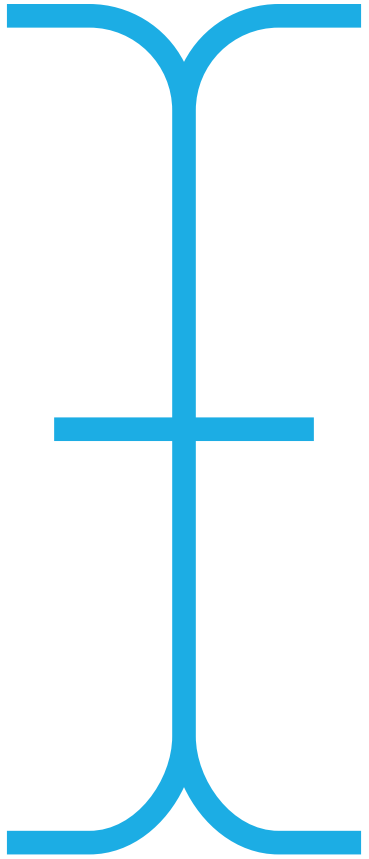
CONTENT TAGS



INTRODUCTION

As we have mentioned many times, HTML is about getting content into your site. We are going to start off simple, getting content onto the site with no real regard for organization other than the order that the content appears.

That's a fancy way of saying this is going to produce ugly sites, but that's okay! Once we have a good understanding of getting content into a site, only then can we really start to understand the proper way to organize or style that content.

Let's make some ugly sites!

TEXT CONTENT

The easiest content to get onto a site is simple text based content. You have many choices when it comes to text content, so let's just list the basics:

1. <p>
   - Used for simple text, no bold, no italic.
2. <i>
   - Used for italic text. Can be nested inside things like p, h1, etc.
3.
   - Used for bold text. Can be nested inside things like p, h1, etc.
4. <h1>, <h2> . . <h6>
   - Used for headers of different size. The largest being h1, smallest h6

And there are many more text based tags that exist! Just know that all text based tags require content and a closing tag.

KNOWLEDGE CHECK

Let's see how we are doing with HTML but also our GitHub / Git workflow. You might need to reference backwards for this one until it really sinks in!

1. Create a new project / folder inside your Scratch directory called HtmlBasics
    1. You can use either mkdir or the projectStarter to accomplish this
2. Open HtmlBasics with VSCode
3. Go through all steps to turn HtmlBasics into a Git repository connected to a GitHub repository
4. Ensure there is an index.html that you can write your code in
5. Add 3 text based tags with some random content inside
6. Add commit and push your code
7. Trying nesting at least 1 tag inside **each tag** you created before
8. Add commit and push your code

NAVIGATION

Have you ever been on a website and clicked on a link? Of course you have.

Links are used to navigate either between pages of the same website (internal links) or to other websites entirely (external links). Links can also be used to download files as well from a website, or even open up applications on a user's device like email.

While links seems simple, there are some nuances and even different behaviour based on the server the website is running on.

ANCHOR TAGS

The **anchor tag** (a tag) is what we use in HTML to create a link. It follows the generic tag form:

```
<a href="link_location">Content</a>
```

So let's break this down. In the opening tag we have:

```
<a href="link_location">
```

Pretty standard, but what is important here is the `href="link_location"`. This is an attribute of the a tag that tells the browser where to go when the user clicks this link.

This href can take you to another tag on the current page, another page on the current site, or even another site all together.

HREF

The **href** (hyperlink reference) is the attribute that actually tells the a tag what to link to. This attribute can take many forms, so let's break them down.

Internal: An internal link is a link to somewhere on the current website. This can be as simple as another destination on the page (pretty much a scrolling shortcut) or moving from one page to another (like going from the home page to the menu page).

External: An external link is simply a link to another website. You can link to any public website from your site, this can be for things like official sources or just cool things you want your users to see.

INTERNAL LINKS

Internal links are the most common to find on a webpage. You need to give your users the ability to navigate your page efficiently. Internal links simply take paths as the href value. For example:

```
<a href=”pages/menu.html">Menu</a>
```

This link will go into the pages folder, and link to the menu.html file. Note that this path is **relative** to the user’s current location. This is best described with examples.

An internal link does not need to go to a different page, it could also link to a file for download:

```
<a href=”files/resume.pdf">Resume</a>
```

EXTERNAL LINKS

External links are links that take you outside the webpage.

```
<a href=”https://google.ca">Google</a>
```

This will simply take you to google. Note that it will overwrite the current tab (which you might not want). To open the link a new tab you can add the **target attribute** and give it a value of **_blank:**

```
<a href=”https://google.ca" target=”_blank" >Google</a>
```

You can also link to an external file for download!

ANCHOR MAGIC

Anchor tags can be used to call functionality on the host device when clicked on.

This can be opening the mail application for emails, or even the phone dial application pre loaded with the number.

Examples:

```
<a href=”tel:40355555555">Call Us</a>
```

```
<a href=”mailto:info@gmail.com">Email Us</a>
```

Notice that the href values start with the special **mailto:** and **tel:** . This informs the browser that this link is supposed to open up a supported application for mail or phoning.

KNOWLEDGE CHECK

Let's add some links to our working project (HtmlBasics)

1. Create a new HTML file called menu.html and add the basic structure plus some text based content on this page
2. Add an **a tag** that will take you from index.html to menu.html
3. Add an **a tag** that will take you from menu.html to index.html
4. Add an **a tag** on either page that will take you to GitHub in a new tab
5. Add commit and push your code.

IMAGES

The img tag is an example to what is called a “void tag”. This is another way of saying it is a tag with no content or closing tag.

The opening tag functions the same as any other tag and can be given attributes. The difference here is that the img tag is entirely dependant on its attributes to function

To get the basic functionality of an image, use the following attributes:

- **src :** The source of the image. This could be a URL to the internet or a path on the already connected to server (just like an a tag)
- **alt :** The alternative tag is used for when the image does not load or when your user is sight impaired and a screen reader is being used. They are also used in SEO, so don’t skip this attribute.

THE SRC ATTRIBUTE

When you want to show an image on the page, the website needs to know where to get the image from. To do this we use the src attribute. The src attribute acts a lot like the href attribute from the a tag. You simple give a internal or external link to an image, and the browser shows it!

This also means that for internal links we need to be careful with our pathing! Depending on the types of paths you use (relative vs absolute) different pages might have different paths to images.

```
<img src=”images/dog" alt="A happy dog">

<img
src="https://media3.giphy.com/media/MeIucAjPKoA120R7sN/200w.webp?cid=ecf05e47bc4lzhu8ua66kuw2jaqsm1pmgokwhwc6behh8mjn&rid=200w.webp&ct=g" alt="A happy clapping cartoon bunny">
```

ASPECT RATIO

The image aspect ratio is the ratio between the height and width of an image.

This ratio must be maintained when resizing the image or you will start to see the image stretch. What this really means is if you change an image's height (let's say you double it), you must also change the image's width by the same factor (double it as well).

The ratio is set by the camera that took the image or the cropping done in editing the image.

ACCEPTED FILE TYPES

Images can be lots of different file types. These file types can vary in their compression saving bandwidth at a loss of quality.

- JPEG
- GIF
- PNG
- APNG
- SVG
- BMP
- BMP ICO.
- PNG ICO.

IMAGE GOTCHA

One thing that students tend to struggle with when it comes to images is getting the correct sizing.

If an image is not given a width or height, it will simply take the full original space of the image source. This means that if the original image is huge, it will take up huge amounts of space on the users screen.

Most of the time we want to limit the size of our images on the screen. We will learn the CSS later to control re-sizing an image, but for now just try to avoid giant image sources.

Pro tip: If you are using local images, there is a VSCode extension called **Luna Paint** that allows you to resize images and even reduce their quality if the file is too large!

VECTOR GRAPHICS

Using vector images is common in 2D graphics and can even be done dynamically on the browser. There are often called SVG images and your browser supports them like any other image.

So if you are on a site for free images and you see an SVG option, this is a good one since they are small in file size and can re-size without losing any quality (unlike a picture taken with a camera).

There is a special HTML tag called “canvas” that allows you to write custom JS that will interact with svgs to create very smooth animations.

https://codepen.io/iamminn/pen/rGaWoK

FREE IMAGE RESOURCES

There are a few places on the internet that you can get free images from. This means you are free to download the images and use them at will (unlike many images you will find on Google).

Just note that some of them do ask that you credit the artist in your code:

- https://iconoir.com/
- https://www.reshot.com/
- https://unsplash.com/
- https://www.pexels.com/

Also, when copying an image from the web to use as an external image, make sure you select the "**Copy Image Address**" option when you right click it! Anything else will give you something you don't want.

KNOWLEDGE CHECK

Let's add some images to our working project (HtmlBasics)

1. Add an **img tag** to an external image
2. Add an **img tag** to an internal image (you will probably have to download one and move it to the project folder)
3. Add commit and push your code.

HTML content is fairly straightforward as long as you take some time to practice.

What get's really interesting is seeing how different tags interact with each other!