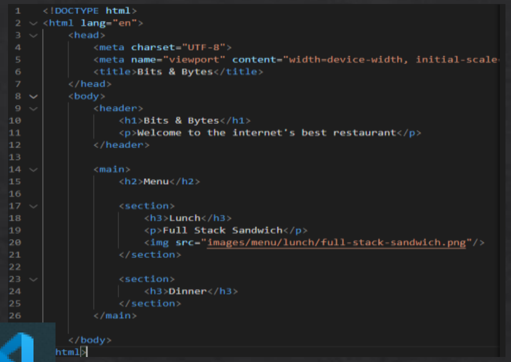
Module 3



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale
        <title>Bits & Bytes</title>
    </head>
    <body>
        <header>
            <h1>Bits & Bytes</h1>
            <p>Welcome to the internet's best restaurant</p>
        </header>

        <main>
            <h2>Menu</h2>

            <section>
                <h3>Lunch</h3>
                <p>Full Stack Sandwich</p>
                <img src="images/menu/lunch/full-stack-sandwich.png"/>
            </section>

            <section>
                <h3>Dinner</h3>
            </section>
        </main>

    </body>
html>
```

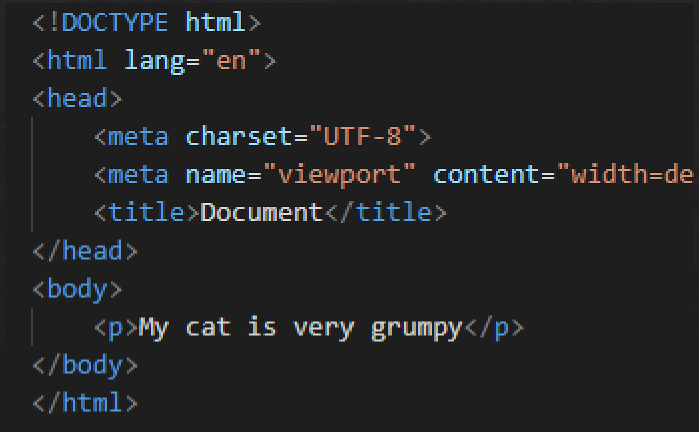
Week 1 - Review

Module 3
Introduction - HTML CSS

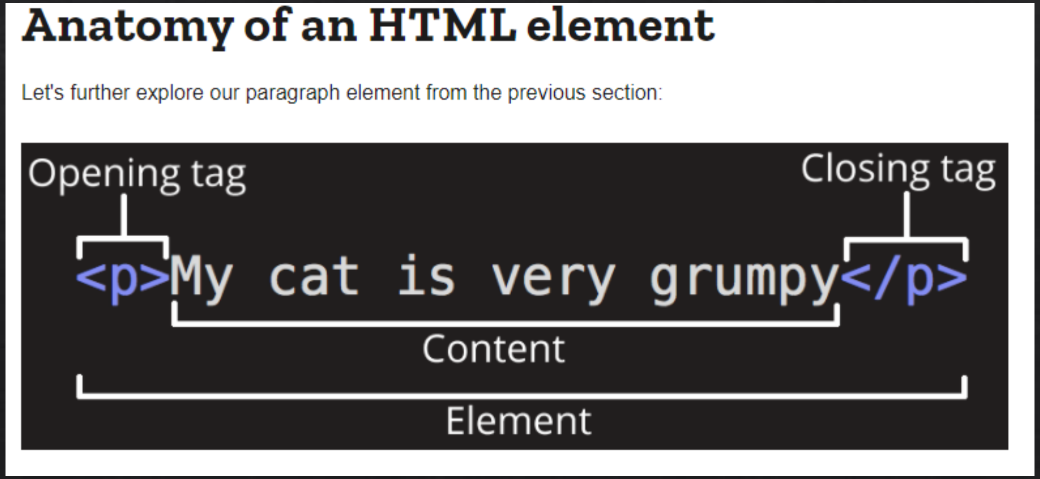
Introduction HTML

HTML stands for **H**yper **T**ext **M**arkup **L**anguage. HTML is a markup language, which means that you take the raw content you want to display and structure it using HTML tags. A web browser needs to be told how to display and structure your content.

Some of the advantages of doing web development with HTML and CSS are that you can start working with the languages pretty quickly, and they have a fast feedback loop.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=de
    <title>Document</title>
</head>
<body>
    <p>My cat is very grumpy</p>
</body>
</html>
```



Anatomy of an HTML element

Let's further explore our paragraph element from the previous section:

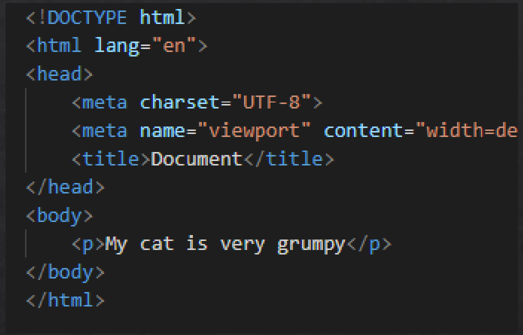
Reference: https://developer.mozilla.org/

Module 3
Introduction - HTML CSS

Elements

These are the building blocks of an HTML page, and are also referred to as nodes.
A few of the elements/nodes shown in the listed code are:

html, body, head
title, div, p etc.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=de
    <title>Document</title>
</head>
<body>
    <p>My cat is very grumpy</p>
</body>
</html>
```

Tags

The tag defines HTML elements, and it is represented within angular brackets < >
example: <html>, <head>, <body>

Attributes

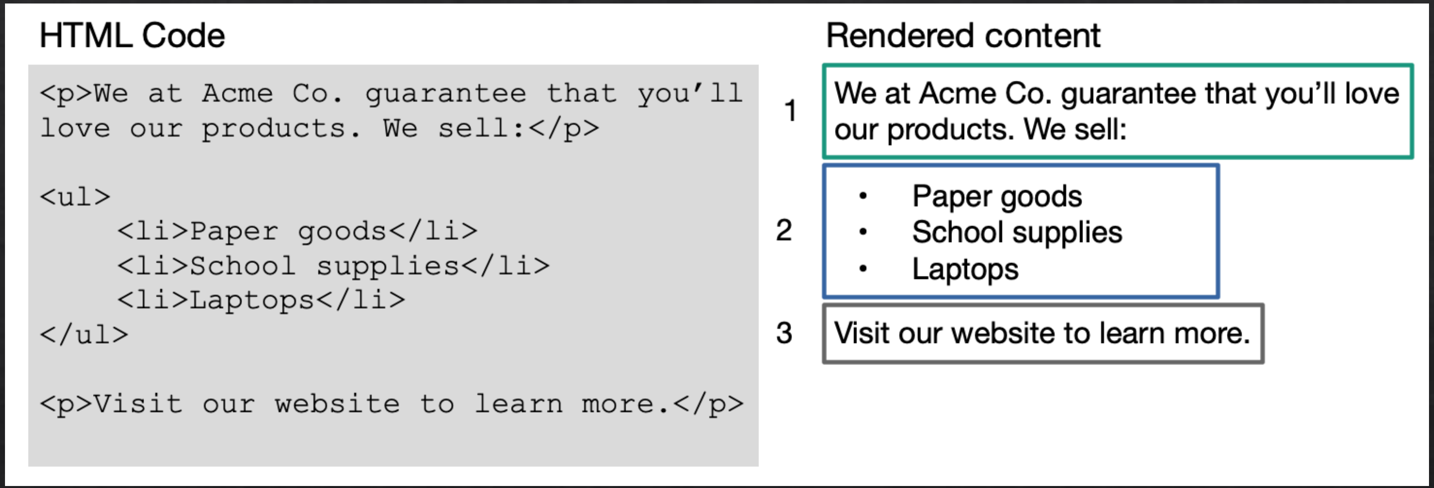
An attribute defines the properties of an HTML element. Here are some of the attribute examples from the listed code:
example: <meta name="viewport"… (name is an attribute for meta)

educative.io

Module 3
CSS Selectors

Box Model

Every element in web design is a rectangular box.
The content, padding, border, and margin can be used to calculate the amount of space that an element takes up.


HTML Code

```
<p>We at Acme Co. guarantee that you'll
love our products. We sell:</p>

<ul>
    <li>Paper goods</li>
    <li>School supplies</li>
    <li>Laptops</li>
</ul>

<p>Visit our website to learn more.</p>
```

Rendered content

1 We at Acme Co. guarantee that you'll love our products. We sell:

2
- Paper goods
- School supplies
- Laptops

3 Visit our website to learn more.

There are three types of "boxes" with different layout behaviors: block, inline, and inline-block. It's important to understand the default layout behavior of these boxes and how they interact

Module 3
CSS Selectors

Block

A block element always starts on a new line and takes up the full width available, meaning that it stretches out to the left and right as far as it can. If it's contained inside another element, it only takes up the width of that element, not the entire page. By default, block elements use as much height as the content needs, but it's possible to change the height and width.

Examples of block level elements include:

- <div>
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

Inline

An inline element doesn't start on a new line and only takes up as much height and width as necessary. It stays inline to the text that it's contained in. You can't set the height and width of an inline element.

Examples of inline elements include:

- <span>
- <a>
- <img>
- <em>
- <strong>

Inline-block

An inline-block element is very similar to inline, but you can set the height and width of it. This can be useful if you want to change the size of certain elements but don't want them to start on a new line.

Examples of inline-block elements include:

- <select>
- <button>

Module 3
CSS Selectors

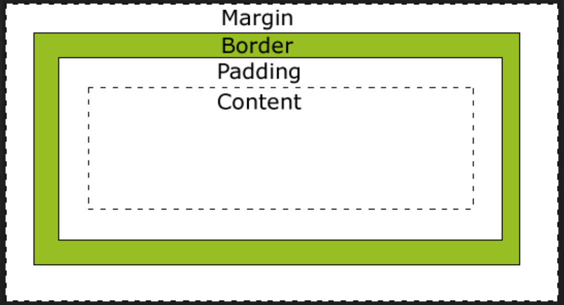
Every element in web design is a rectangular box.
The content, padding, border, and margin can be used to calculate the amount of space that an element takes up.

Calculate the total space taken up on the screen (in pixels)

CSS:

```
div {
    width: 200px;
    height:50px;
    border: 5px solid lightgreen;
    padding: 20px;
    margin: 12px 4px;
}
```

height width



Double the pixel counts for padding, margin, and border + width/height.

Total Space: Width: 258px; height: 124px

Module 3
CSS Selectors

Basic selectors

A CSS selector selects the element or elements that you want to style on the page. There are many ways to write selectors based on how broad or specific you want to be.

CSS is used to change the default styles of HTML elements in the browser. You can change element colors, borders, widths, heights, fonts, and display styles. To define these new rules, you first need to "select" the elements that you want to change.

Element selector

Class selector

ID selector

Module 3
CSS Selectors

Advanced selectors

Now that you've seen basic selectors, there are a few more "advanced" selectors to know. These may not be used as much as the other selector types, but you might find yourself needing one of these selectors or come across it in an existing project.

- **Universal selector**
- **Attribute selector**
- **Pseudo-class selector**
- **Combo combinator**
- **Multiple combinator**
- **Child combinator**
- **Descendant combinator**

Module 3
CSS Selectors

CSS rules

Along with the selector, a CSS rule consists of at least one property and value. For instance, if you wanted to change the background color of all divs to red, you could write a rule like this:

```
div {
    background-color: red;
}
```

In this example, background-color is the property, and red is the value that's being set to that property. Note that the property and value are separated by a colon : and the CSS rule has a semi-colon ; at the end. The semi-colons are important when you have multiple property-value pairs.

If you wanted to also style those divs to be only 500 pixels wide, the CSS rule would look like this:

```
div {
    background-color: red;
    width: 500px;
}
```

You can find all the CSS properties that can be changed on MDN's [CSS Reference](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference page)

Module 3

CSS Grid

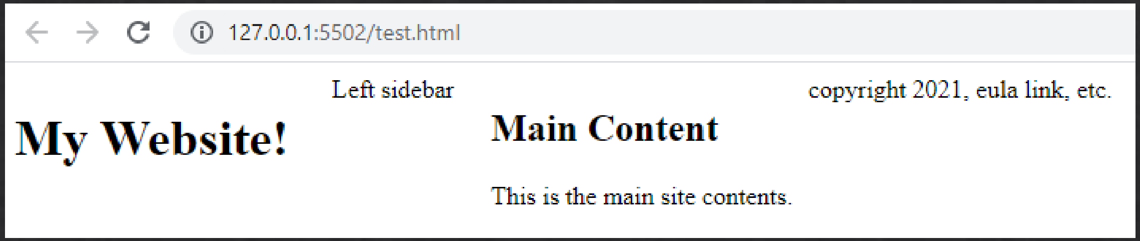
CSS Grid Layout, or Grid

Grid is a fairly recent addition to CSS, but it's already adopted by more than 95% of browsers in use today. Grid provides the ability to split an HTML page into rows and columns and then assign elements of that page into the "grid" that is created.

Grid is typically used to split up the major sections of a page—like a header, footer, nav, and main content—and then assign elements to those sections. The browser then calculates and figures out how everything fits and how big it all should be.

Module 3
CSS Grid

Defining the Grid


127.0.0.1:5502/test.html

My Website!

Left sidebar

Main Content

This is the main site contents.

copyright 2021, eula link, etc.

You can create this layout with grid. To define a grid, you'd use the grid value of the display property. Any element can be a grid container.

When you set display: grid, all of the direct children of the container become grid items:

CSS

```
body {
    display:grid;
    …
  }
```

The body is now a grid container, which by default gives you a one-column grid.

Module 3
CSS Grid

Grid Template Columns

To define the columns in your grid, you can use the property grid-template-columns. The value of this property is the number of columns you want to define and the size of the column.

CSS

```css
body {
    display: grid;
    grid-template-columns: 200px 100px 200px 300px;
  }
```

Grid Template Rows

To be more explicit with your row definitions in your grid, you can use the property grid-template-rows.

```css
.container {
    height: 500px;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
    grid-template-rows: 100px 1fr 100px;
    gap: 20px;
  }
```

Grid Gap

Grid gap
You might also want some space between the grid items. You can do this by adding padding or margins to your elements, but it's better to add some space between the columns and rows instead. You can do this by adding the gap property to your grid definition:

```css
body {
    display: grid;
    grid-template-columns: 1fr 1fr  1fr  1fr;
    gap: 40px;

  }
```

Module 3
CSS Grid

CSS Units: vh and vw

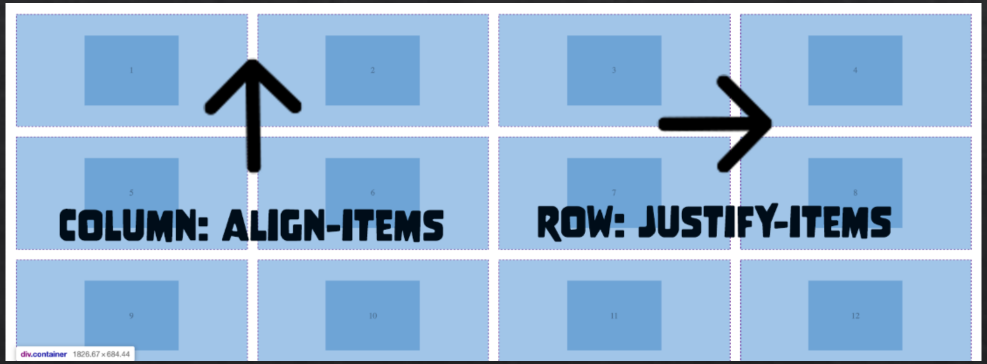
There are two CSS units of measure that allow you to stretch the entire height and width of the viewport

vh: Relative to 1% of the height of the viewport
vw: Relative to 1% of the width of the viewport

Module 3
CSS Grid

Box alignment in CSS Grid

When working with grid layouts, there are two axes you can align items against: the block axis (column) and the inline (row) axis.



These properties are defined on the grid container and define how all of the grid items are aligned:

- **align-items**: Aligns grid items along the block (column) axis
- **justify-items**: Aligns grid items along the inline (row) axis

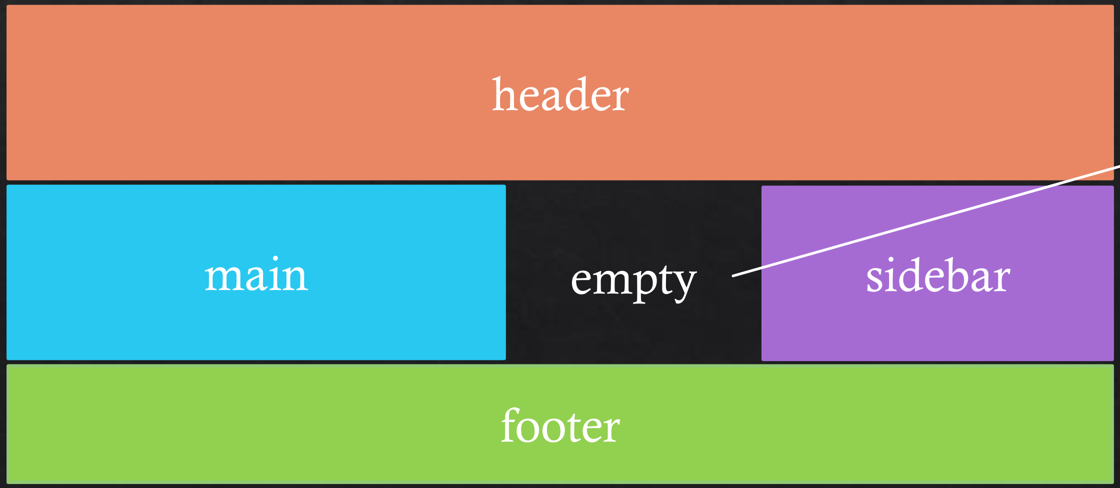
*

Module 3
CSS Grid

Grid Template Areas

There are times when you'll need a grid item to span multiple columns, rows, or both. To do this, you use the grid property called grid-templates-areas.

With grid template areas, you define a grid of rows and columns that your page should be split up into, and then assign elements from your HTML into those grid areas. Using the same markup from before, you can create a grid that looks like this:



empty is Indicated in CSS with a

. (period/dot)

*

Module 3
CSS Grid

Grid Template Areas

You define the template in CSS, defining grid areas and naming them so that you can insert your content into them. First, you'd define the grid-template-areas property and set it to a string template.

If you define the same name to each column in a row, that means that element spans that entire row. If you put a . in one of the areas, that means that the area is empty.

You then have to give these grid names to your elements that are in the body element:

```
body {
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
    grid-template-areas:
      "header header header header"
      "content content . sidebar"
      "footer footer footer footer";
  }

  main {
    grid-area: content;
  }

  header {
    grid-area: header;
  }

  footer {
    grid-area: footer;
  }

  aside {
    grid-area: sidebar;
  }
```

*

What’s Responsive Web Design?

Responsive web design is a set of practices that allows pages to alter their layout and appearance to suit different screen sizes and resolutions. The goal of responsive design is to create one site that works on every screen.

Designing a responsive interface
Applying a responsive design consists of addressing three key areas:

1. Flexible, or fluid, grid layouts
2. Resizable images
3. CSS media queries

Whether a visitor to your site uses a phone or a large screen desktop, your site must automatically switch to accommodate the screen's resolution and support larger image sizes. Even if the user doesn't switch devices, but changes the orientation of the screen from portrait to landscape, you want your design to accommodate the extra space and fill it with content accordingly.

Module 3
CSS Grid

Mobile First

With the shift in internet consumption coming from mobile devices, mobile screens are now the primary means in which users interact with your applications. As such, the term mobile-first was coined to indicate that applications should be developed with a "mobile-first mindset."

As developers, instead of adding breakpoints into the design as the width of the screen gets smaller, you should create breakpoints in the design when the width of the screen gets larger.

One approach that helps provide a mobile-first approach is to simulate mobile devices using Chrome or Firefox's Responsive Web Design tools.

A strategy to follow for introducing breakpoints is to start with the small screen first, then expand until it looks bad. At this point, it's time to insert a new breakpoint.

Module 3

Flexbox

What is Flexbox?

Flexbox is a layout option in CSS that provides a flexible way to lay out items on a page. It's been around for several years now and is supported in about 98% of users' browsers.

Flexbox dynamically handles alignment and the space between the items without much calculation on the developer's part. It consists of a flex container that contains one or more flex items.

Typically, you don't use Flexbox to lay out an entire page, but to lay out a group of like elements on a part of the page.

Module 3

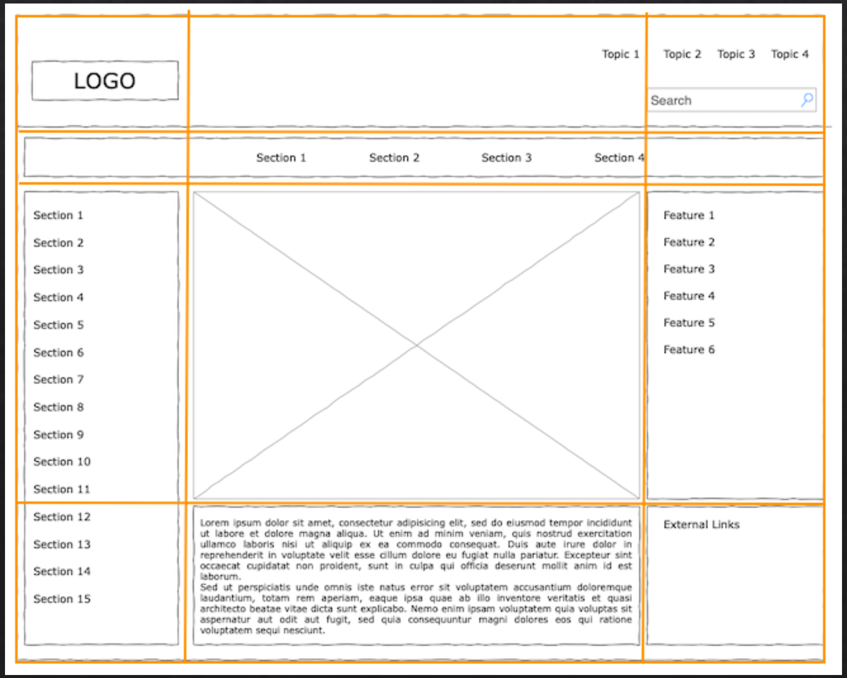
Flexbox

When to use Flexbox or Grid

Grid and Flexbox don't have to be used separately. They're often used together to solve problems related to their individual strengths.

Layout symmetry in both rows and columns. There is spacing between the rows and columns. These are strengths of the grid layout.

Grid is most appropriate for this design



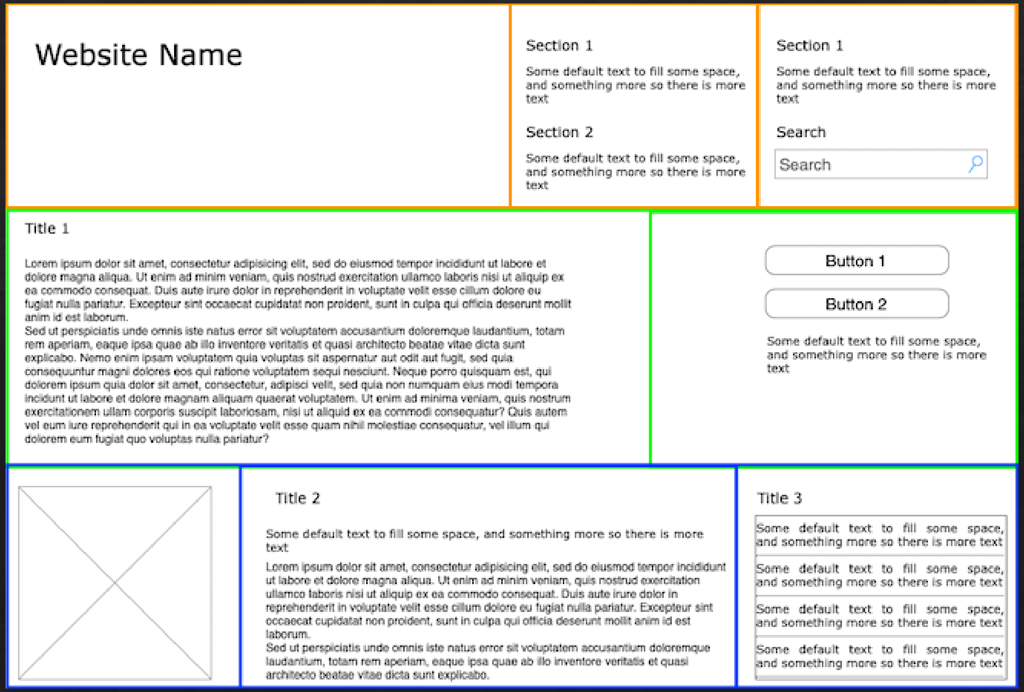
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

Module 3
Flexbox

When to use Flexbox or Grid

The first thing to notice in this image is the lack of symmetry. This page is seemingly made up of three rows that each have their own individual layout problems to solve. Arranging related content is the strength of Flexbox.

Flexbox is most appropriate for the rows in this design


Website Name
Section 1
Some default text to fill some space, and something more so there is more text
Section 2
Some default text to fill some space, and something more so there is more text
Section 1
Some default text to fill some space, and something more so there is more text
Search
Search
Title 1
Button 1
Button 2
Some default text to fill some space, and something more so there is more text
Title 2
Some default text to fill some space, and something more so there is more text
Title 3
Some default text to fill some space, and something more so there is more text
Some default text to fill some space, and something more so there is more text
Some default text to fill some space, and something more so there is more text
Some default text to fill some space, and something more so there is more text

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Module 3

Flexbox

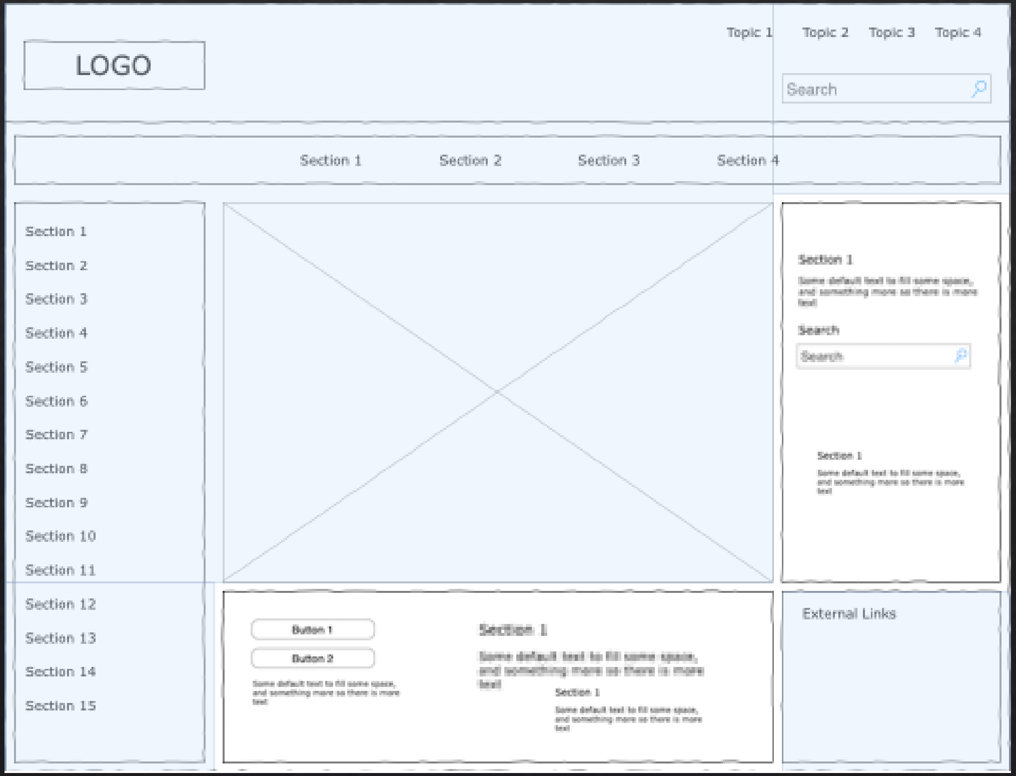
When to combine Flexbox and Grid

Grid and Flexbox don't have to be used separately. They're often used together to solve problems related to their individual strengths.

Grid is for building the layout of the page. Flexbox is for positioning elements relative to each other.

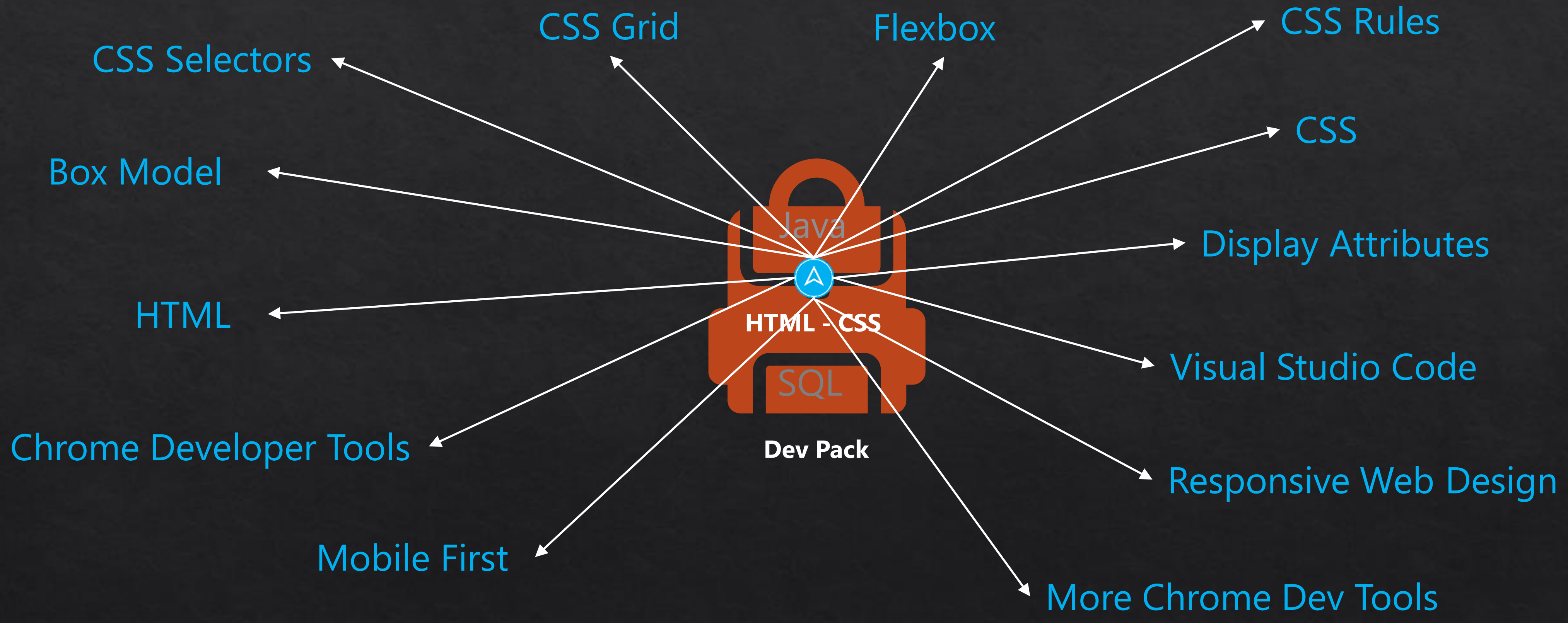
Grid is used for two-dimensional layouts, and Flexbox is used for one-dimensional layouts.

Using them together


LOGO
Topic 1
Topic 2
Topic 3
Topic 4
Search
Section 1
Section 2
Section 3
Section 4
Section 1
Section 2
Section 3
Section 4
Section 5
Section 6
Section 7
Section 8
Section 9
Section 10
Section 11
Section 12
Section 13
Section 14
Section 15
Section 1
Some default text to fill some space, and something more so there is more text
Search
Search
Section 1
Some default text to fill some space, and something more so there is more text
Button 1
Button 2
Some default text to fill some space, and something more so there is more text
Section 1
Some default text to fill some space, and something more so there is more text
Section 1
Some default text to fill some space, and something more so there is more text
External Links

Module 3
CSS Grid

New Tools!



Week 1 - Review