week3_welcome.css

week3_intro.css

Frontend Development

[Using html, css, js & jsx] {

```
console.log("<p>Welcome to <b>Frontend track</b></p>")
```

}

RastaXarm: +2348141161177

week3_intro.css

week3_introduction.css

WEEK_03 {

[Introducing CSS]

```
func1("Box Model", ()⇒{
    func2("Selectors", ()⇒{
        func3("Variables", ()⇒{
          console.log("Measurements");
        })
    })
```

}

Introduction

Websites developed with HTML5 alone can be functional, but they lack this important element of visual appeal. To improve the appearance of a website by including color, formatting text, and adding margins, borders, and shadows, for example, you need to apply styles created with Cascading Style Sheets (CSS).

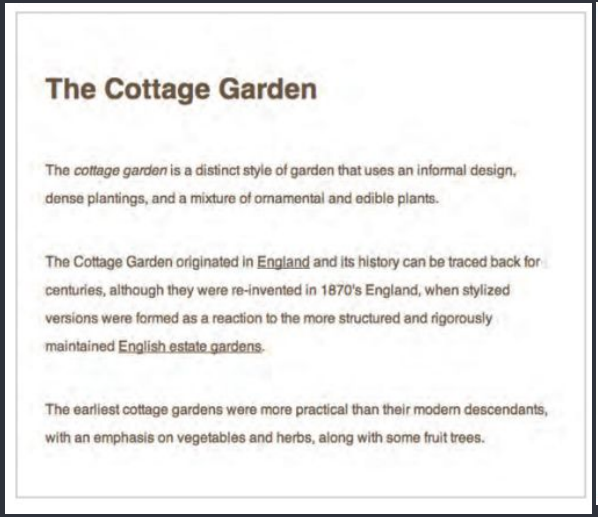
CSS allows you to create rules that specify how the content of an element should appear. For example, you can specify that the background of the page is cream, all paragraphs should appear in gray using the Arial typeface, or that all level one headings should be in a blue, italic, Times typeface.

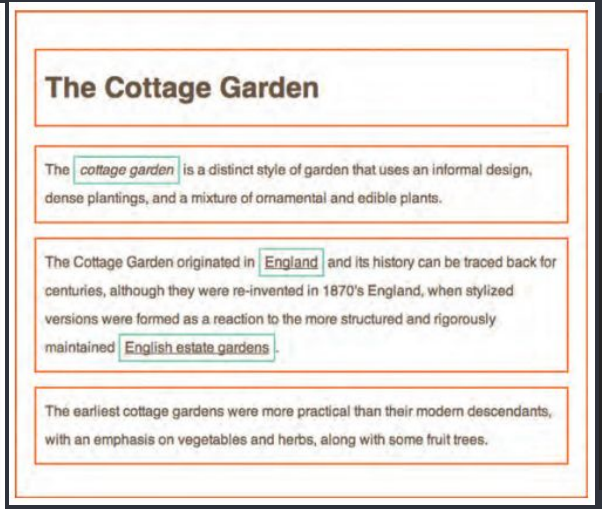
week3_css_box_model.css

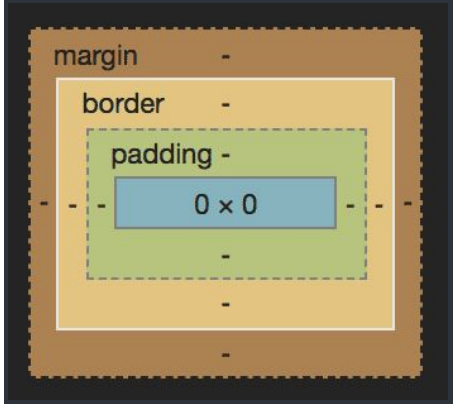
week3_using_css_in_html_inline.css

#css{c:r;}: Box Model

The CSS Box Model is a box that wraps around every HTML element. This box model consists of content, paddings, borders and margins. All the HTML elements are treated as boxes in CSS.







#css{c:r;}: CSS In HTML → Inline

CSS can be added to HTML documents in 3 ways: Inline, Internal and External

Inline CSS is used to apply a unique style to a single HTML element.

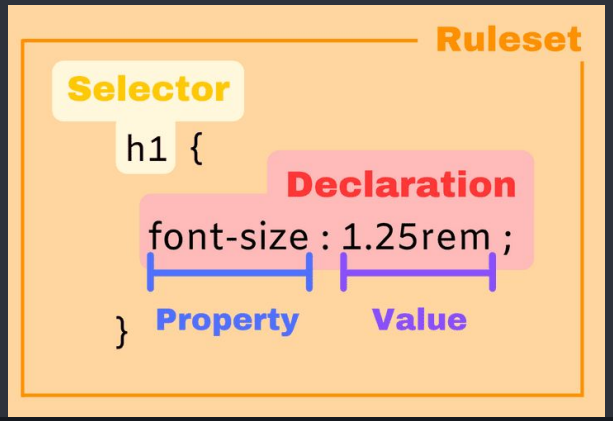
To add inline CSS, you use a style attribute and place it inside the opening tag of an HTML element.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Document</title>
  </head>
  <body>
    <h1 style="color: red">Intro to CSS</h1>
    <p style="background-color: blue">
      Lorem ipsum dolor sit amet consectetur.
    </p>
    <a class="active" href="index.html">Home</a>
  </body>
</html>
```

#css{c:r;}: CSS In HTML → Embedded

Embedded or Internal CSS is used to define a style for a single HTML page.

Internal CSS is defined in the <head> section of an HTML page, within a <style> element using CSS Ruleset.




```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <style>
      a.active {
        color: red;
        background-color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Intro to CSS</h1>
    <p>Lorem ipsum dolor sit amet consectetur.</p>
    <a class="active" href="index.html">Home</a>
  </body>
</html>
```

#css{c:r;}: CSS In HTML → External

External CSS is used to define the style for many HTML pages. This is the most efficient way of using CSS in HTML. It requires a separate document with a .css extension containing all CSS ruleset without HTML tags.

To use an external style sheet, add a <link> in the <head> to tell the browser where to find the CSS file used to style the page.

- href: This specifies the path to the CSS file.
- rel: This specifies the relationship between the HTML page and the file it is linked to.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>Intro to CSS</h1>
    <p>Lorem ipsum dolor sit amet consectetur.</p>
  </body>
</html>
```

```css
#selector1{
    color: red;
    background-color: blue;
    font-size: 1.5rem;
}
.selector2{
    color: green;
    background-color: yellow;
    border: 2px solid red;
}
```

#css{c:r;}: CSS Selectors

CSS selectors are used to "find" or “select” the HTML elements you want to style (i.e CSS selectors define the elements to which a set of CSS rules apply) and Selectors are the part of CSS ruleset.

CSS selectors is divided into five categories:

- Basic selectors: select elements based on name/type, id and class
- Attribute selectors: select elements based on an attribute or attribute value.
- Pseudo-class selectors: select elements based on a certain state.
- Pseudo-elements selectors: select and style a part of an element.
- Combinatory selectors: select elements based on a specific relationship between them.

#css{c:r;}: Basic Selectors

Basic

Name	CSS	Description	Results
Universal Selector	*	Select all elements	a b c d
Type Selector	div	Select elements of that type Select div elements	a div c div
Class Selector	.c	Select elements with that class Select elements with the c class	.a .b .c .d
Id Selector	#i	Select elements with that id Select elements with the i id *It is best practice to not use ids in CSS	#a #b #i #d

#css{c:r;}: Attributes Selectors

Attribute

Name	CSS	Description	Results
Has Attribute	[a]	Select elements that have that attribute Select elements with the a attribute	[a] [a="1"] [c] d
Exact Attribute	[a="1"]	Select elements that have that attribute with exactly that value Select elements with the a attribute with a value of 1	[a] [a="1"] [c] d
Begins With Attribute	[a^="1"]	Select elements that have that attribute which start with that value Select elements with the a attribute with a value that starts with 1	[a="12"] [a="21"]
Ends With Attribute	[a$="1"]	Select elements that have that attribute which end with that value Select elements with the a attribute with a value that ends with 1	[a="12"] [a="21"]

#css{c:r;}: Pseudo Element Selectors

Ends With Attribute	[a$="1"]	Select elements that have that attribute which end with that value Select elements with the a attribute with a value that ends with 1	[a="12"] [a="21"]
Substring Attribute	[a*="1"]	Select elements that have that attribute which contain that value anywhere Select elements with the a attribute with a value that contains a 1	[a="12"] [a="21"]

Pseudo Element

Name	CSS	Description	Results
Before Selector	div::before	Creates an empty element directly before the children of selected element	div before c after
After Selector	div::after	Creates an empty element directly after the children of selected element	div before c after

#css{c:r;}: Pseudo Class Selectors

Pseudo Class State

Name	CSS	Description
Hover Selector	button:hover	Select elements that are hovered by the mouse Select buttons that are being hovered
Focus Selector	button:focus	Select elements that are focused. Select buttons that are being focused *Focus is set by either tabbing to an element or clicking an element such as a button or anchor tag
Required Selector	input:required	Select inputs that are required Select inputs with the required attribute
Checked Selector	input:checked	Select checkboxes/radio buttons that are checked Select inputs that are checked
Disabled Selector	input:disabled	Select inputs that are disabled Select inputs with the disabled attribute

week3_css_selectors_pseudo_class.css

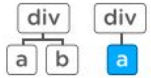
#css{c:r;}: Pseudo Class Selectors

Pseudo Class Position/Other

Name	CSS	Description	Results
First Child Selector	a:first-child	Select elements that are the first child inside a container Select anchors that are the first child	div: a b; div: b a
Last Child Selector	a:last-child	Select elements that are the last child inside a container Select anchors that are the last child	div: a b; div: b a
Nth Child Selector	a:nth-child(2n)	Select elements that are the nth child inside a container based on the formula Select anchors that are even numbered children	div: a a b a
Nth Last Child Selector	a:nth-last-child(3)	Select elements that are the nth child inside a container based on the formula counting from the end Select anchors that are the third to last child	div: a a b a

week3_css_selectors_pseudo_class.css

#css{c:r;}: Pseudo Class Selectors

Nth Last Child Selector	a:nth-last-child(3)	nth child inside a container based on the formula counting from the end Select anchors that are the third to last child	
Only Child Selector	a:only-child	Select elements that are the only child inside a container Select anchors that are the only child	
First Of Type Selector	a:first-of-type	Select elements that are the first of a type inside a container Select the first anchor in a container	
Last Of Type Selector	a:last-of-type	Select elements that are the last of a type inside a container Select the last anchor in a container	
Nth Of Type Selector	a:nth-of-type(2n)	Select elements that are the nth of a type inside a container based on the formula Select every second anchor	
Nth Last Of Type Selector	a:nth-last-of-type(2)	Select elements that are the nth of a type inside a container based on the formula counting from the end Select the second to last anchor	

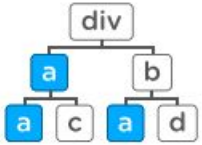
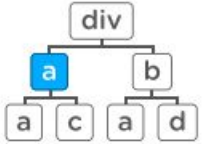
#css{c:r;}: Pseudo Class Selectors

Last Of Type Selector	a:last-of-type	Select elements that are the last of a type inside a container Select the last anchor in a container	div: b a a b
Nth Of Type Selector	a:nth-of-type(2n)	Select elements that are the nth of a type inside a container based on the formula Select every second anchor	div: b a a b
Nth Last Of Type Selector	a:nth-last-of-type(2)	Select elements that are the nth of a type inside a container based on the formula counting from the end Select the second to last anchor	div: b a a b
Only Of Type Selector	a:only-of-type	Select elements that are the only of a type inside a container Select anchors that are the only anchor in a container	div: b b a b
Not Selector	a:not(.c)	Select all elements that do not match the selector inside the not selector Select all anchor tags that do not have the c class	b a.c a a.d

week3_css_selectors_combinatory.css

#css{c:r;}: Combinatory Selectors

Combination

Name	CSS	Description	Results
Descendant Selector	div a	Select elements that are descendants of the first element Select anchors that are inside a div	
Direct Child Selector	div > a	Select elements that are direct children of the first element Select anchors that are direct children of a div	
General Sibling Selector	div ~ a	Select elements that are siblings of the first element and come after the first element Selects all anchors that are siblings of a div and come after the div	

#css{c:r;}: Combinatory Selectors

General Sibling Selector	div ~ a	Select elements that are siblings of the first element and come after the first element Selects all anchors that are siblings of a div and come after the div	a div a a
Adjacent Sibling Selector	div + a	Select elements that are siblings of the first element and come **directly** after the first element Selects all anchors that are siblings of a div and come **directly** after the div	a div a a
Or Selector	div, a	Select elements that match any selector in the list Selects all anchors and all divs	a div a b
And Selector	div.c	Select elements that match all the selector combinations Selects all divs with the class c	.a div.c .c div

week3_css_comments.css

week3_css_variables.css

#css{c:r;}: CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

To comment in CSS, simply place your plain text inside /* */ marks.

Comments in CSS are ignored by the browser and have no effect on how styles are rendered on the front end.

```
/* below are style rules for all p tags in the document */
p {
  color: white; /* set the text color to white */
  background-color: #FF5C35; /* set the background color to orange */
  padding: 10px; /* set the padding to 10 pixels */
}
```

week3_css_variables.css

week3_css_measurements.css

#css{c:r;}: CSS Variables

CSS variables are custom names that you can create to represent values and reuse throughout in your css.

To declare a variable in CSS, come up with a name for the variable, then append two hyphens "--" as the prefix. And You can access the variable value using var() function in CSS.

CSS variables can have a global or local scope. Global variables can be accessed in any selectors, while local variables can be used only inside the selector where it is declared.

```
:root {
  --primary-color: #f28c;
  --fs: 17px;
  --mx: auto;
}

div {
  --bg: #f8b3d3cc;
  color: var(--primary-color);
  margin-inline: var(--mx);
  background-color: var(--bg);
}
```

#css{c:r;}: CSS Measurements

Many CSS properties like width, margin, padding, and font-size take a length, and CSS has many different ways to express length.

In CSS, length is a number followed by a unit. For example, 5px, 0.9em, 3fr, 50% and so on.

There are two general kinds of units used for length and size in CSS: absolute and relative.

Absolute Units

Absolute length units are fixed and a length expressed in any of these will appear as exactly that size. E.g: px, in, ch, pt, mm, cm etc.

#css{c:r;}: CSS Measurements

Relative Units

Relative length units are relative to another relative to another length property or settings.

- em Relative to the font-size of the element
- rem Relative to font-size of the root element
- % Relative to the parent element
- vw Relative to 1% of the width of the viewport
- vh Relative to 1% of the height of the viewport
- vmin Relative to 1% of viewport smaller dimension
- vmax Relative to 1% of viewport larger dimension
- fr Fractional Unit

RastaXarm: +2348141161177

Thank {
You;
}