Linking and Navigation

by Sophia

WHAT'S COVERED

In this lesson, you will learn about the <a> anchor tag and its ability to create hyperlinks that link to other pages and resources. Furthermore, you will learn about how anchor tags are organized and used to create a navigation menu that allows users to easily navigate the main pages of a site. Finally, you will learn about the footer menu and how it often differs from a site's main navigation menu.

Specifically, this lesson will cover the following:

1. Hyperlinks
2. Navigation Menu
3. Footer Navigation Menu

1. Hyperlinks

Hyperlinks are the tool of navigation on websites. Every element on a page that, when clicked, takes you to a different page is a hyperlink. Hyperlinks, when clicked, make an HTTP request to the web server for a specific webpage. Remember that the HTTP request is a pack of information sent to a web server for the purpose of requesting that the web server return the requested resource. The server prepares the requested page and transmits it to the user's browser. The user's browser then clears the page and renders the new page that it just received.

Hyperlinks are created using the **anchor tag <a>**. The anchor tag surrounds the content that will become the clickable hyperlink, which can be just about anything. Text surrounded by anchor tags is formatted with the classic blue color that you often see on webpages with an underline. Anchor tags require one required attribute and possess a few optional tags as well.

EXAMPLE

```
<a href="index.html">Homepage</a>
```

The required attribute is "href," which is short for "hyper reference." This attribute is set equal to the file or **web path** pointing to the destination page or resource. The path value given to href is either a **relative path** or an

absolute path.

Relative paths point to another file on the same web server and within the same domain name, relative to the current page's location within the web server's file directory. For example, if index.html and about.html are located within the same folder on the server, then a link from the index.html would have the href value of "about.html" or "./about.html". If the page is located in a **subdirectory**, then the link from the homepage to the other would have an href value of "subdirectory/about.html" or "./subdirectory/about.html".

> **IN CONTEXT**
>
> Note that relative paths can either begin with the next directory or filename or begin with the "./" prefix. The "./" is an old convention from command-line interfaces, wherein "./" refers to the current directory as the starting place of the path, as opposed to starting from the very root of a storage drive. Additionally, for context, a "../" double period references the parent directory of the current directory.

Absolute paths are used to create a hyperlink to a page or resource outside of the current web server. For example, if a webpage on a mysite.com domain links to another website on another domain, such as friendsWebsite.com, then you would include in the href an absolute path starting with the HTTP protocol friendsWebsite.com

EXAMPLE

```
href="http://friendsWebsite.com/specificPage.html"
```

In addition to helping navigate to another webpage, anchor tags can also be used to perform a number of other operations such as preparing an email to an email address, creating a clickable phone number for making web calls, and jumping to specific sections within a webpage. To do this, we modify the href value with different prefix keywords or symbols to add different meanings or functions to the clickable hyperlink.

Email Hyperlink	
href="mailto:emailaddress@email.com"	When the href value is prefixed with "mailto:", the link opens an email program or gmail page with a blank email addressed to the email address in the href value.
Phone Hyperlink	
href="tel:+15555551234"	When the href value is prefixed with "tel:", the hyperlink will attempt to open an Internet call application or VoIP application with the specified number ready to dial.
Jumping to a Section on a Page	

href="webpage.html#section2"
href="#section2"
	When you postfix a # symbol and the value of an element's id attribute, the hyperlink will navigate to a page when specified and

	automatically scroll down to the element with the matching id value.
Executing JavaScript	
href="javascript:alert('Hello World!');"	When the href value is prefixed with “javascript:” and there is javascript code after the colon, the hyperlink will execute the code when the user clicks on the hyperlink. This can be used to call javascript functions located at the head of the page or execute javascript commands directly, as in the example, which opens a popup alert window with the message “Hello World.” Note: Notice that within the value surrounded by double quotes, the text ‘Hello World’ is surrounded by single quotes. This prevents the browser from interpreting the Hello World’s leading quote as the terminator for the href value.

Another important attribute to know is the **target** attribute. The default behavior is for the hyperlink to open the linked document in the same window or browser tab. However, you change this by providing one of the following values to the target attribute:

- _blank: This opens a linked document in a new window or tab.
- _parent: This opens a linked document in the parent frame.
- _self: This opens in the same window or tab (default).
- _top: This opens a linked document in the full body of the window.
- framename: When using frames, you can force a linked document to open within a specific frame.

Additional attributes include the following:

Attribute	Description
download	Indicates the linked document should be downloaded instead of rendered in the browser
hreflang	Indicates the language of the linked document
media	Indicates the media or device type for which the linked document is optimized
ping	Specifies a comma-separated list of URLs that get posted to a server in the body of a ping request and is used for tracking
referrerpolicy	Specifies which **referrer** information to send with the hyperlink
rel	Specifies the relationship between the current document and the linked document
type	Specifies the media type of the linked document

WATCH

View the following video for more on HTML hyperlinks.

TERMS TO KNOW

Anchor Tag <a>
An HTML tag used to create a clickable hyperlink.

Web Path
A string of text that indicates the directories and filename of a resource online.

Relative Path
A web path that points to another resource, possibly in other directories, that is relative to the current file's location.

Absolute Path
A fully qualified web path that begins with the HTTP attribute and includes the domain, directories, and filename of the resource.

Subdirectory
A file directory (folder) located within another file directory.

Target
An HTML attribute used to determine what window a resource will be displayed in.

Referrer
The web address (URL) of the previous page that led to the current webpage.

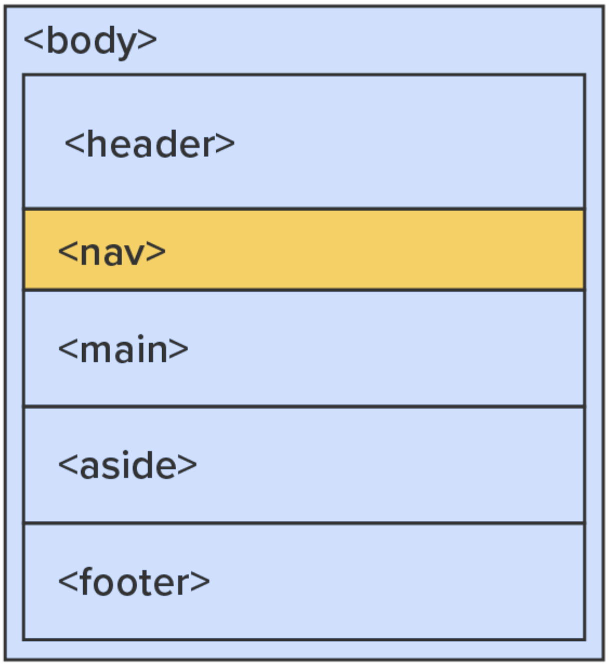
2. Navigation Menu

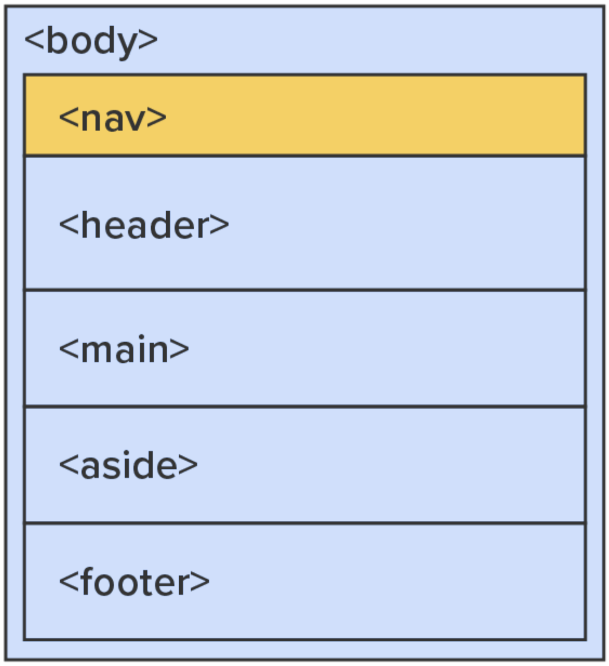
Navigation menus are a cornerstone of any website as they allow users to navigate to the key pages within the site with just a click. Navigation menus can be as simple as an unordered list of anchor tags or an interactive menu with dropdown menus that are created using <div> elements and hyperlinks, are formatted like buttons, and change color when the user hovers their mouse over them.

When considering a navigation menu for the site, there are a couple of aspects that need to be decided. The first is the location: Will the menu be located horizontally across the top of the page or vertically along the left or right side? The second is determined by how many pages need hyperlinks and whether this will be a one-level menu or multilevel menu. One-level menus are just a simple list of clickable hyperlinks with no submenus or dropdown menus. Multilevel menus include either a submenu when a primary hyperlink is hovered over or clicked or a dropdown menu that appears from a primary hyperlink when the user clicks or hovers over it.

Navigation menus can be positioned either above or below the header.

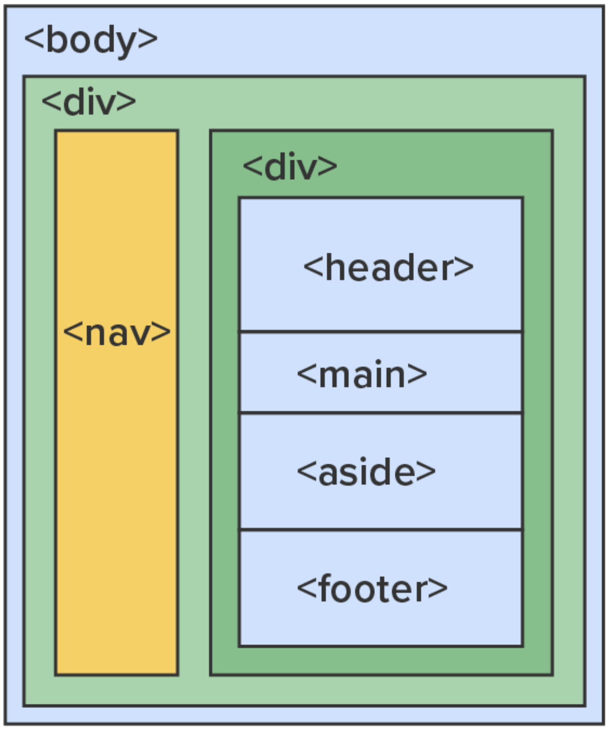
EXAMPLE
The following example shows the navigation menu positioned horizontally around the header. Note that the following examples only look at the <body> section of the page.

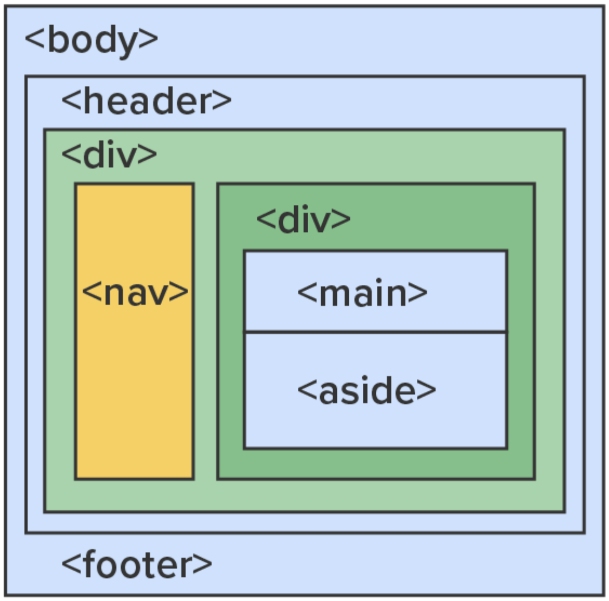




EXAMPLE

The following examples show two different options to position the navigation menu positioned vertically around the header.





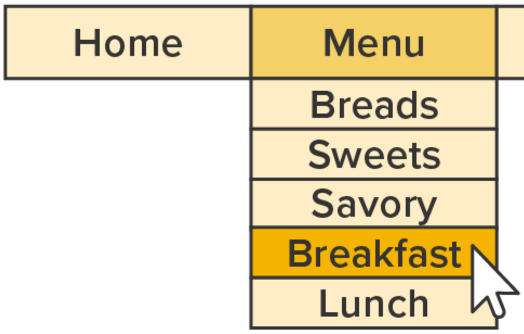
THINK ABOUT IT

Did you notice the additional <div> elements in the last examples above? These divisions are necessary in order to position the sections of content side by side. In HTML, most HTML elements want to stack vertically by default. In order to force items to sit side by side on a page, there are two options. The lesser option is to apply the CSS “float” property, setting one item to float left and the other to float right. The problem here is that you are limited to just two items sitting side by side.

The better option is to surround the items with a division element and set it to be a CSS Flexbox container. That way, you gain much more control over how the children position themselves within the container. Furthermore, this lends itself to responsive web design (RWD) as media queries can easily reposition the children.

Dropdown menus require more CSS setup, which will be covered in the next unit.

EXAMPLE Dropdown menu:

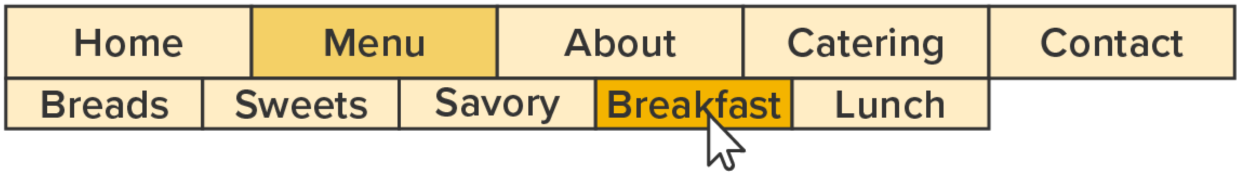


Let’s take a quick look at two different types of multilevel menus. Remember, the examples below have been built in the HTML and stylized using CSS. Creating stylized menus like the example will be covered in a future tutorial.

EXAMPLE Single-level dropdown menu:

Home
Menu
About
Catering
Contact

EXAMPLE Multilevel dropdown menu:

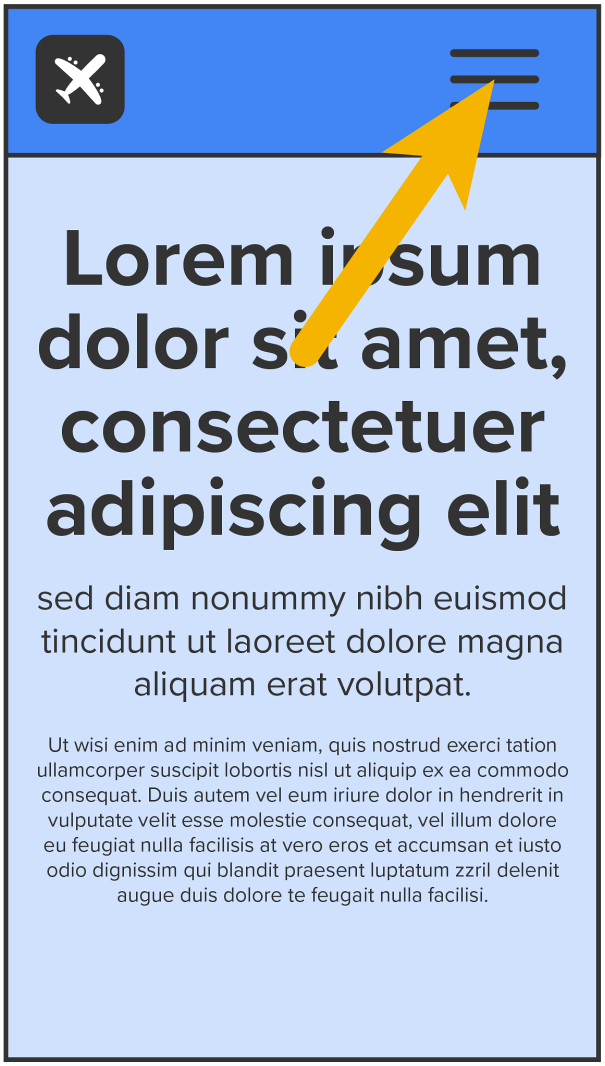


Something else to keep in mind about navigation menus is consistency. It does not make sense to the user, nor does it make it more convenient (in fact, just the opposite), to change the organization or layout of the navigation menu when moving from page to page on a single website. The only exception would be if the site includes a web application that may need to reposition the menu to make more room for the app interface.

Keeping your menus consistent throughout the website is also related to accessibility. When a user is visually impaired, they tend to use their own memory to remember how to get back to the different pages based on the navigation link's location within the menu. When the navigation menu changes from page to page, this makes it more difficult for users with a disability to effectively navigate the site.

Another note about navigation menus is not to make them difficult to locate. CSS can be used on menus to animate them and hide them off-screen. This saves valuable content space for the webpage. However, when hiding a menu off-screen and animating it to slide out, make sure there is a clear indicator that the menu is located off-screen. A commonly used technique is to use a hamburger button that visitors can click on to see the navigation menu.

EXAMPLE Hamburger menu:


Lorem ipsum dolor sit amet, consectetuer adipiscing elit
sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Many websites hide the navigation menu and use the hamburger button for smaller screen sizes to allow more screen room for the website content. You will see how this is accomplished in the next challenge.

TERM TO KNOW

Dropdown Menus
Types of form control that expand when the user clicks them to show additional options that the user can choose from.

3. Footer Navigation Menu

The footer of a website often includes a simplified and sometimes expanded navigation menu. Regardless of how the primary navigation menu is formatted, the footer navigation menu often uses regular hyperlink text that is not formatted to look like buttons, just clickable hyperlinks. Footer hyperlinks are usually organized into related groups and arranged to take up less space and be easy to view.

Additionally, while footer navigation menus contain the same hyperlinks as the main navigation menu, there are often additional hyperlinks in the footer navigation for things like the privacy statement, the site map, hyperlinks to affiliated organizations, the blog, and a calendar of events.

Now that you have learned about the semantic tags and their place in the document, it's time to try it yourself.

TRY IT

Directions:

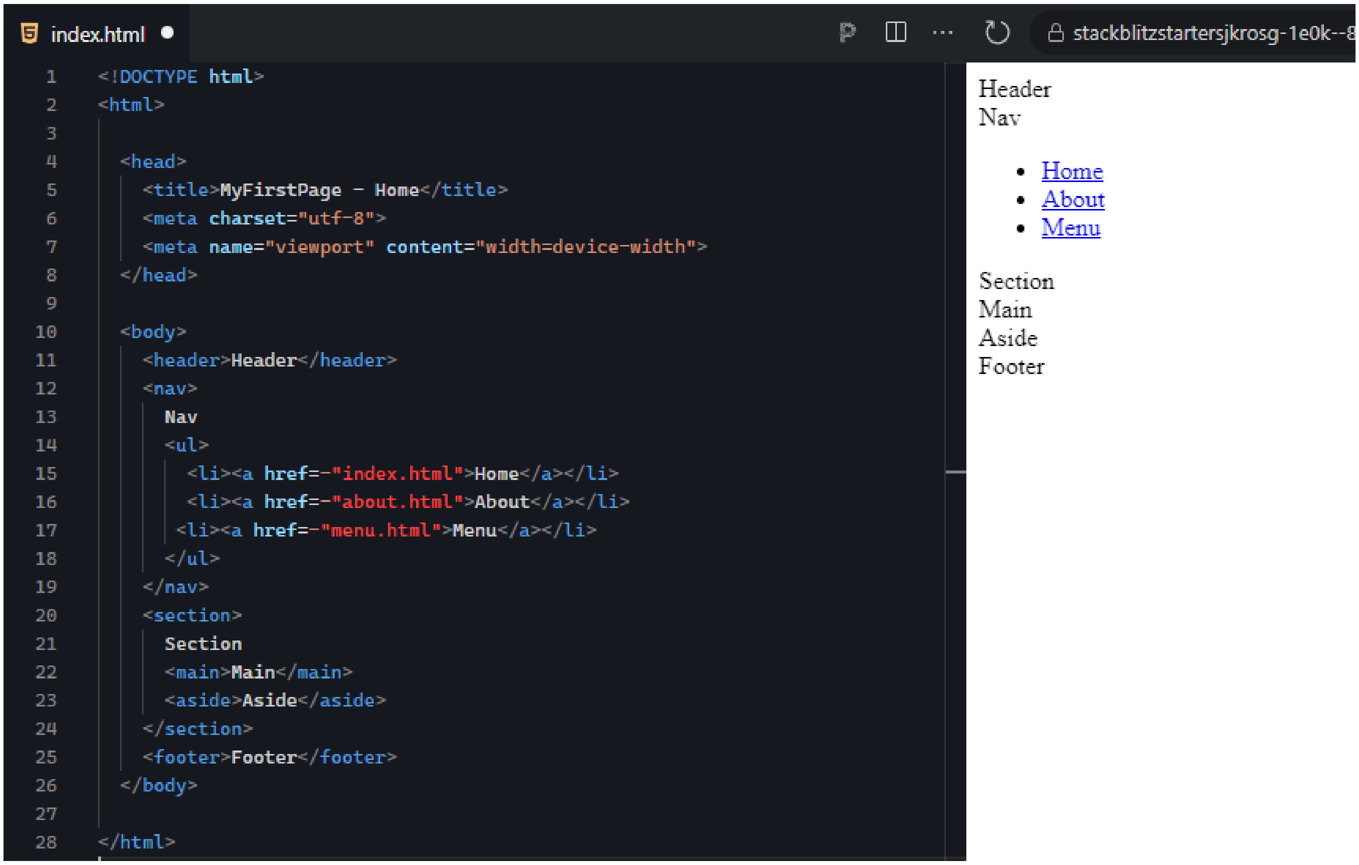
Note: At any time if you do not see the change occur, make sure to select the refresh button in the IDE.

1. Return to your project from the previous lesson, position your insertion point in between the <nav> tags, and create an unordered list with three list items contained within.

2. Within each list item, create an anchor tag with the following attributes and values:

   a. <a href="index.html" >Home</a>
   b. <a href="about.html">About</a>
   c. <a href="menu.html">Menu</a>

This is the expected output:



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

  <head>
    <title>MyFirstPage - Home</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
  </head>

  <body>
    <header>Header</header>
    <nav>
      Nav
      <ul>
        <li><a href=-"index.html">Home</a></li>
        <li><a href=-"about.html">About</a></li>
       <li><a href=-"menu.html">Menu</a></li>
      </ul>
    </nav>
    <section>
      Section
      <main>Main</main>
      <aside>Aside</aside>
    </section>
    <footer>Footer</footer>
  </body>

</html>
```

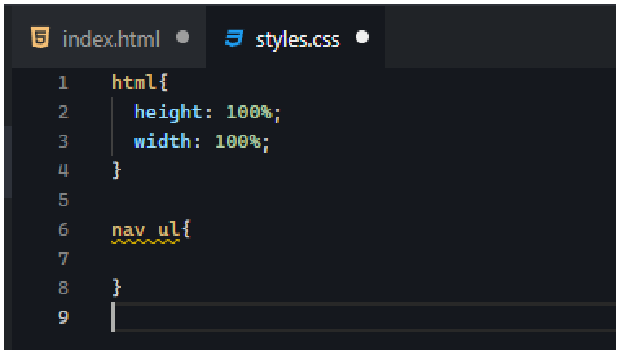
```
Header
Nav

  • Home
  • About
  • Menu

Section
Main
Aside
Footer
```

3. Notice that the bulleted list, per its default behavior, created a vertical list or stack of links with the bullet markers in place. Let’s add a CSS style rule to remove the bullet and change the orientation of the list. First, we need to link the HTML page to the CSS stylesheet. In the <head> section of the page and below the title and meta elements, add the <link> tag with the following attributes: <link href="style.css" rel="stylesheet">.

Remember that the link tag is self-closing.

4. Open style.css and add a style rule with a combinator selector of using “nav” and its child “ul.” This will ensure the style rule only selects the unordered list that is a child of the “nav” element.

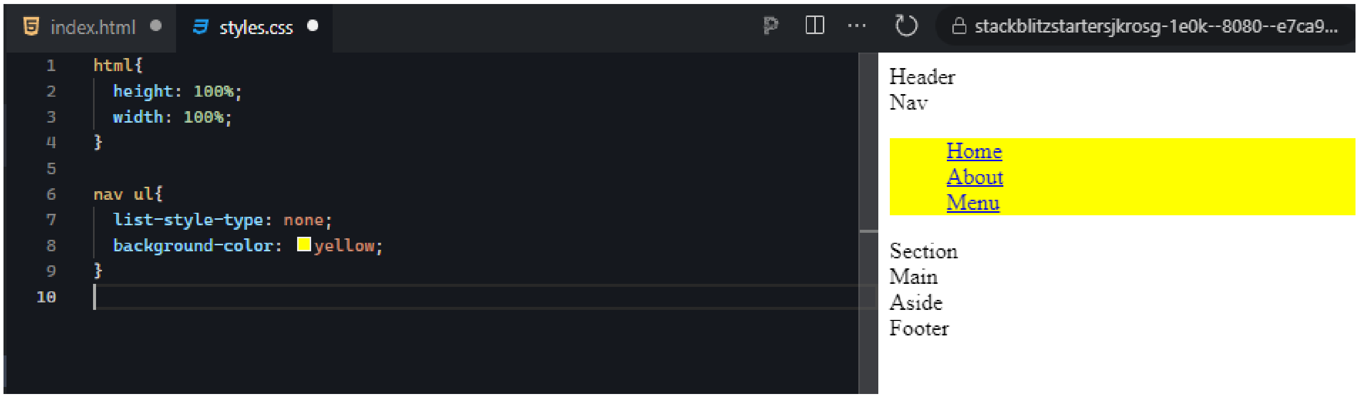
This is the expected code:



```css
html{
  height: 100%;
  width: 100%;
}

nav ul{

}

```

5. Next, let’s override the CSS property “list-style-type” with the value of “none,” and give the list container a value for its background-color property.

This is the expected output:

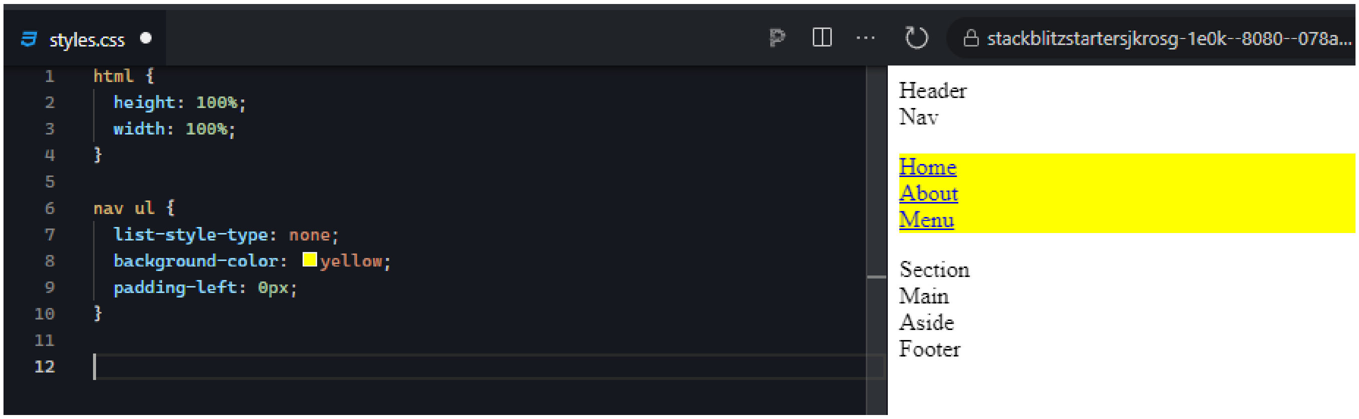


```css
html{
  height: 100%;
  width: 100%;
}

nav ul{
  list-style-type: none;
  background-color: yellow;
}

```

```
Header
Nav

        Home
        About
        Menu

Section
Main
Aside
Footer
```

You may notice the indentation of the navigation links. This is due to a default property within all ordered and unordered lists that adds 40 pixels of padding to the left of the entire list. We can increase or remove this by overriding the padding-left on the list container. In the example below, we are setting the padding-left to 0 pixels.

This is the expected output:

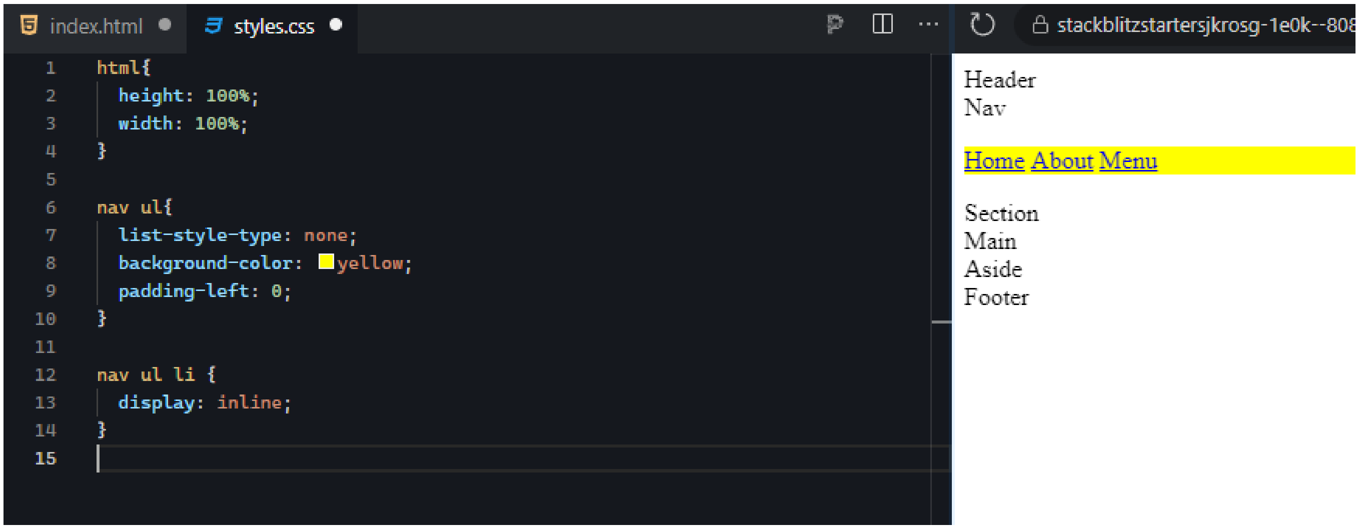


```css
html {
  height: 100%;
  width: 100%;
}

nav ul {
  list-style-type: none;
  background-color: yellow;
  padding-left: 0px;
}


```

```
Header
Nav

Home
About
Menu

Section
Main
Aside
Footer
```

6. Now, let's change the orientation by overriding the list item's CSS “display” property and setting it to “inline.” Setting the display to “inline” will force the list items to automatically arrange themselves and flow more like regular text as opposed to block elements. Also, remember to use a combinator selector to ensure that only these navigation list items will receive the styling.

This is the expected output:

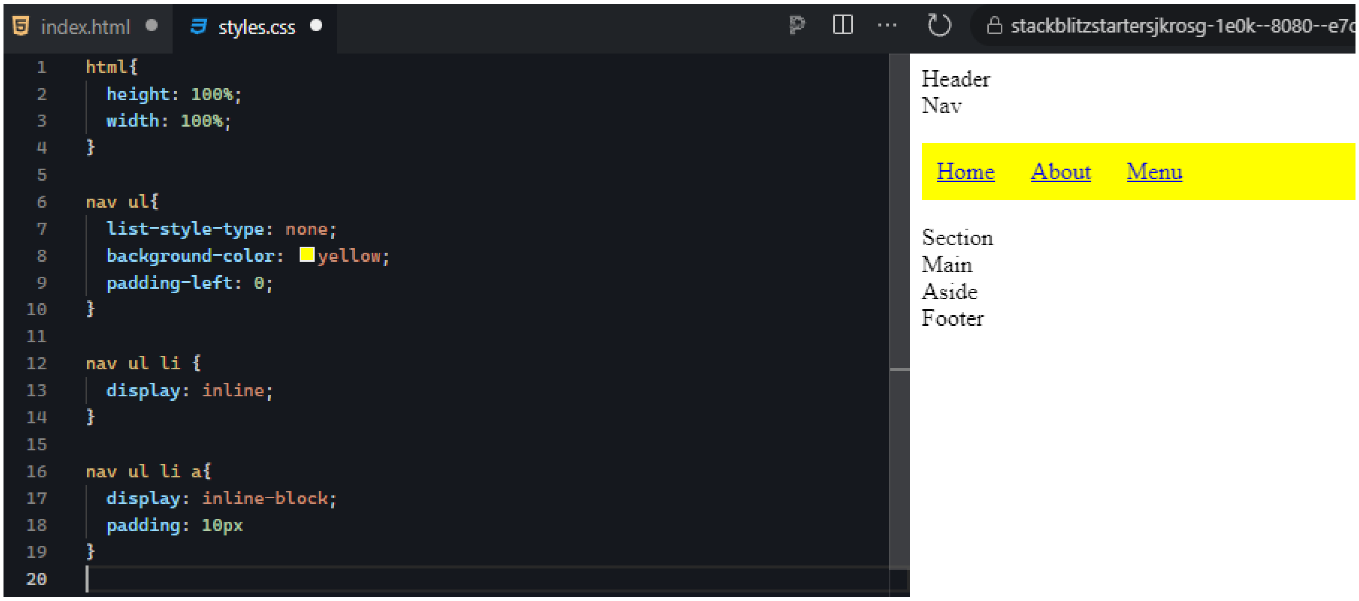


```css
html{
  height: 100%;
  width: 100%;
}

nav ul{
  list-style-type: none;
  background-color: yellow;
  padding-left: 0;
}

nav ul li {
  display: inline;
}

```

```
Header
Nav

Home About Menu

Section
Main
Aside
Footer
```

7. Now that we have adjusted the unordered list and removed the properties we did not want, we can style the anchor links themselves to add height and width to each button. We will do that by adding padding to the anchor tags in the nav list and give it the padding property with a single value of 10 pixels (padding: 10px;). We also want to change the link's display property to inline-block so that their positioning continues to behave more like regular text, yet they now possess height and width.

This is the expected output:



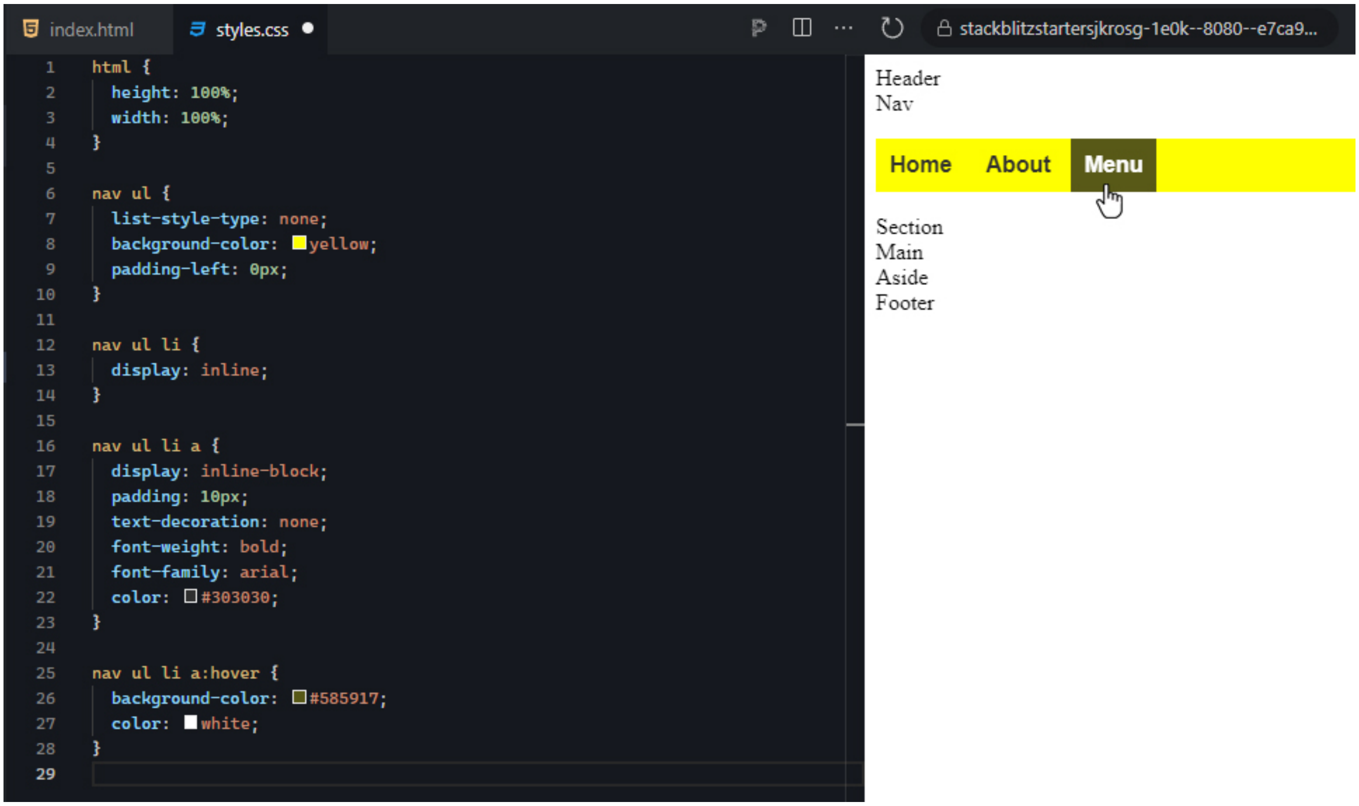
```css
html{
  height: 100%;
  width: 100%;
}

nav ul{
  list-style-type: none;
  background-color: yellow;
  padding-left: 0;
}

nav ul li {
  display: inline;
}

nav ul li a{
  display: inline-block;
  padding: 10px
}

```

```
Header
Nav

Home    About    Menu

Section
Main
Aside
Footer
```

8. The last thing we want to do is to stylize the text and use a CSS pseudo-class to highlight each button when the user's mouse hovers over it.

a. Remove the underline by setting the text-decoration property to none, and set the font-weight to bold, the font-family to arial, and the color to #303030.
b. Then, duplicate the anchor selector you just created for the anchor tags, and add the ":hover" pseudo-class to the "a" tag selector.
c. For the hover style rule, change the background-color property to a dark yellow (#585917) and the text color to white (#FFFFFF).

This is the expected output:



```css
html {
  height: 100%;
  width: 100%;
}

nav ul {
  list-style-type: none;
  background-color: yellow;
  padding-left: 0px;
}

nav ul li {
  display: inline;
}

nav ul li a {
  display: inline-block;
  padding: 10px;
  text-decoration: none;
  font-weight: bold;
  font-family: arial;
  color: #303030;
}

nav ul li a:hover {
  background-color: #585917;
  color: white;
}

```

```
Header
Nav

Home  About  Menu

Section
Main
Aside
Footer
```

You should now have a horizontal navigation bar that responds properly to the mouse hover.

9. If you haven't already done so, now is a good time to save your project.

REFLECT

Navigation bars can be designed in a variety of ways. This example gives you the basic process for setting up a modern navigation menu. Furthermore, utilizing navigation buttons that change color when the user hovers their mouse over them helps users identify navigation menus quickly. You will be able to further customize your navigation menu as you learn more about CSS styling in the next challenge.

SUMMARY

In this lesson, you learned about the <a> anchor tag used to navigational **hyperlinks**. You also learned about how anchor tags are used to construct the primary and footer **navigation menus** for a website. You saw examples of both horizontal and vertical navigation menus and two different examples of multilevel navigation menus. Finally, you learned about how **footer navigation menus** are simplified yet expanded versions of the primary navigation menu.

Source: This Tutorial has been adapted from "The Missing Link: An Introduction to Web Development and Programming " by Michael Mendez. Access for free at https://open.umn.edu/opentextbooks/textbooks/the-missing-link-an-introduction-to-web-development-and-programming. License: Creative Commons attribution: CC BY-NC-SA

TERMS TO KNOW

Absolute Path

A fully qualified web path that begins with the HTTP attribute and includes the domain, directories, and filename of the resource.

Anchor Tag <a>

An HTML tag used to create a clickable hyperlink.

Dropdown Menus

Types of form control that expand when the user clicks them to show additional options that the user can choose from.

Referrer

The web address (URL) of the previous page that led to the current webpage.

Relative Path

A web path that points to another resource, possibly in other directories, that is relative to the current file's location.

Subdirectory

A file directory (folder) located within another file directory.

Target

An HTML attribute used to determine what window a resource will be displayed in.

Web Path

A string of text that indicates the directories and filename of a resource online.