



Internet Technologies

CSS Revision

Kavindu Samarasinghe

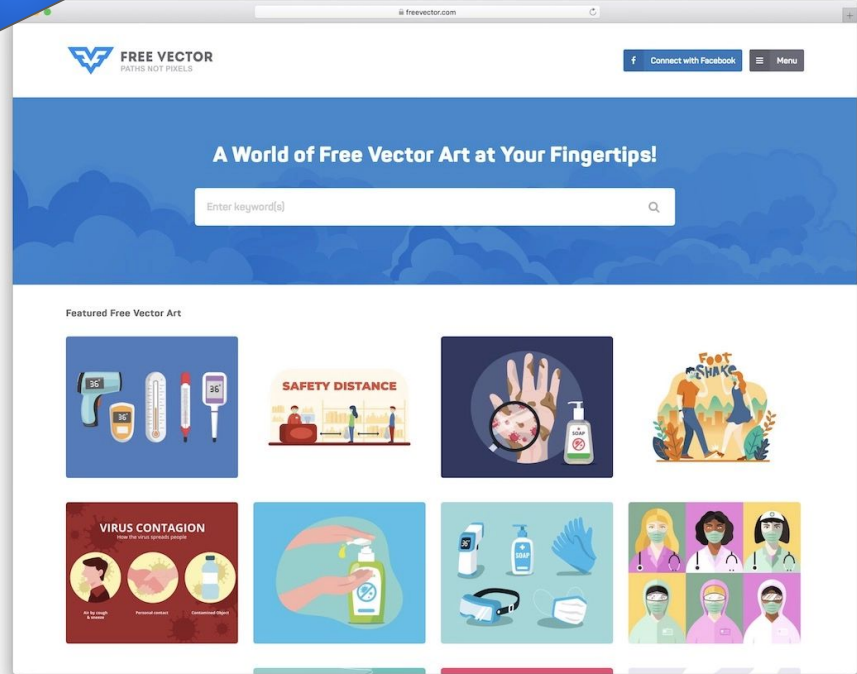
B.Sc (Hons) in Computer Science (1st class), Graduate Dip in Software Engineering,
Certificate in Digital Marketing (APIDM)

Software Applications

Desktop App



Web App



Mobile App



**FRONT
END**



VS

**BACK
END**



FULL STACK DEVELOPMENT



Can we develop websites drag and drop way?

Yes! but you will limit with set of limitations if you learn that way only.



wix



SQUARESPACE

Dw

As such learn core technologies and become unlimited powerful developer.

HTML CSS JavaScript



HTML



HTML + CSS



HTML + CSS + JavaScript

What is CSS?

What is CSS?

CSS stands for Cascading Style Sheets, and it is a stylesheet language used to **describe the presentation and styling** of HTML or XML documents.



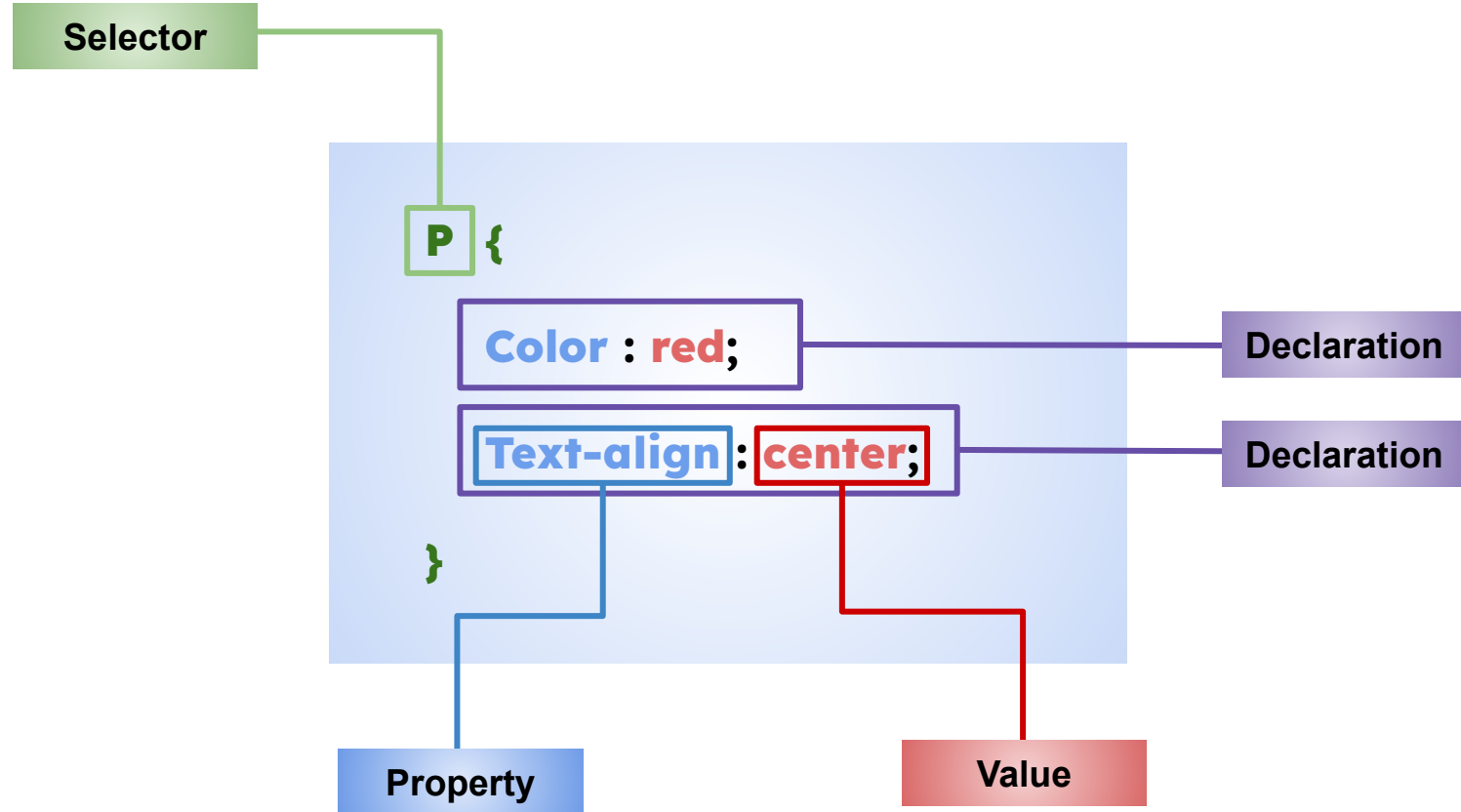
How to apply CSS to HTML?

Inline CSS- in an element

Internal CSS- inside the `<style>` tag in head

External Stylesheets - `<link rel="stylesheet" href="assets/styles/basic.css">` in head

CSS Rule



CSS Selectors

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

CSS Selectors

Type Selector

The CSS type selector **matches elements by node name**. In other words, it selects all elements of the given type within a document.

```
1 <html>
2   <head>
3     <style>
4       h1 {
5         color: red;
6       }
7     </style>
8   </head>
9   <body>
10    <h1>Hello CSS</h1>
11  </body>
12 </html>
```

CSS Selectors

ID Selector

The CSS ID selector **matches an element based on the value of the element's id attribute**. In order for the element to be selected, its id attribute must match exactly the value given in the selector

```
1 <html>
2   <head>
3     <style>
4       #myh1 {
5         color: red;
6       }
7     </style>
8   </head>
9   <body>
10    <h1 id="myh1">Hello CSS 1</h1>
11    <h1>Hello CSS 2</h1>
12  </body>
13 </html>
```

CSS Selectors

Class Selector

The CSS class selector **matches elements based on the contents of their class attribute.**

```
1 <html>
2   <head>
3     <style>
4
5       .main-headers {
6         color: red;
7       }
8
9     </style>
10  </head>
11  <body>
12
13    <h1 class="main-headers">Hello CSS 1</h1>
14    <h1>Hello CSS 2</h1>
15
16  </body>
17 </html>
18
```

CSS Selectors

Universal selector

The CSS universal selector (*) **matches**
elements of any type.

```
1 <html>
2   <head>
3     <style>
4       * {
5         color: red;
6       }
7     </style>
8   </head>
9   <body>
10    <h1>Hello</h1>
11    <h2>Hello</h1>
12    <h3>Hello</h1>
13  </body>
14 </html>
```

CSS Selectors

Attribute Selector

The CSS attribute selector **matches** elements based on the presence or value of a given attribute.

```
1 <html>
2 <head>
3   <style>
4     a[title] {
5       color: red;
6     }
7
8     a[title = "online shop - LK"] {
9       color: green;
10    }
11  }
12 </style>
13 </head>
14 <body>
15   <a href="http://www.ebay.com" title="online shop - USA">Visit ebay.com</a>
16   <a href="http://www.kapruka.com" title="online shop - LK">Visit kapruka.com</a>
17   <a href="http://www.amazon.com" title="online shop - USA">Visit amazon.com</a>
18 </body>
19 </html>
```

CSS Selectors: Attribute Selector Syntax

[attr]

Represents elements with an attribute name of attr.

[attr=value]

Represents elements with an attribute name of attr whose value is exactly value.

[attr~=value]

Represents elements with an attribute name of attr whose value is a whitespace-separated list of words, one of which is exactly value.

[attr|=value]

Represents elements with an attribute name of attr whose value can be exactly value or can begin with value immediately followed by a hyphen, - (U+002D). It is often used for language subcode matches.

[attr^=value]

Represents elements with an attribute name of attr whose value is prefixed (preceded) by value.

[attr\$=value]

Represents elements with an attribute name of attr whose value is suffixed (followed) by value.

CSS Selectors: Attribute Selector Syntax

[attr*=value]

Represents elements with an attribute name of attr whose value contains at least one occurrence of value within the string.

[attr operator value i]

Adding an i (or I) before the closing bracket causes the value to be compared case-insensitively (for characters within the ASCII range).

[attr operator value s]

Adding an s (or S) before the closing bracket causes the value to be compared case-sensitively (for characters within the ASCII range)

CSS Selectors

Pseudo-Class Selectors (·)

A CSS pseudo-class is a keyword added to a selector that **specifies a special state of the selected element(s)**.

```
1 <html>
2   <head>
3     <style>
4       h1 {
5         color: blue;
6       }
7
8       h1:hover {
9         color: green;
10      }
11    </style>
12  </head>
13  <body>
14    <h1>Hello CSS</h1>
15  </body>
16 </html>
```

CSS Selectors

Pseudo-Element Selectors (::)

A CSS pseudo-element is a keyword added to a selector that lets you **style a specific part of the selected element(s)**.

```
1 <html>
2   <head>
3     <style>
4       p::first-line {
5         color: red;
6       }
7     </style>
8   </head>
9   <body>
10    <p>
11      It is a long established fact that a reader
12      will be distracted by the readable content of
13      a page when looking at its layout.
14    </p>
15  </body>
16 </html>
```

CSS Selectors

Grouping Selector

The CSS grouping selector is used to **select multiple elements and style them together.**

```
1 <html>
2   <head>
3     <style>
4       h1, h2 {
5         color: red;
6       }
7     </style>
8   </head>
9   <body>
10    <h1>Hello</h1>
11    <h2>Hello</h2>
12    <h3>Hello</h3>
13  </body>
14 </html>
```

CSS Combinators

CSS Combinators

We use CSS combinators to make relationships between the selectors.

There are 4 types of combinators,

- **Child Combinator** - check direct child (>)
- **Decendent Combinator** - select the child if exist anywhere inside the parent (" " space)
- **Adjacent Combinator** - select the element which is located first after given first element (+)
- **General Siblings Combinator** - select the element which is located after given first element (~)

Cascading - “C” in CSS

Cascading

Process of combining different stylesheets and resolving conflicts between different CSS rules and declarations, when more than one rule applies to a certain element.

There are 3 steps to resolve the cascading:



Cascading - “C” in CSS

Important



Specificity



Source order

- Inline Selector = 1000
- ID Selector = 100
- Class or Pseudo Class or Attribute Selector = 10
- Type or Pseudo Elements Selector = 1

CSS Box Modeling

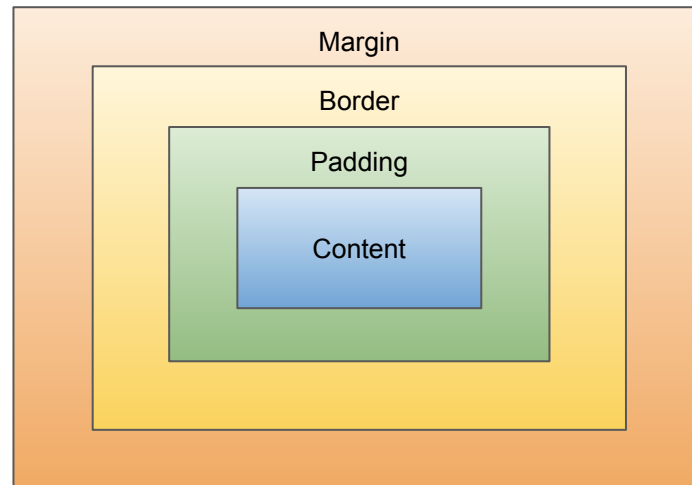
Box Modeling

The browser's rendering engine **represents each element as a rectangular box** according to the standard CSS basic box model.

CSS determines the size, position, and properties (color, background, border size, etc.) of these boxes.

🚀 Every HTML element has box model

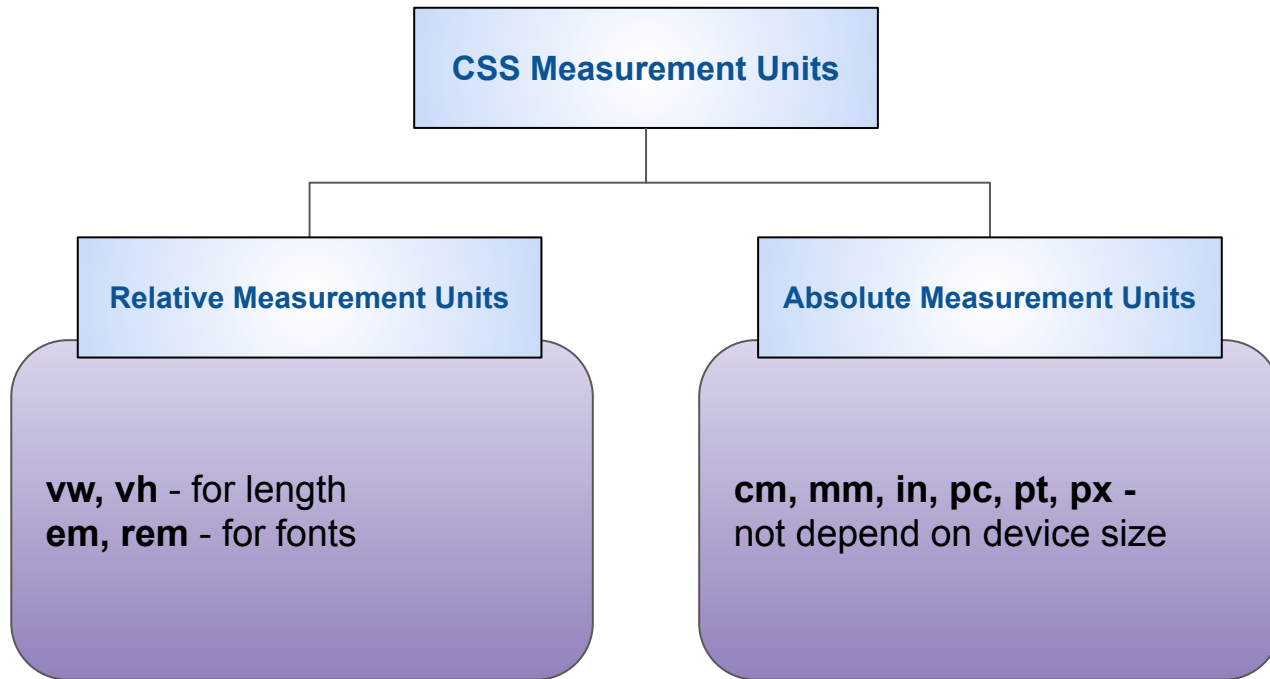
🚀 Some times markup tags has own default box model values which are applied by browsers



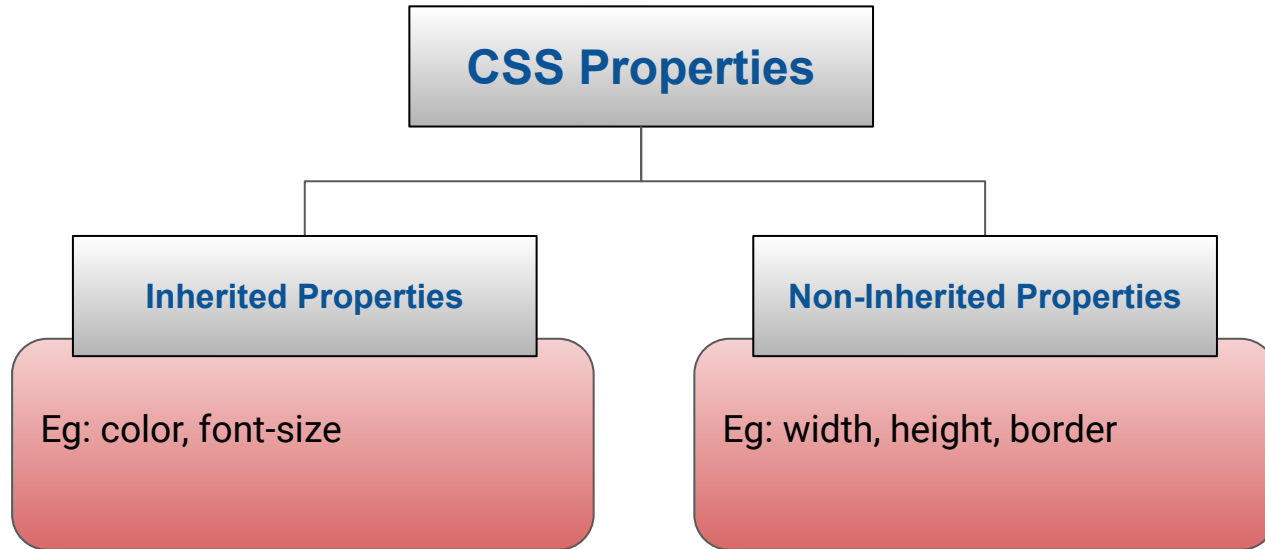
There are 4 types of combinators,

- 👉 **Content** - Content of the element
- 👉 **Padding** - The area which is between content and the border
- 👉 **Border** - Border of the element
- 👉 **Margin** - Margin of the element which is exist over the border

CSS Measurement Units



CSS Inheritance



CSS Box Types

block

- ◆ get 100% parent's width
- ◆ vertically
- ◆ box model applies
- ◆ `h1 {display: block}`

inline

- ◆ Only get content width
- ◆ horizontally - No line breaks
- ◆ No height and width (auto x auto)
- ◆ `h1 {display: inline}`

inline-block

- ◆ Only get content width
- ◆ horizontally - No line breaks
- ◆ box model applies
- ◆ `h1 {display: inline-block}`

CSS Positioning

CSS Positions

- static (default)

- relative

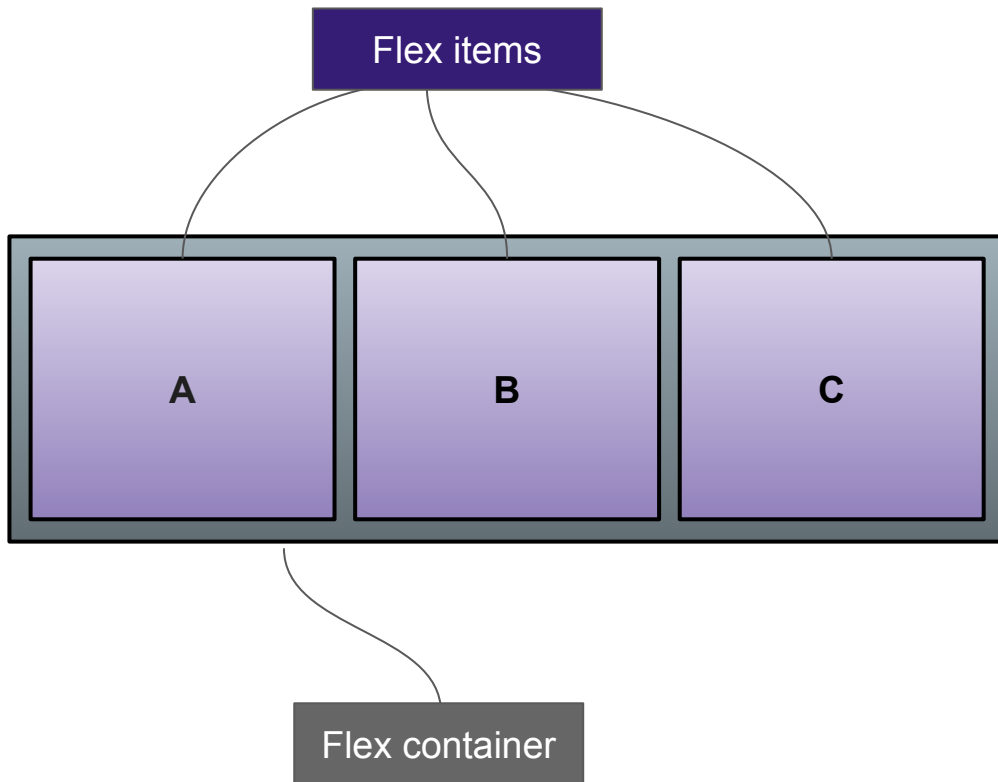
- absolute

- fixed

- sticky

CSS Flex-Box

```
1 <html>
2   <head>
3     <style>
4       section {
5         width: 400px;
6         height: 200px;
7         display: flex;
8         border: 5px solid green;
9       }
10    </style>
11    <body>
12      <section>
13        <div>A</div>
14        <div>B</div>
15        <div>C</div>
16      </section>
17    </body>
18  </html>
```

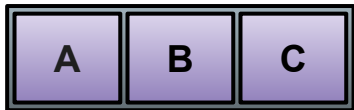


CSS Flex-Box

flex-direction

The flex-direction CSS property sets how flex items are placed in the flex container defining the main axis and the direction (normal or reversed).

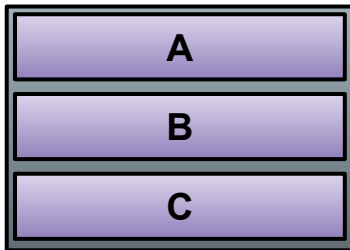
1. The direction text is laid out in a line
flex-direction: row;



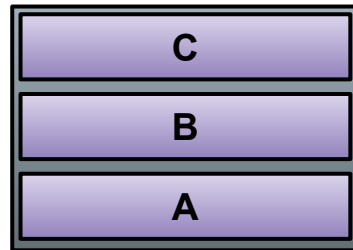
2. Like <row>, but reversed
flex-direction: row-reverse;



3. The direction in which lines of text are stacked
flex-direction: column;



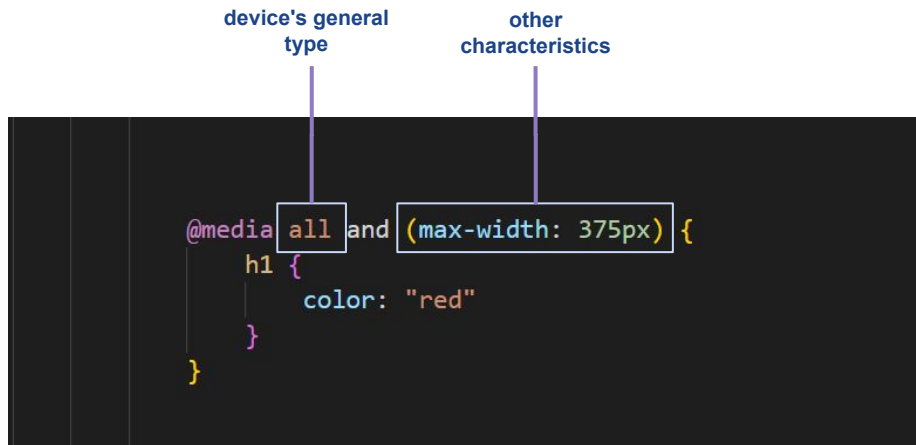
4. Like <column>, but reversed
flex-direction: column-reverse;



CSS Media Queries

Media Queries

Media queries allow you to apply CSS styles depending on a device's general type (such as print vs. screen) or other characteristics such as screen resolution or browser viewport width.



The diagram shows a CSS media query code snippet on a dark background. Two purple lines with labels point to parts of the code. The label 'device's general type' points to the word 'all' in the media query. The label 'other characteristics' points to the expression '(max-width: 375px)' in the media query. The code snippet is as follows:

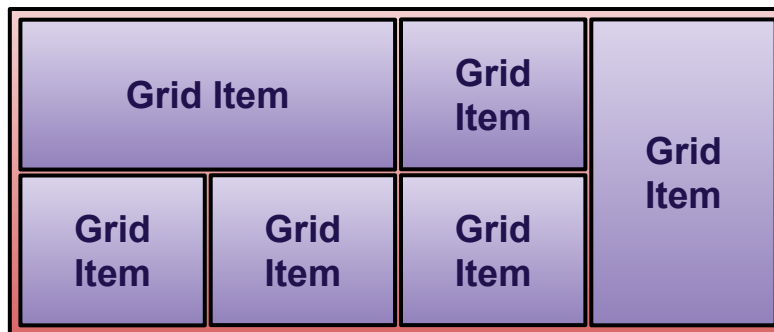
```
@media all and (max-width: 375px) {  
  h1 {  
    color: "red"  
  }  
}
```

CSS Grid Layout

Grid Layout

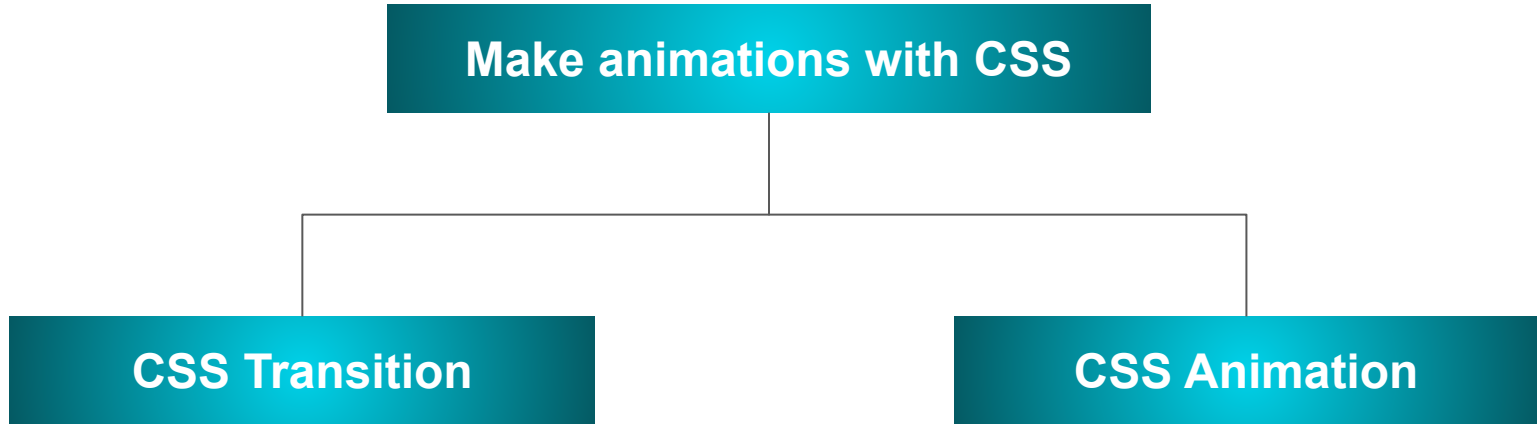
CSS Grid Layout excels at dividing a page into major regions or defining the relationship in terms of size, position, and layer, between parts of a control built from HTML primitives.

```
1 <html>
2 <head>
3   <style>
4     section {
5       width: 1000px;
6       height: 400px;
7       border: 5px solid green;
8       display: grid;
9       grid-template-columns: repeat(4, 1fr); /* fr - fractional */
10      grid-template-rows: repeat(2, 1fr); /* fr - fractional */
11    }
12    section > div {
13      border: 5px solid orange;
14    }
15    section > div:first-child {
16      grid-column: 1/3;
17    }
18    section > div:nth-child(3) {
19      grid-column: 4/5;
20      grid-row: 1/3;
21    }
22  </style>
23 </head>
24 <body>
25   <section>
26     <div></div>
27     <div></div>
28     <div></div>
29     <div></div>
30     <div></div>
31   </section>
32 </body>
33 </html>
```



Grid Wrapper

Make animations with CSS



Make animations with CSS

CSS TRANSITIONS	CSS ANIMATIONS
<ul style="list-style-type: none">• Can only move from initial to final state — no intermediate steps• Can only run once• Require a trigger to run (like mouse hover)• Run forwards when triggered and in reverse when trigger is removed• Easier to use with JavaScript• Best for creating a simple change from one state to another	<ul style="list-style-type: none">• Can move from initial to final state, with intermediate steps in between• Can loop infinitely thanks to animation-iteration-count property• Can be triggered but can also run automatically• Can run forwards, in reverse, or alternate directions• More difficult to use with JavaScript• Best for creating a complex series of movements



Kavindu Samarasinghe

B.Sc (Hons) in Computer Science (1st class), Graduate Dip in Software Engineering,
Certificate in Digital Marketing (APIDM)