"1"`: for numbers (default for all levels)
  - `"A"`: for uppercase letters
  - `"a"`: for lowercase letters
  - `"I"`: for uppercase roman numbers
  - `"i"`: for lowercase roman numbers

- **NOTE**: The CSS `list-style-type` property provides more options for enumerations and decorations.

CSS List-Style-Type Values

Value	Description
`disc`	Default value. The marker is a filled circle
`armenian`	The marker is traditional Armenian numbering
`circle`	The marker is a circle
`cjk-ideographic`	The marker is plain ideographic numbers
`decimal`	The marker is a number
`decimal-leading-zero`	The marker is a number with leading zeros (01, 02, 03, etc.)
`georgian`	The marker is traditional Georgian numbering
`hebrew`	The marker is traditional Hebrew numbering

CSS List-Style-Type Values

Value	Description
`hiragana`	The marker is traditional Hiragana numbering
`hiragana-iroha`	The marker is traditional Hiragana iroha numbering
`katakana`	The marker is traditional Katakana numbering
`katakana-iroha`	The marker is traditional Katakana iroha numbering
`lower-alpha`	The marker is lower-alpha (a, b, c, d, e, etc.)
`lower-greek`	The marker is lower-greek
`lower-latin`	The marker is lower-latin (a, b, c, d, e, etc.)
`lower-roman`	The marker is lower-roman (i, ii, iii, iv, v, etc.)

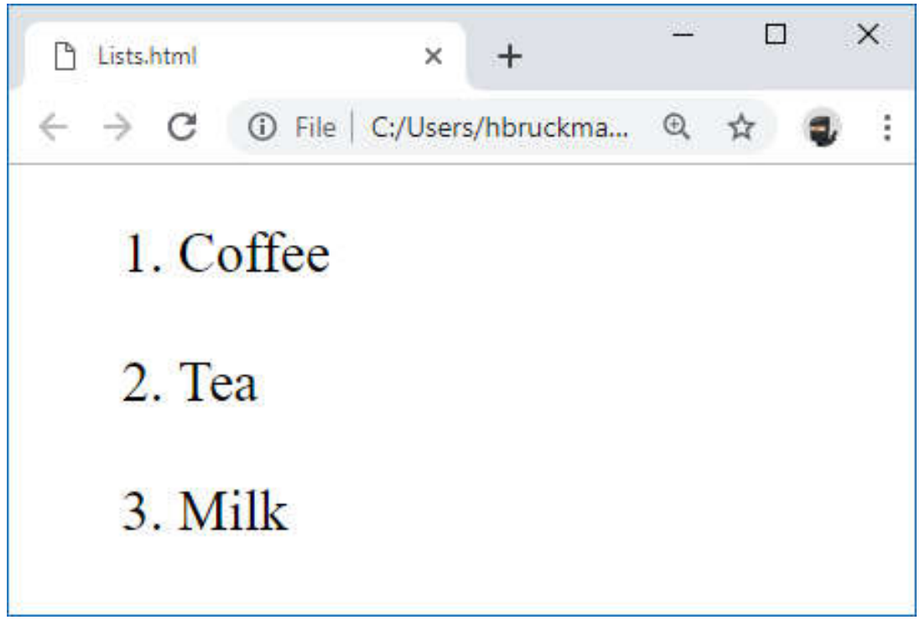
CSS List-Style-Type Values

Value	Description
`none`	No marker is shown
`square`	The marker is a square
`upper-alpha`	The marker is upper-alpha (A, B, C, D, E, etc.)
`upper-greek`	The marker is upper-greek
`upper-latin`	The marker is upper-latin (A, B, C, D, E, etc.)
`upper-roman`	The marker is upper-roman (I, II, III, IV, V, etc.)
`initial`	Sets this property to its default value. Read about initial
`inherit`	Inherits this property from its parent element. Read about inherit

HTML Lists

- **O**rdered **L**ist with **Default** Enumeration Example:

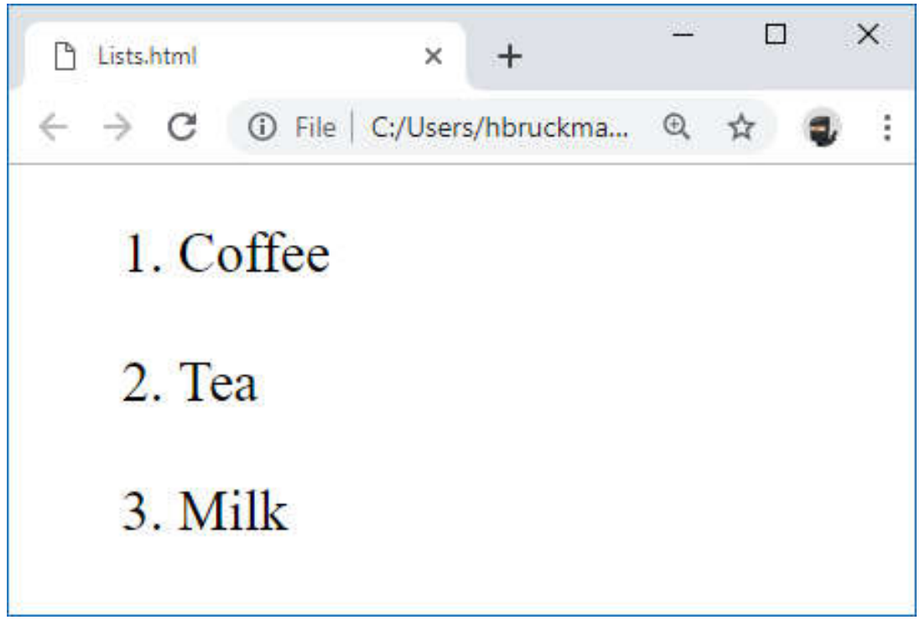
```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



HTML Lists

- **O**rdered **L**ist with **Numbers**
  Enumeration Example:

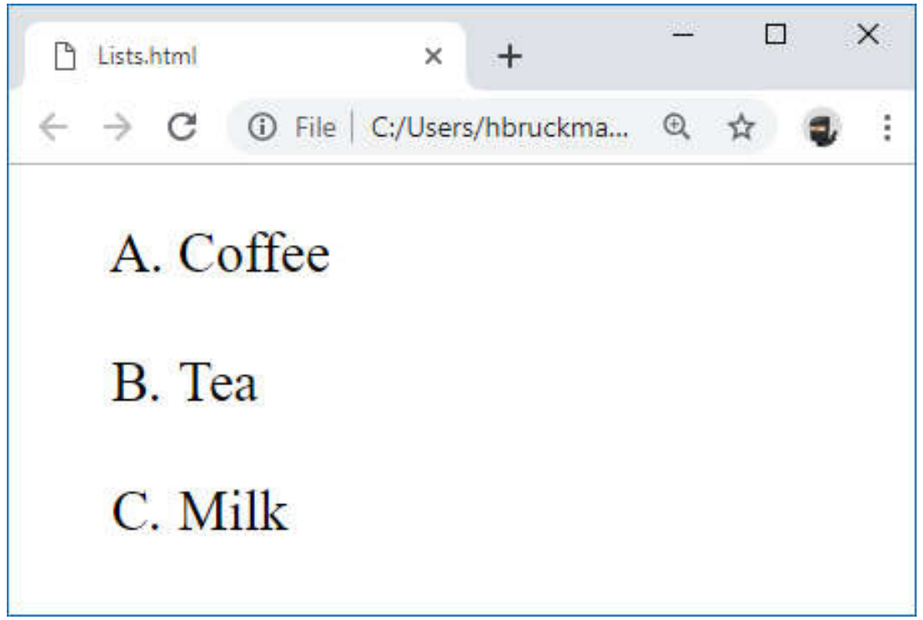
```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



HTML Lists

- **O**rdered **L**ist with **Uppercase Letter** Enumeration Example:

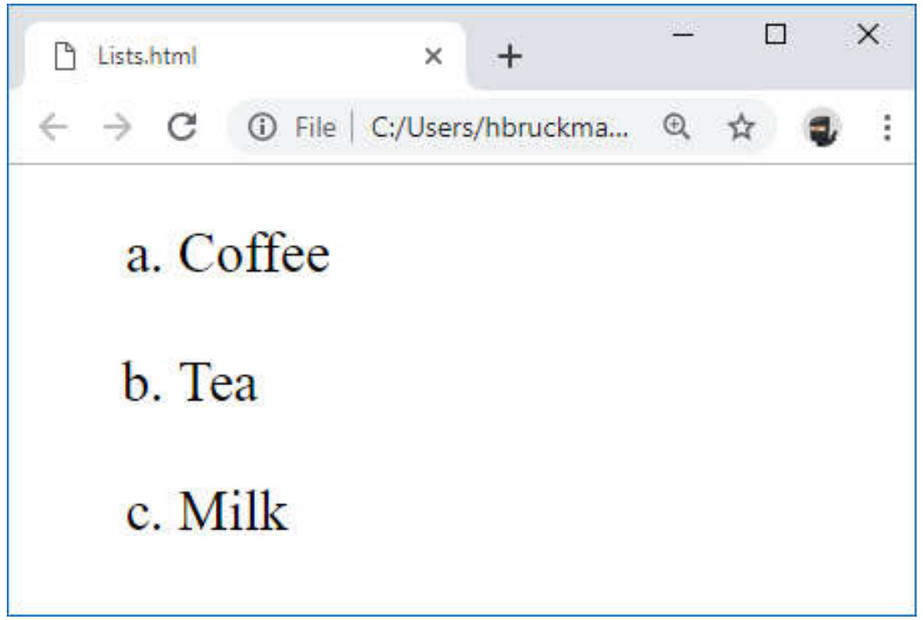
```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



HTML Lists

- **O**rdered **L**ist with **Lowercase Letter** Enumeration Example:

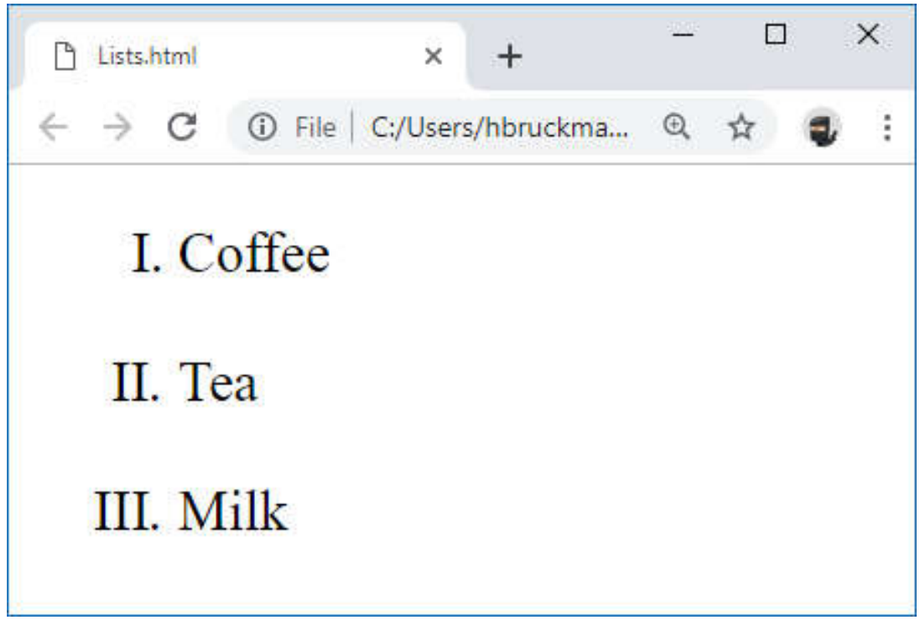
```
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



HTML Lists

- **O**rdered **L**ist with **Uppercase Roman** Enumeration Example:

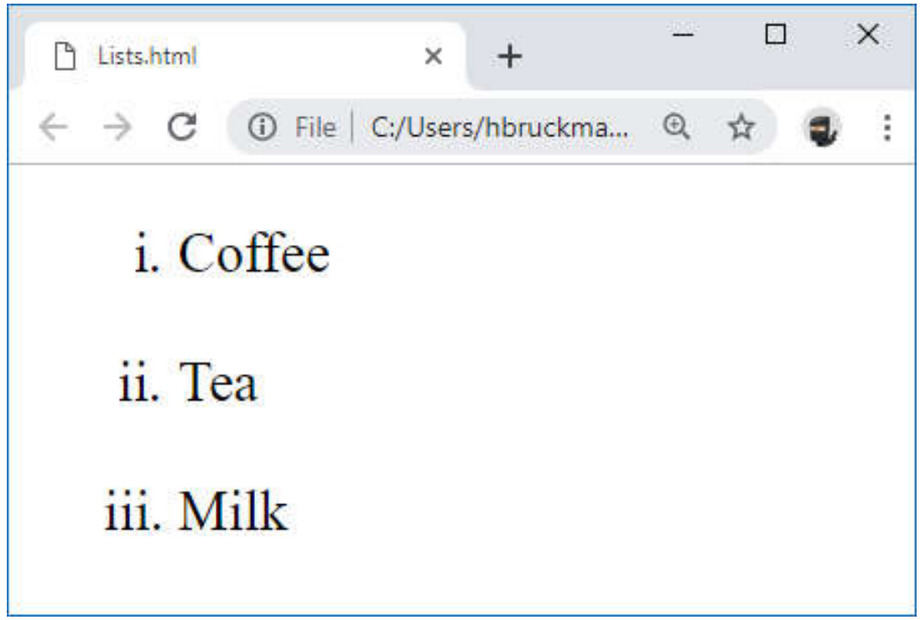
```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



HTML Lists

- **O**rdered **L**ist with **Lowercase Roman** Enumeration Example:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



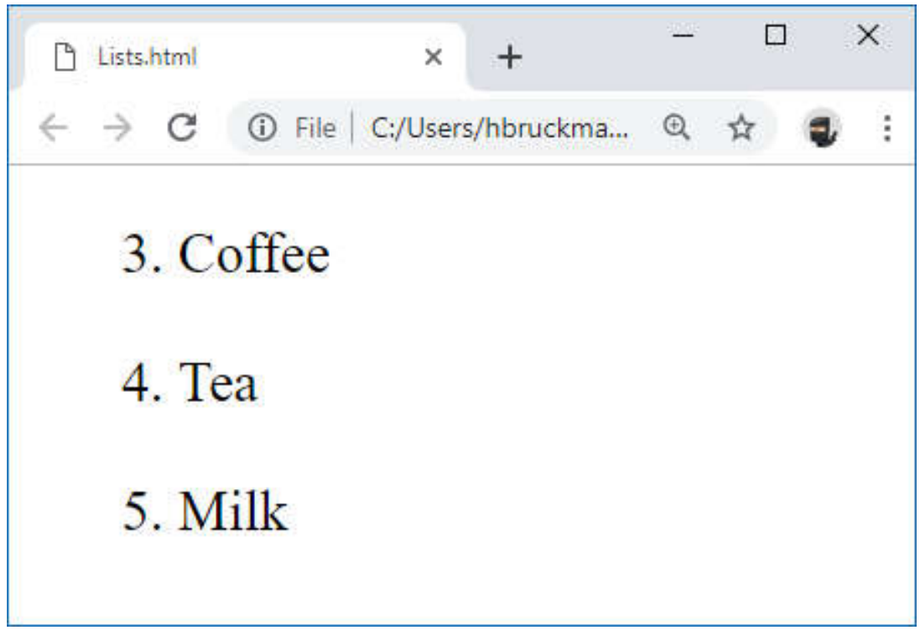
HTML Lists

- Ordered lists `<ol>` may have the `start` attribute to specify the number at which the list should begin counting.
  - `<ol start="3">`
- Ordered lists `<ol>` may have the `reversed` attribute to specify that the order of the enumeration should be backwards.
  - `<ol reversed>`
- The list items `<li>` within ordered lists `<ol>` may have the `value` attribute to specify and/or override the number of each item.
  - `<li value="5">`

HTML Lists

- **O**rdered **L**ist with **Starting Number** Enumeration Example:

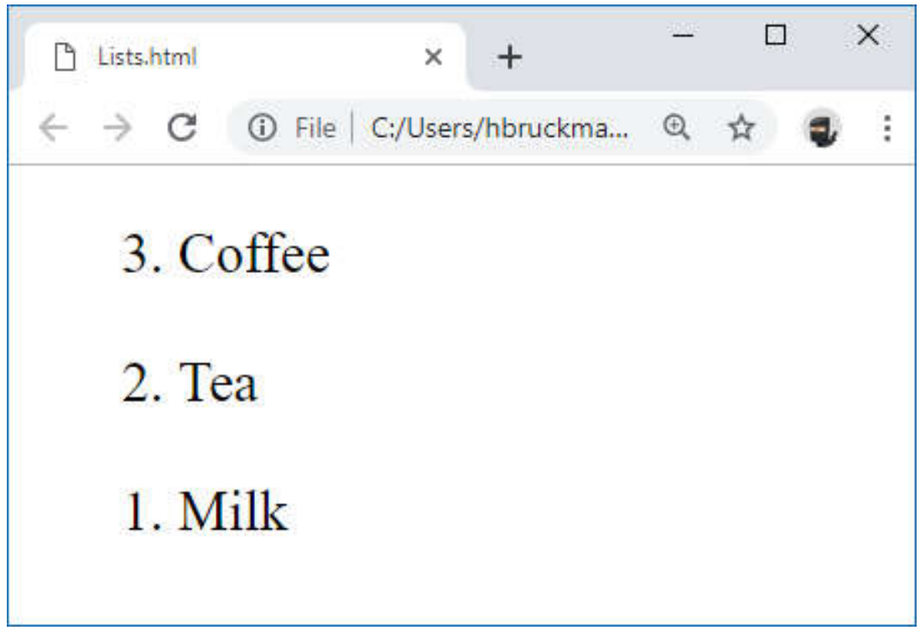
```
<ol start="3">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



HTML Lists

- **O**rdered **L**ist with **Reversed** Enumeration Example:

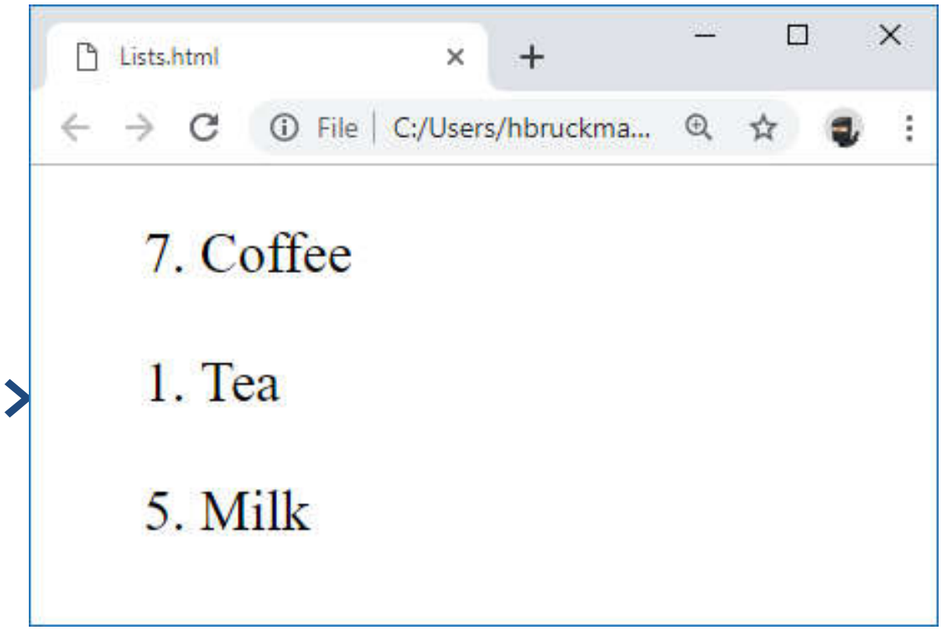
```
<ol reversed>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



HTML Lists

- **O**rdered **L**ist with **Specified Values** Enumeration Example:

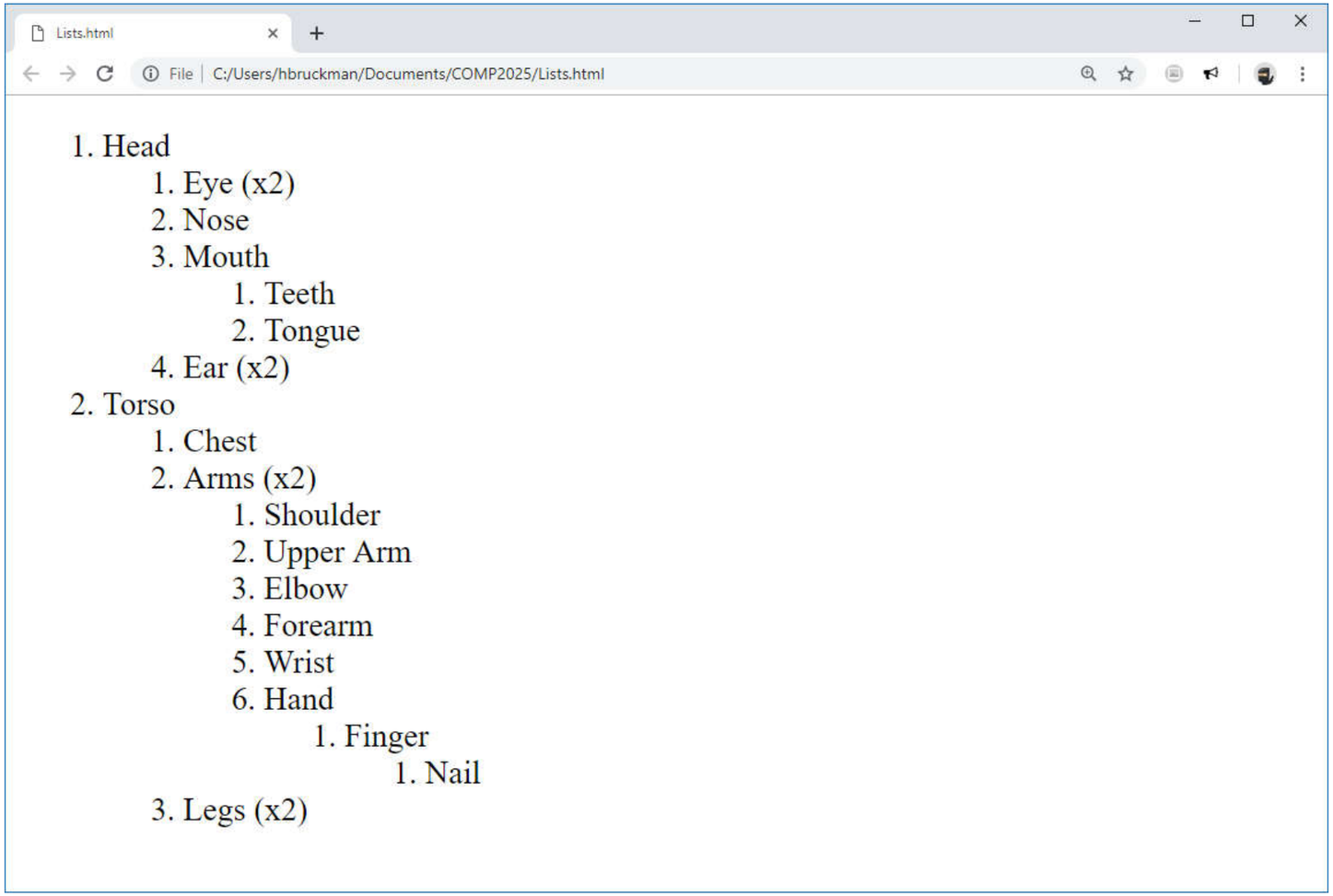
```
<ol>
  <li value="7">Coffee</li>
  <li value="1">Tea</li>
  <li value="5">Milk</li>
</ol>
```



HTML Nested Lists

```
<ol>
  <li>Head</li>
  <ol>
    <li>Eye (x2)</li>
    <li>Nose</li>
    <li>Mouth</li>
    <ol>
      <li>Teeth</li>
      <li>Tongue</li>
    </ol>
    <li>Ear (x2)</li>
  </ol>
  <li>Torso</li>
  <ol>
    <li>Chest</li>
    <li>Arms (x2)</li>
      <ol>
        <li>Shoulder</li>
        <li>Upper Arm</li>
        <li>Elbow</li>
        <li>Forearm</li>
        <li>Wrist</li>
        <li>Hand</li>
        <ol>
          <li>Finger</li>
          <ol>
            <li>Nail</li>
          </ol>
        </ol>
      </ol>
      <li>Legs (x2)</li>
    </ol>
</ol>
```

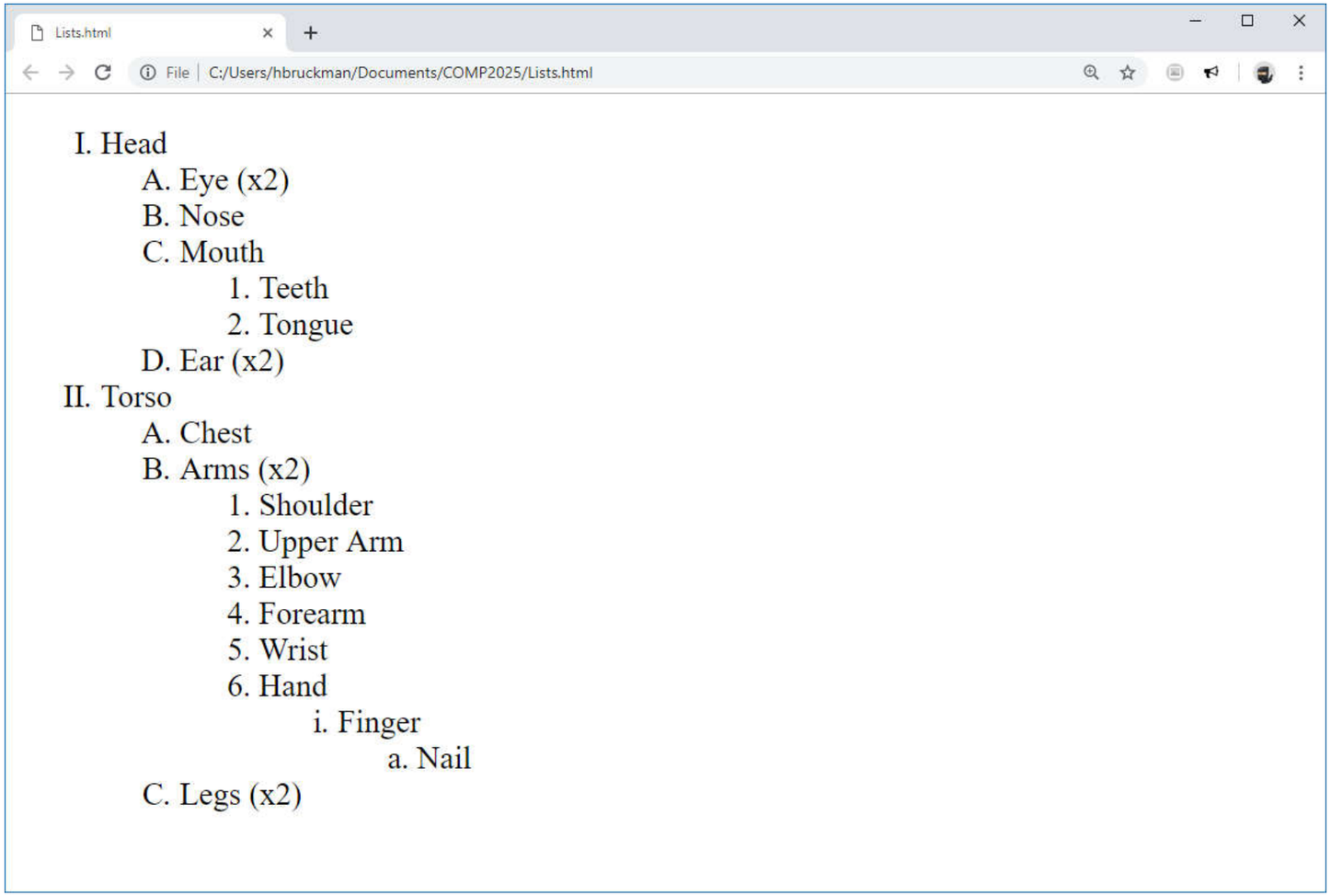
Example


Lists.html
File | C:/Users/hbruckman/Documents/COMP2025/Lists.html
1. Head
1. Eye (x2)
2. Nose
3. Mouth
1. Teeth
2. Tongue
4. Ear (x2)
2. Torso
1. Chest
2. Arms (x2)
1. Shoulder
2. Upper Arm
3. Elbow
4. Forearm
5. Wrist
6. Hand
1. Finger
1. Nail
3. Legs (x2)

HTML Nested Lists

```
<ol type="I">
  <li>Head</li>
  <ol type="A">
    <li>Eye (x2)</li>
    <li>Nose</li>
    <li>Mouth</li>
    <ol type="1">
      <li>Teeth</li>
      <li>Tongue</li>
    </ol>
    <li>Ear (x2)</li>
  </ol>
  <li>Torso</li>
  <ol type="A">
    <li>Chest</li>
    <li>Arms (x2)</li>
    <ol type="1">
      <li>Shoulder</li>
      <li>Upper Arm</li>
      <li>Elbow</li>
      <li>Forearm</li>
      <li>Wrist</li>
      <li>Hand</li>
      <ol type="i">
        <li>Finger</li>
        <ol type="a">
          <li>Nail</li>
        </ol>
      </ol>
    </ol>
    <li>Legs (x2)</li>
  </ol>
</ol>
```

Example


Lists.html
File | C:/Users/hbruckman/Documents/COMP2025/Lists.html
I. Head
A. Eye (x2)
B. Nose
C. Mouth
1. Teeth
2. Tongue
D. Ear (x2)
II. Torso
A. Chest
B. Arms (x2)
1. Shoulder
2. Upper Arm
3. Elbow
4. Forearm
5. Wrist
6. Hand
i. Finger
a. Nail
C. Legs (x2)

HTML Lists

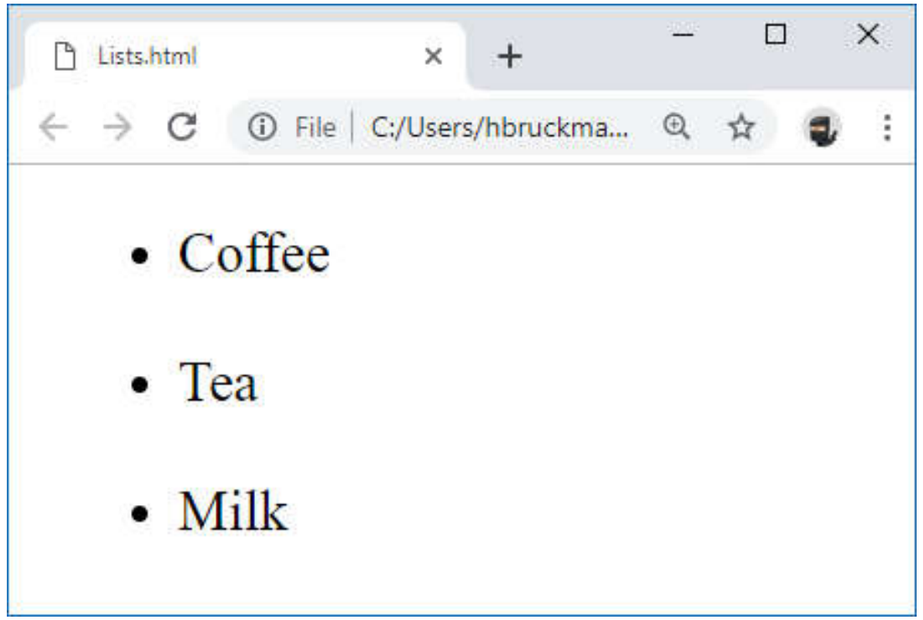
- Ordered list `<ul>` may have the `type` attribute with the following values:
  - `"disc"`: for filled black circles (default 1st level)
  - `"circle"`: for hollow black circles (default 2nd level)
  - `"square"`: for filled black squares (default 3rd level and onward)
  - `"none"`: for no bullet decoration

- **NOTE**: The `type` attribute is not officially supported in `<ul>` tags. Rather, HTML5 uses the CSS `list-style-type` property with the values `circle`, `disc`, `square`, or `none`.

HTML Lists

- **U**nordered **L**ist with **Default** Decoration Example:

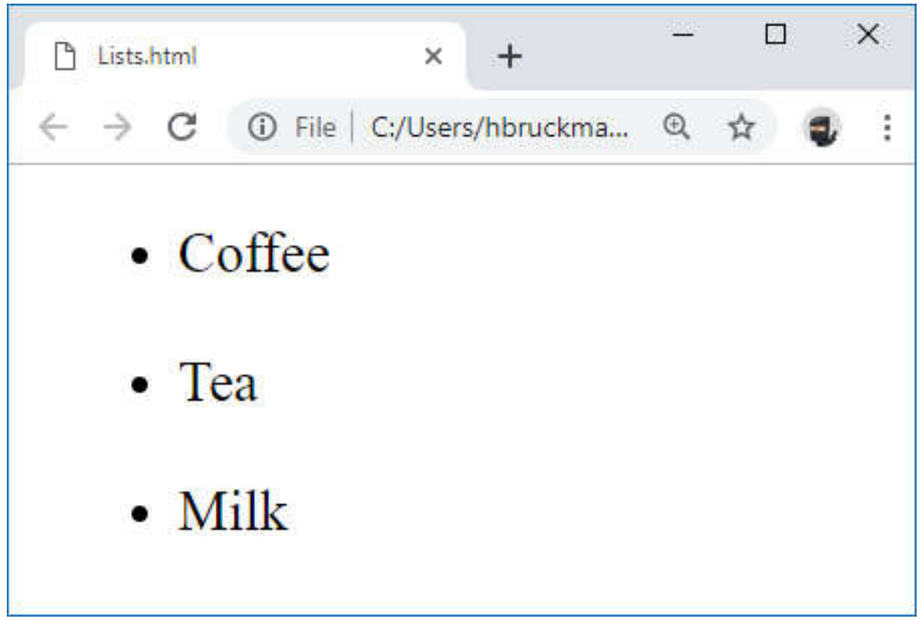
```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



HTML Lists

- **U**nordered **L**ist with **Disc** Decoration Example:

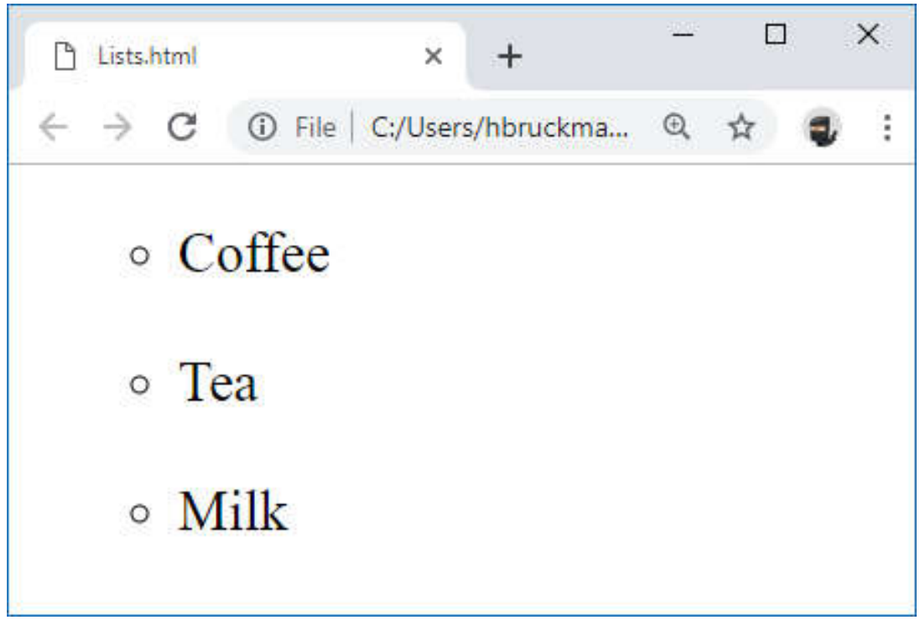
```
<ul type="disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



HTML Lists

- **U**nordered **L**ist with **Circle** Decoration Example:

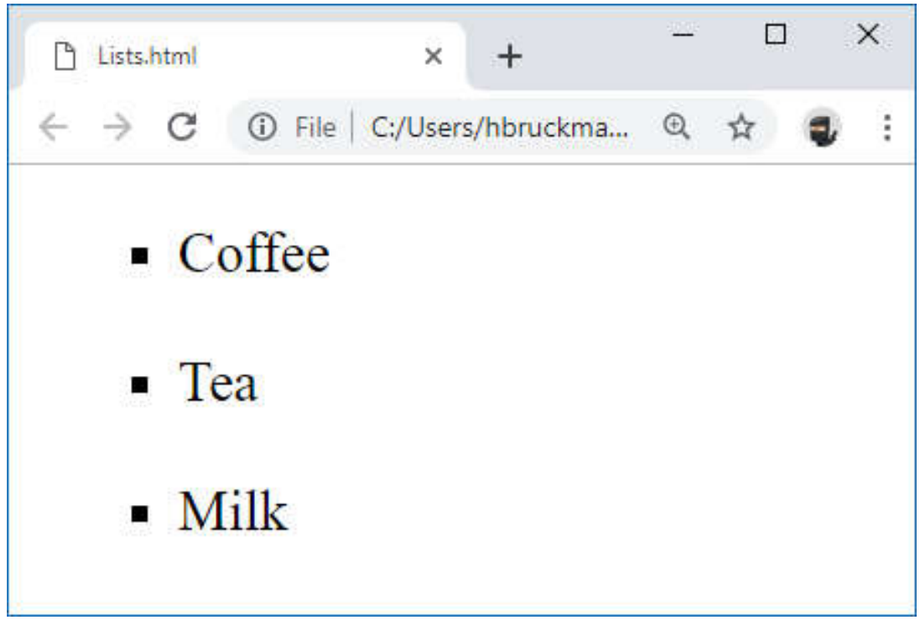
```
<ul type="circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



HTML Lists

- **U**nordered **L**ist with **Square** Decoration Example:

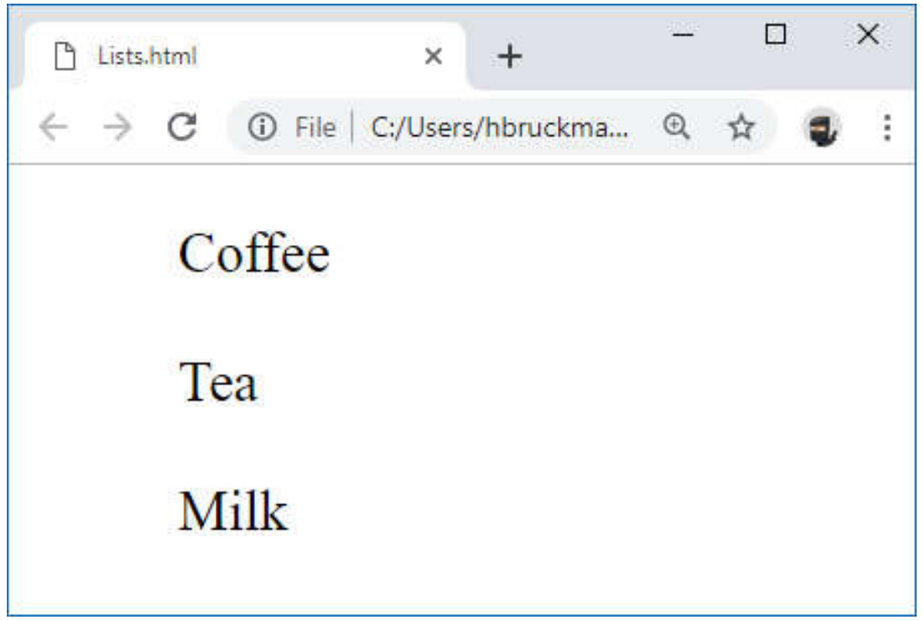
```
<ul type="square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



HTML Lists

- **U**nordered **L**ist with **None** Decoration Example:

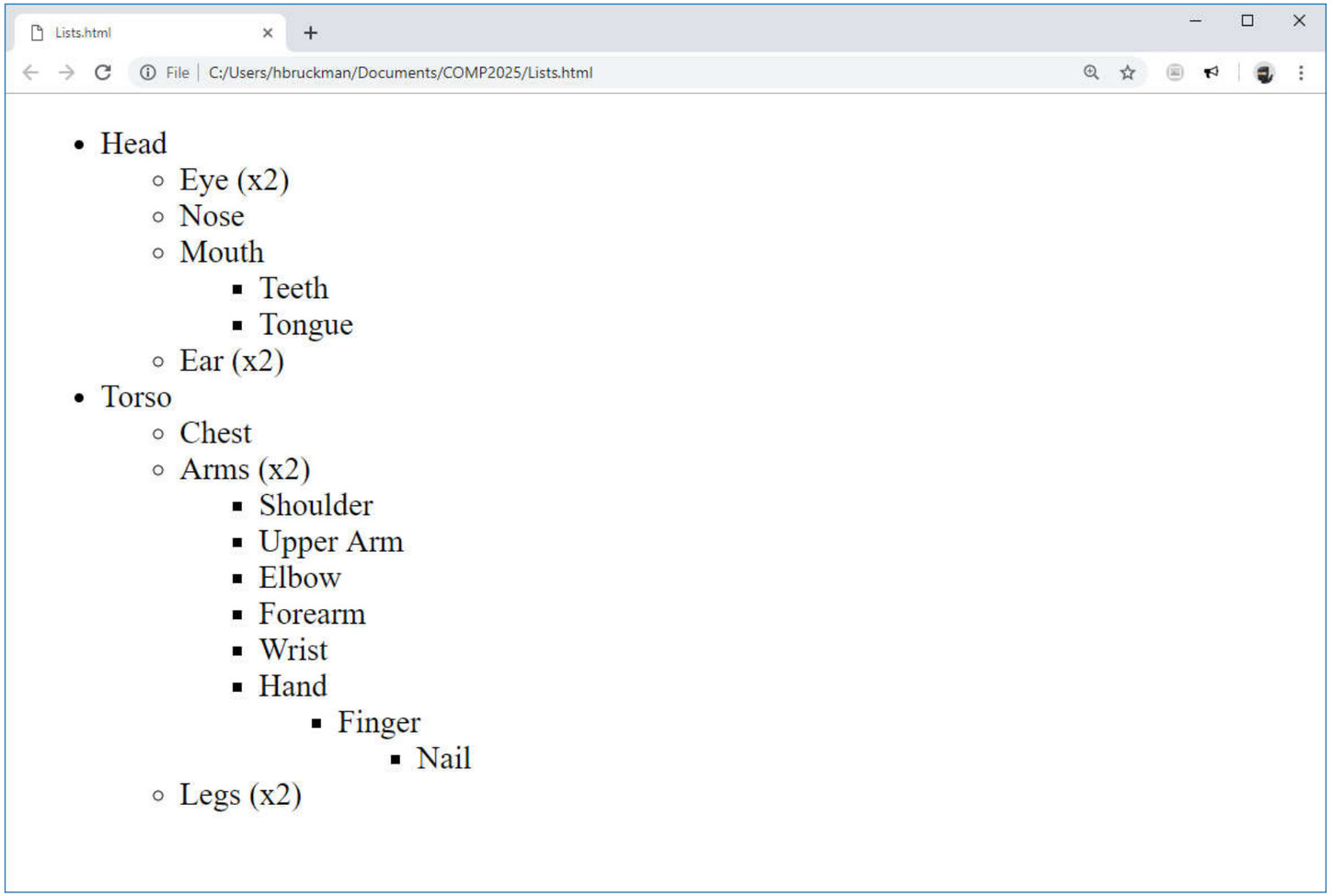
```
<ul type="none">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



HTML Nested Lists

```
<ul>
  <li>Head</li>
  <ul>
    <li>Eye (x2)</li>
    <li>Nose</li>
    <li>Mouth</li>
    <ul>
      <li>Teeth</li>
      <li>Tongue</li>
    </ul>
    <li>Ear (x2)</li>
  </ul>
  <li>Torso</li>
  <ul>
    <li>Chest</li>
    <li>Arms (x2)</li>
    <ul>
      <li>Shoulder</li>
      <li>Upper Arm</li>
      <li>Elbow</li>
      <li>Forearm</li>
      <li>Wrist</li>
      <li>Hand</li>
      <ul>
        <li>Finger</li>
        <ul>
          <li>Nail</li>
        </ul>
      </ul>
    </ul>
    <li>Legs (x2)</li>
  </ul>
</ul>
```

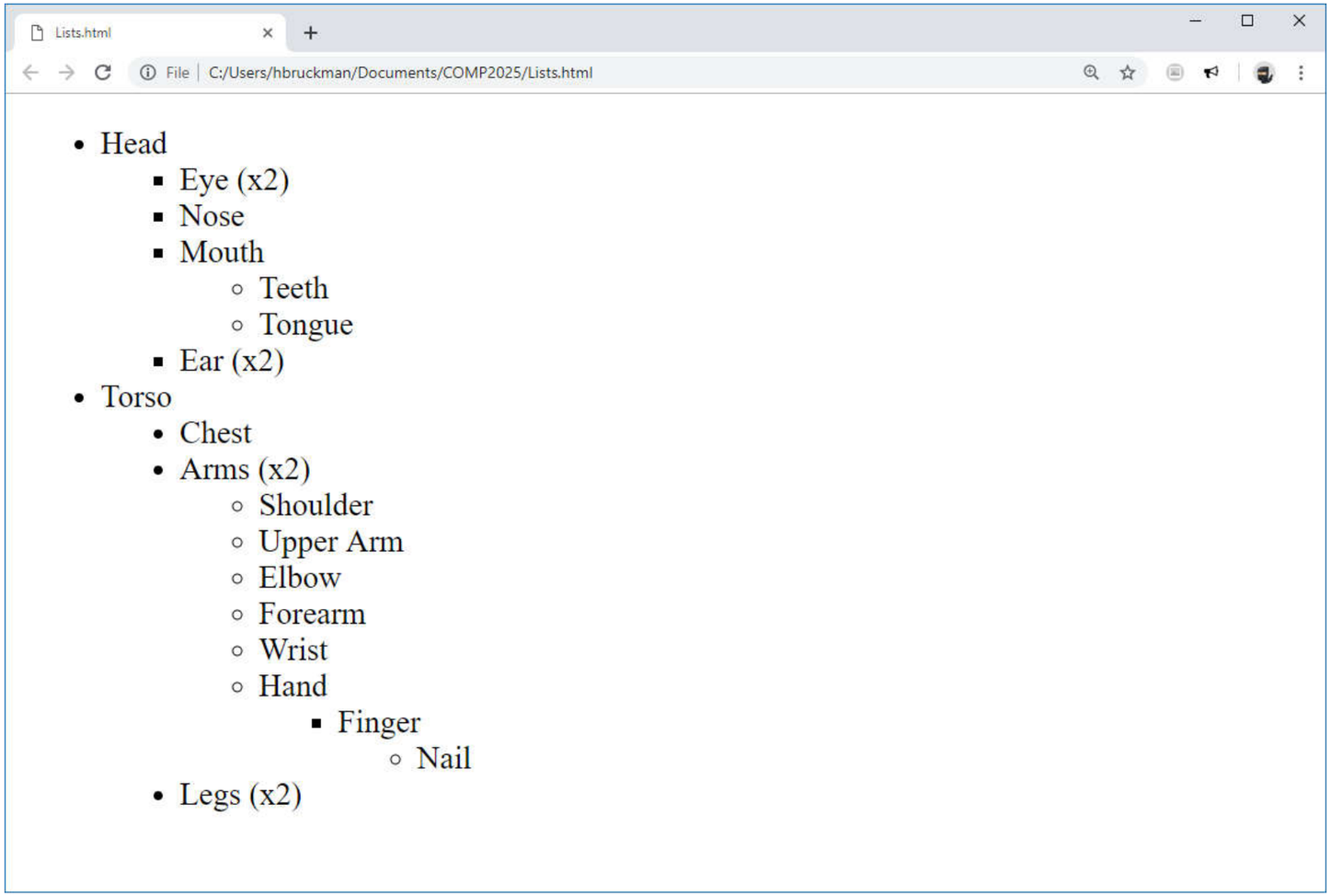
Example


Lists.html
File | C:/Users/hbruckman/Documents/COMP2025/Lists.html
Head
Eye (x2)
Nose
Mouth
Teeth
Tongue
Ear (x2)
Torso
Chest
Arms (x2)
Shoulder
Upper Arm
Elbow
Forearm
Wrist
Hand
Finger
Nail
Legs (x2)

HTML Nested Lists

```
<ul type="disc">
  <li>Head</li>
  <ul type="square">
    <li>Eye (x2)</li>
    <li>Nose</li>
    <li>Mouth</li>
    <ul type="circle">
      <li>Teeth</li>
      <li>Tongue</li>
    </ul>
    <li>Ear (x2)</li>
  </ul>
  <li>Torso</li>
  <ul type="disc">
    <li>Chest</li>
    <li>Arms (x2)</li>
    <ul type="circle">
      <li>Shoulder</li>
      <li>Upper Arm</li>
      <li>Elbow</li>
      <li>Forearm</li>
      <li>Wrist</li>
      <li>Hand</li>
      <ul type="square">
        <li>Finger</li>
        <ul type="circle">
          <li>Nail</li>
        </ul>
      </ul>
    </ul>
    <li>Legs (x2)</li>
  </ul>
</ul>
```

Example


Lists.html
File | C:/Users/hbruckman/Documents/COMP2025/Lists.html
Head
Eye (x2)
Nose
Mouth
Teeth
Tongue
Ear (x2)
Torso
Chest
Arms (x2)
Shoulder
Upper Arm
Elbow
Forearm
Wrist
Hand
Finger
Nail
Legs (x2)

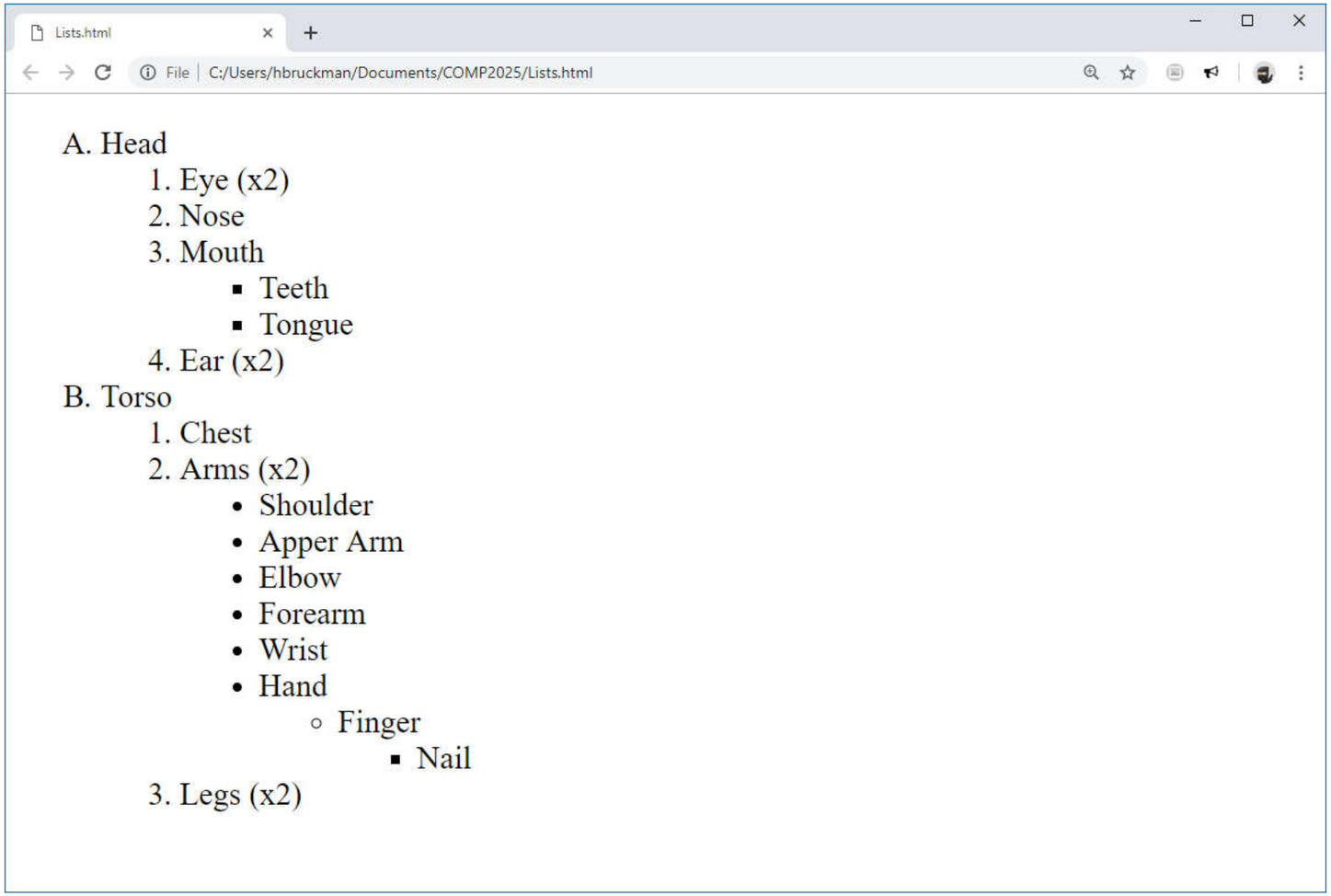
HTML Nested Lists (Mixed)

You can also nest unordered lists within ordered list and vice versa.

HTML Nested Lists (Mixed)

```
<ol type="A">
  <li>Head</li>
  <ol type="1">
    <li>Eye (x2)</li>
    <li>Nose</li>
    <li>Mouth</li>
    <ul>
      <li>Teeth</li>
      <li>Tongue</li>
    </ul>
    <li>Ear (x2)</li>
  </ol>
  <li>Torso</li>
  <ol type="1">
    <li>Chest</li>
    <li>Arms (x2)</li>
    <ul type="disc">
      <li>Shoulder</li>
      <li>Apper Arm</li>
      <li>Elbow</li>
      <li>Forearm</li>
      <li>Wrist</li>
      <li>Hand</li>
      <ul type="circle">
        <li>Finger</li>
        <ul type="square">
          <li>Nail</li>
        </ul>
      </ul>
    </ul>
    <li>Legs (x2)</li>
  </ol>
</ol>
```

Example


Lists.html
File | C:/Users/hbruckman/Documents/COMP2025/Lists.html
A. Head
1. Eye (x2)
2. Nose
3. Mouth
▪ Teeth
▪ Tongue
4. Ear (x2)
B. Torso
1. Chest
2. Arms (x2)
• Shoulder
• Apper Arm
• Elbow
• Forearm
• Wrist
• Hand
◦ Finger
▪ Nail
3. Legs (x2)

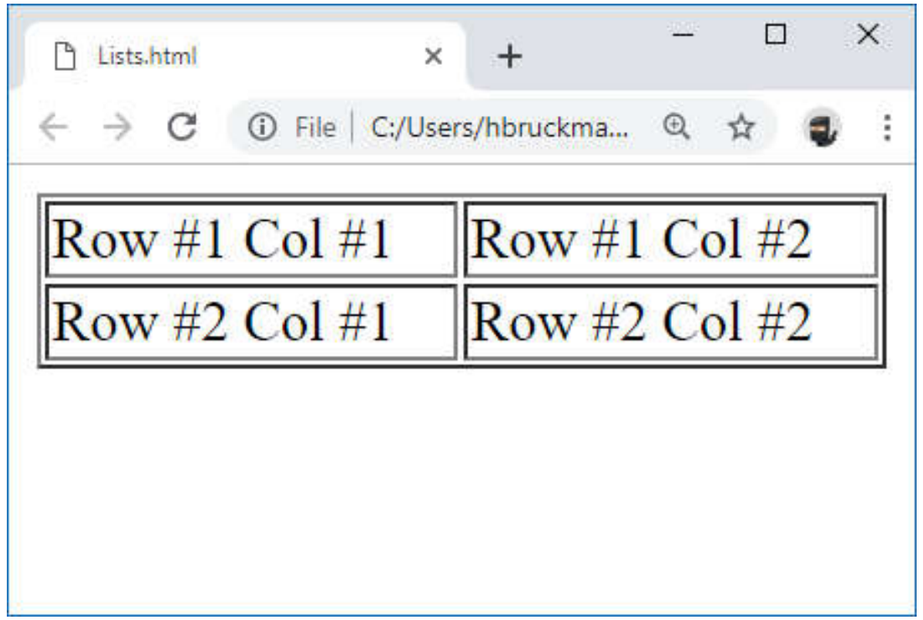
HTML Tables

- HTML tables are created using the `<table>` tag.

HTML Tables

- The most basic way to create a table is:

```
<table border="1">
  <tr>
    <td>Row #1 Col #1</td>
    <td>Row #1 Col #2</td>
  </tr>
  <tr>
    <td>Row #2 Col #1</td>
    <td>Row #2 Col #2</td>
  </tr>
</table>
```



HTML Tables

- **`align`**: sets the horizontal alignment of the table relative to the window. Valid values include [**`"left"`**|**`"center"`**|**`"right"`**]. This attribute is **[deprecated]**, so use the **`margin-left`** and **`margin-right`** CSS properties instead.

- **`bgcolor`**: sets a background color for the table. Valid value formats include [**`"red"`**|**`"#ff0000"`**|**`"rgb(255,0,0)"`**]. This attribute is **[deprecated]**, so use the **`background-color`** CSS property instead.

- **`border`**: with a value of **`"0"`** (default) the table is drawn without a border (for layout purposes); with a value of **`"1"`** the table is drawn with borders (for table purposes). This attribute is **[deprecated]**, so use the **`border`** CSS property instead.

HTML Tables

- **`cellpadding`**: specifies the space between the cell wall and the cell content in pixels. This attribute is **[deprecated]**, so use the **`padding`** CSS property on the **`<th>`** and **`<td>`** instead.

- **`cellspacing`**: specifies the space between cells in pixels. This attribute is **[deprecated]**, so use the **`border-spacing`** CSS property instead.

- **`width`**: sets the width of the table either in pixels or in percentage of the window. This attribute is **[deprecated]**, so use the **`width`** CSS property instead.

HTML Tables

```
<table border="1">
  <caption align="bottom">Figure 1</caption>
  <colgroup>
    <col style="background-color: lightgray">
    <col span="3">
    <col style="background-color: lightblue">
  </colgroup>
  <thead>
    <tr>
      <th>Name</th>
      <th>Exam #1</th>
      <th>Exam #2</th>
      <th>Total Points</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Pedro</th>
      <td>89</td>
      <td>94</td>
      <td>183</td>
    </tr>
    <tr>
      <th>Maria</th>
      <td>95</td>
      <td>92</td>
      <td>187</td>
    </tr>
    <tr>
      <th>Juan</th>
      <td>90</td>
      <td>80</td>
      <td>170</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Max. Total</th>
      <th>100</th>
      <th>100</th>
      <th>200</th>
    </tr>
  </tfoot>
</table>
```