

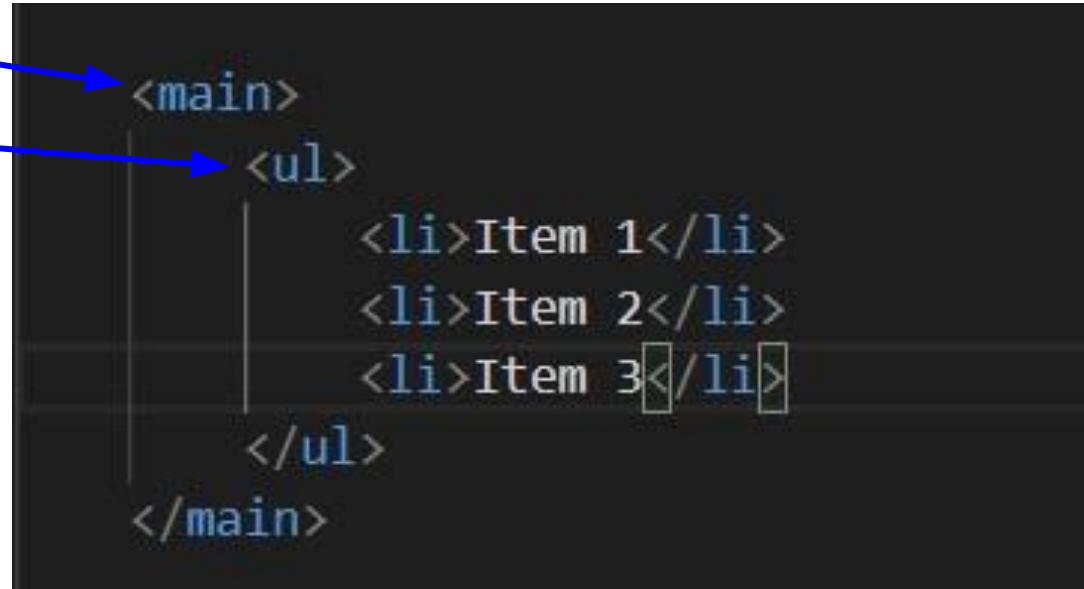
CSS SELECTORS & LAYOUT POSITIONING

HTML HIERARCHY

Parent

Child

```
<main>
|
|   <ul>
|   |
|   |   <li>Item 1</li>
|   |   <li>Item 2</li>
|   |   <li>Item 3</li>
|   |
|   </ul>
|
</main>
```

A diagram illustrating HTML hierarchy. On the left, two boxes labeled 'Parent' and 'Child' have blue arrows pointing to the corresponding HTML tags in a code block on the right. The 'Parent' box points to the opening <main> tag, and the 'Child' box points to the opening tag. The code block shows a nested structure: <main> contains , which contains three elements (Item 1, Item 2, Item 3). The closing tags and </main> are also shown.

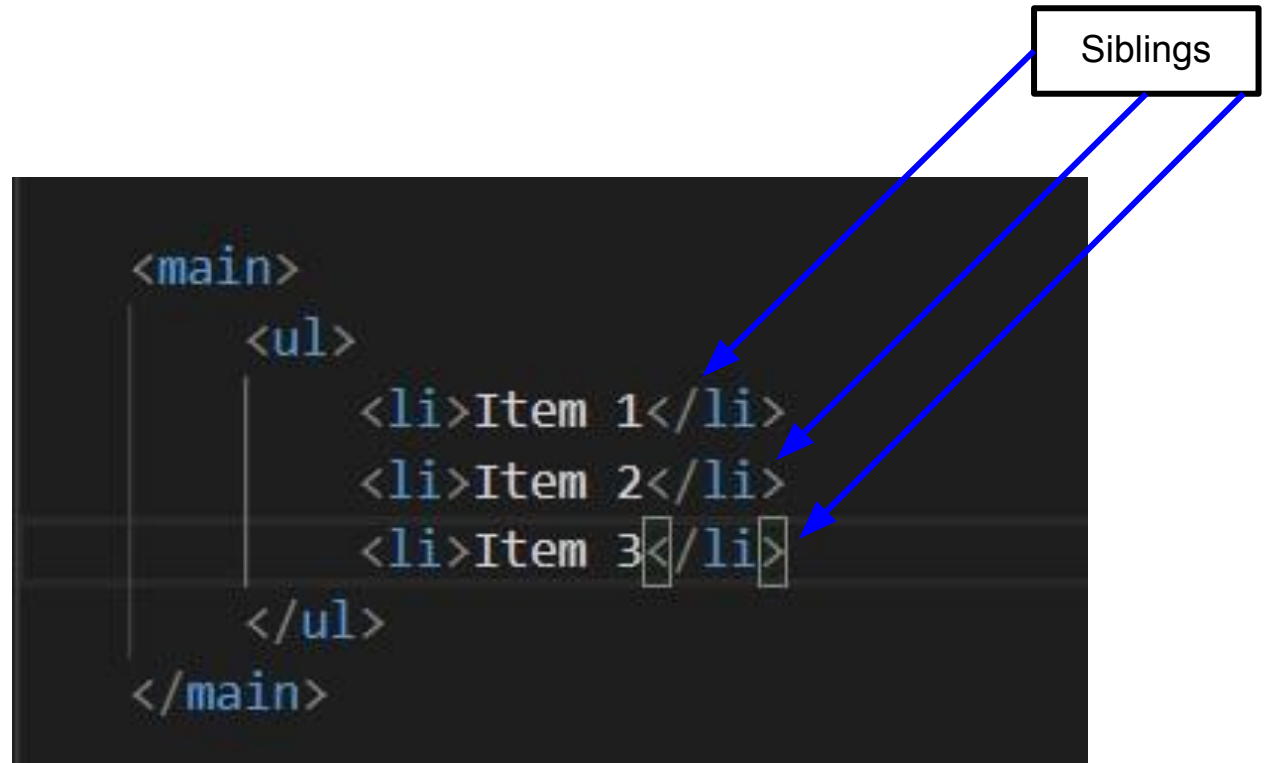
HTML HIERARCHY

Parent

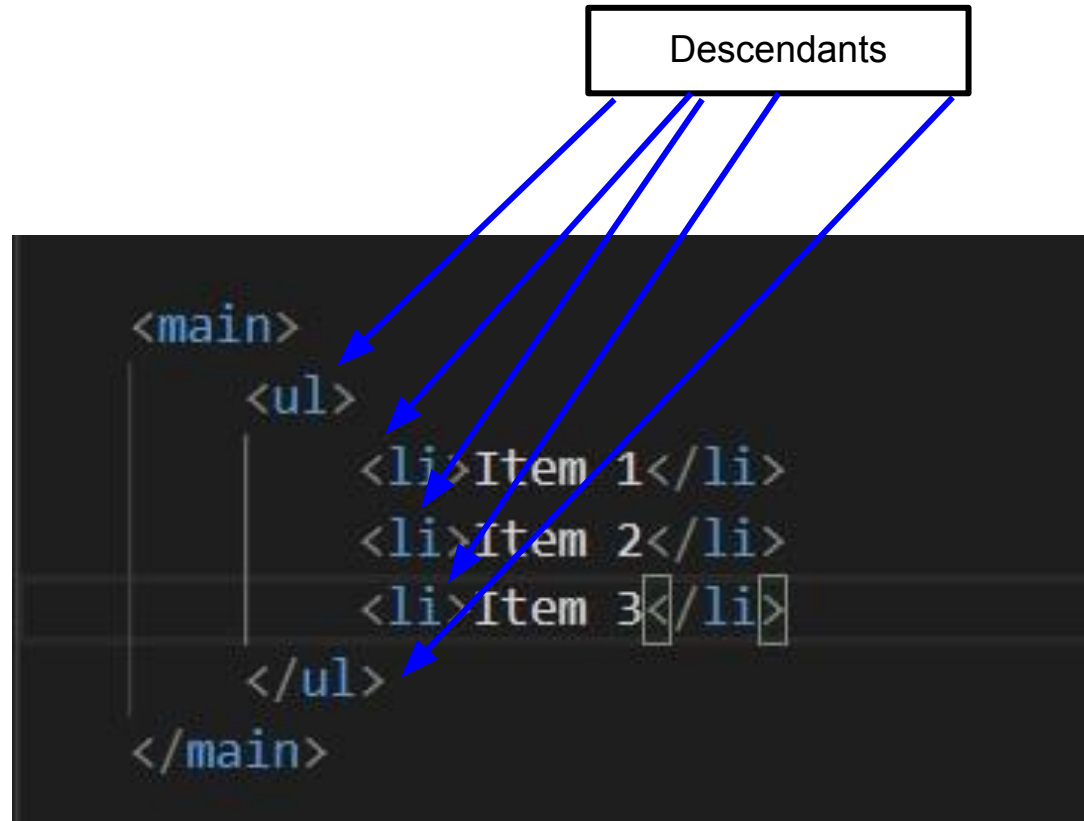
Child

```
<main>
|
|  <ul>
|  |
|  |  <li>Item 1</li>
|  |  <li>Item 2</li>
|  |  <li>Item 3</li>
|  |
|  </ul>
|
</main>
```

HTML HIERARCHY



HTML HIERARCHY



STYLING WITH CSS

CSS can be added in several ways:

- Inline with the <style> element

```
<h1 style="color: darkred">My Heading</h1>
```

- Within the document in a <style> element

```
<head>  
  <style>  
    h1 {  
      color: darkred;  
    }  
  </style>  
</head>
```

STYLING WITH CSS

CSS can be added in several ways:

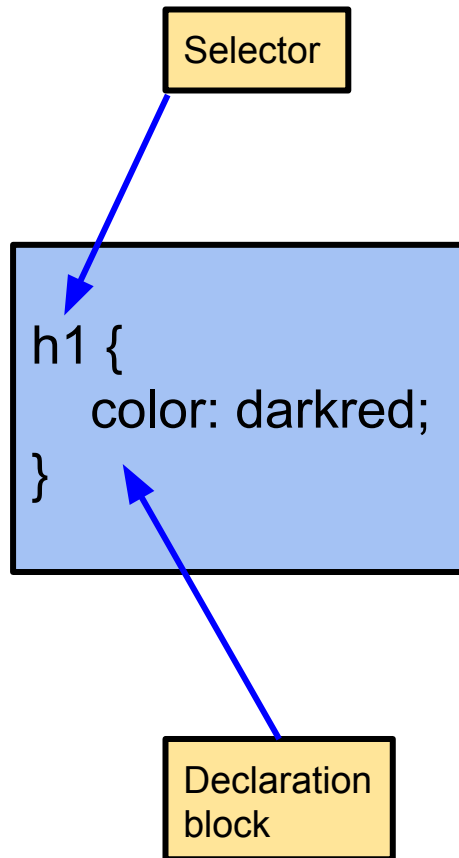
- Through an external document reference with a `<link href>` element

```
<head>
  <link
    rel="stylesheet" href="style.css"
  />
</head>
```

CSS RULE

CSS rules are made up of:

- Selector
 - Indicates which elements to affect
- Declaration block
 - Specifies styling for affected elements



SELECTOR TYPES

There are several main types of selectors:

- **Universal**

- Specified by *
- Selects all elements

- **Element**

- Specified by element tag (i.e. `h1`)
- Selects all elements of the specified type

- **Id**

- Specified by id of element preceded by # (i.e. `#mybox`)
- Selects the element with the specified id

- **Class**

- Specified by class preceded by . (i.e. `.redbox`)
- Selects all elements with the specified class

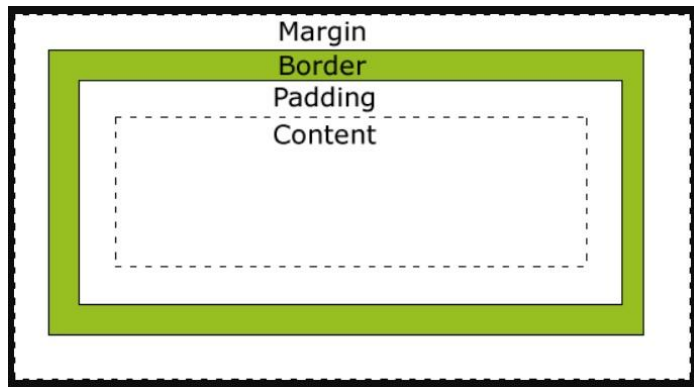
ADVANCED SELECTOR TYPES

- **Combinations**
 - `div.warning` (`div` elements with `warning` class)
- **Multiple Selector (comma)**
 - `h1, h2` (`h1` elements and `h2` elements)
- **Descendant Selector (space)**
 - `body h1` (`h1` descendants of `body`)
- **Child selector (>)**
 - `body > header` (`header` children of `body`)
- **Attribute Selector**
 - `input[type=number]` (`input` elements whose `type` attribute has a value of `number`)
- **Pseudo-class**
 - `a:hover` (`a` elements in `hover` state)

THE BOX MODEL

BOX MODEL

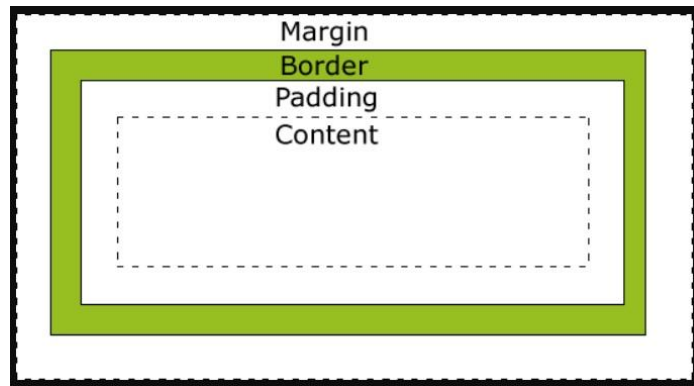
- Every **element** in web design is a **rectangular box**.
- The **content, padding, border, and margin** can be used to calculate the amount of space that an element takes up.
- Margin is the space **outside an element**. It does not affect the size of the box but affects other content that interacts with the box.
- Padding is the space **inside an element**.



BOX MODEL

The **box-sizing** css property allows you to change how the width and height of an element are calculated. The possible values for this property are:

- **content-box** (default) includes only size of element in size calculations.
- **border-box** includes content, padding, and border of element but do not include the margin, when calculating size.



DISPLAY PROPERTY

Every HTML element has a default display value depending on what type of element it is. The default `display` value for most elements is **block** or **inline**.

- A **block level** element always starts on a **new line** and takes up the **full width** available—meaning it stretches out to the left and right as far as it can.
 - Examples:
 - `<div>`
 - `<h1>` - `<h6>`
 - `<p>`
 - `<form>`
 - `<header>`
 - `<footer>`
 - `<section>`

DISPLAY PROPERTY

- An **inline element** does not start on a new line and only takes up **as much width as necessary**.
 - Examples:
 - ``
 - `<a>`
 - ``
- **inline** is the default value for elements like `` and ``.
 - It **accepts margin and padding** while ignoring height and width.
- **inline-block** is very similar to inline, but **it allows you to set a width and height**.
- **display:none** can be used to hide an element

POSITIONING

- The **normal flow** of a page is for elements to appear left to right and top to bottom based on the order in which they appear in the HTML document and the rules of block and inline display.
- **Static** position by default means the element **conforms to normal flow**.
- **Relative** position means **relative to where it would otherwise be positioned in the normal flow**
 - You can set the *top*, *right*, *bottom*, and *left* positioning attributes.

POSITIONING

- **Absolute** position places the element relative to the parent ancestor—that is, the containing element—**exactly where you specify**.
 - These elements are removed from the flow of the page.
 - Setting both *top* and *bottom*, or both *left* and *right* allows you to "stretch" an element's dimensions.
- **Fixed** position is **relative to the browser window** and does not scroll with the page.
 - You can set the *top*, *right*, *bottom*, and *left* positioning attributes.

UNITS OF MEASUREMENT

- Pixels
 - i.e. `40px`;
- Other absolute units:
 - i.e. `in`, `cm`, `mm`, `pt`, `pc`
- Relative to the current font
 - i.e. `2.5em`;
- Relative to the Root Element font
 - i.e. `1.8rem`;
- Percentage relative to parent
 - i.e. `width: 50%`