IJSE - INSTITUTE OF SOFTWARE ENGINEERING

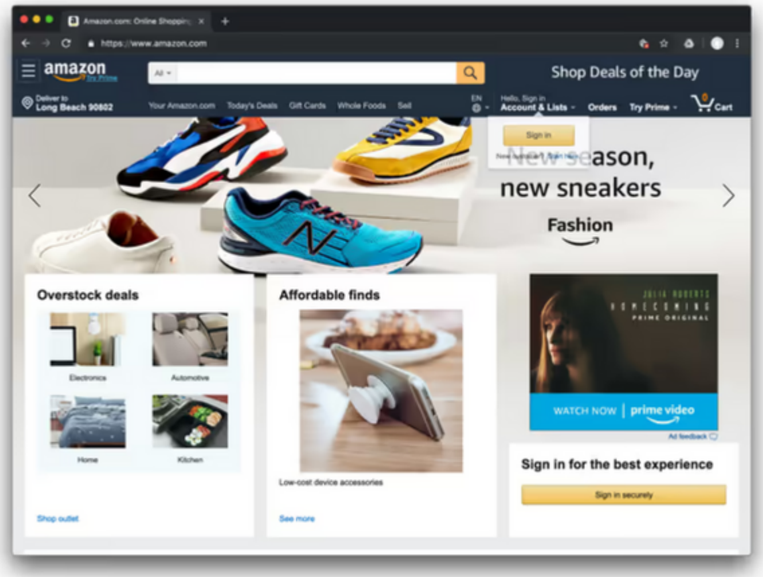
# CSS

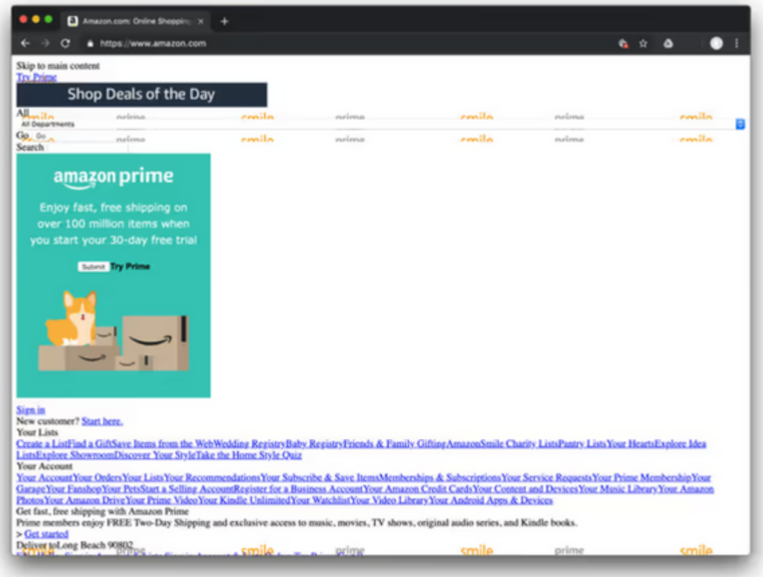
- INTRODUCTION TO CSS
- CSS COMBINATORS
- CSS SELECTORS
- CSS CASCADE
- SPECIFICITY CALCULATION
- BOX MODEL
- CSS POSITIONING

**Kavindu Samarasinghe**
B.Sc in Computer Science,
G.Dip in Software Engineering

# WHAT IS CSS?

CSS, or Cascading Style Sheets, is a style sheet language used to describe the presentation of HTML documents. CSS can be used to control the appearance of text, images, and other elements on a web page.



# THE FATHER OF CSS

- Hakon wium lie is know as father of css.
- CSS was proposed in 1994 as a web styling language, to solve some of the problems of Html 4.

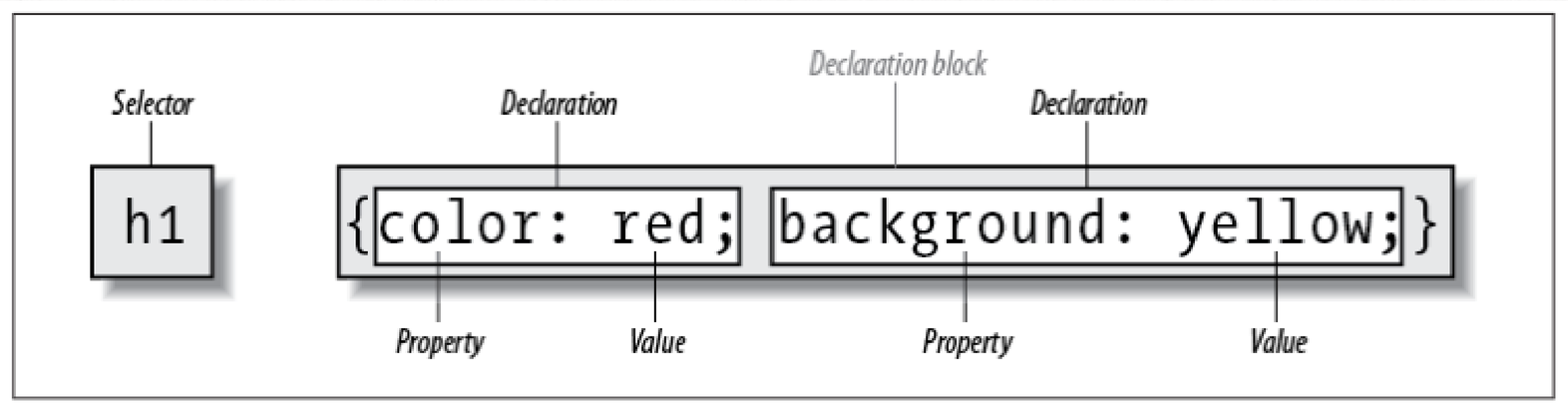
# APPLY CSS

There are 3 ways to apply CSS to HTML

| Inline CSS | Internal CSS (Embedded CSS) | External CSS |
| --- | --- | --- |

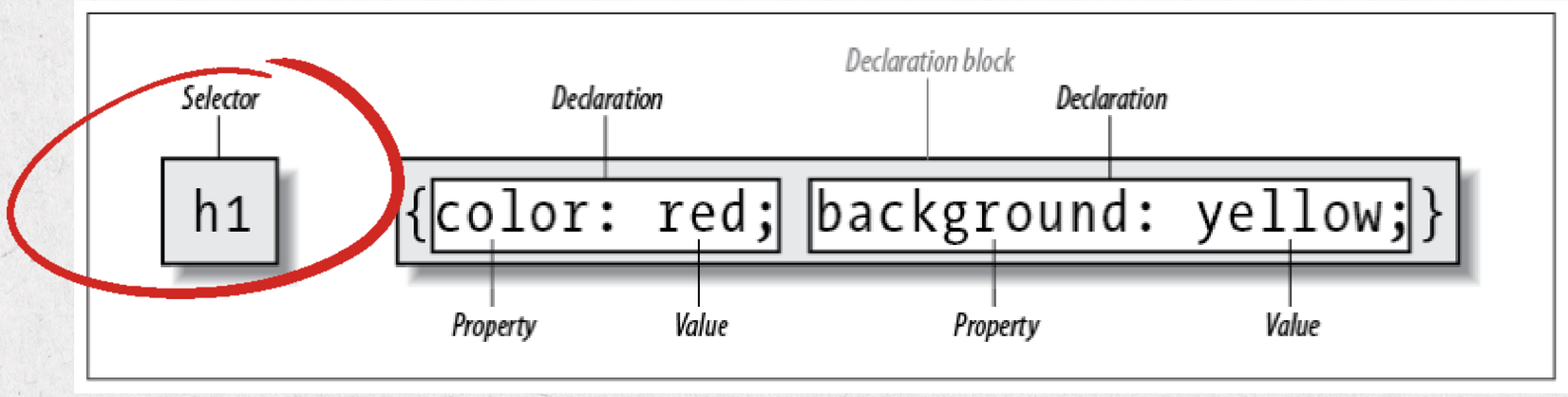
# CSS RULE

A CSS rule consists of a **CSS selector and a set of CSS properties.** The CSS selector determines what HTML elements to target with the CSS rule. The CSS properties specifies what to style of the targeted HTML elements.



# CSS SELECTOR

A CSS selector is the first part of a CSS Rule. It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them.



# CSS SELECTOR

# CSS SELECTORS LIST

- Type selector
- Class selector
- ID selector
- Universal selector
- Attribute selector
- Pseudo-classes
- Pseudo-elements
- Grouping selector

# TYPE SELECTOR

The type selector selects elements based on their HTML tag name. It targets all elements of a specific tag type on the page.

CSS

```
h1 {
  color: red;
  font-size: 24px;
}
```

# CLASS SELECTOR

A class selector is used to select all elements with a particular class name. For example, the selector .my-class would select all elements with the class name my-class.

CSS

```
.my-class {
  color: blue;
  font-size: 16px;
}
```

# ID SELECTOR

An ID selector is used to select a single element with a particular ID. For example, the selector #my-id would select the element with the ID my-id.

CSS

```
#my-id {
  color: green;
  font-size: 12px;
}
```

# UNIVERSAL SELECTOR

The universal selector (*) is used to select all elements on a web page. For example, the selector * would select all elements on a web page.

CSS

```
* {
  color: black;
  font-size: 14px;
}
```

# ATTRIBUTE SELECTOR

An attribute selector is used to select elements with a particular attribute. For example, the selector a[href] would select all a elements with an href attribute.

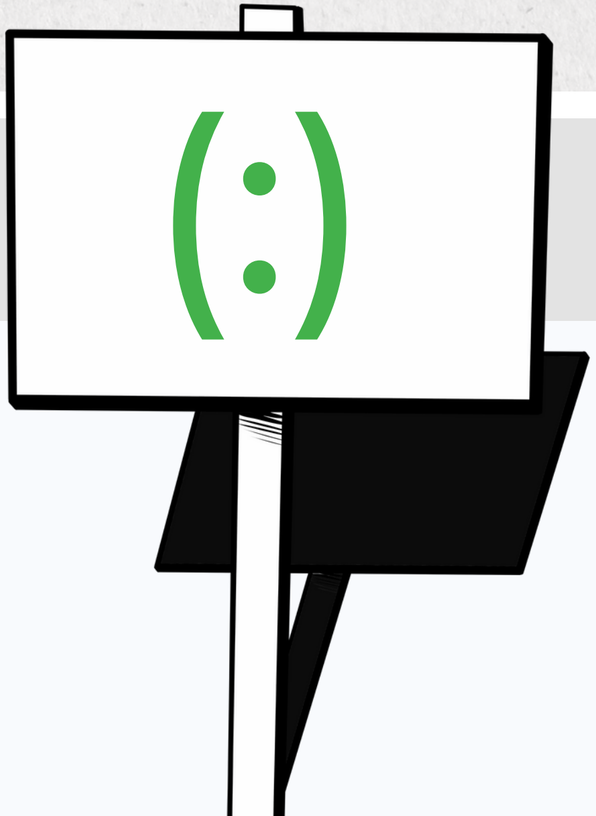
CSS

```
a[href] {
  color: red;
  text-decoration: underline;
}
```

# PSEUDO CLASS SELECTOR

CSS

```
a:hover {
  color: orange;
  text-decoration: none;
}
```

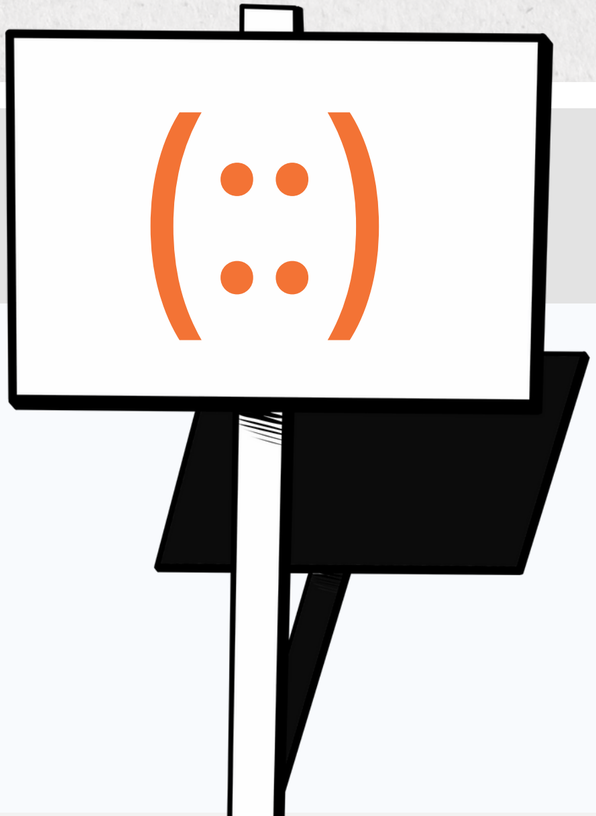


# PSEUDO ELEMENTS SELECTOR

Pseudo-elements are used to select parts of elements. For example, the pseudo-element ::first-letter would select the first letter of an element.

CSS

```
p::first-letter {
  font-size: 24px;
  font-weight: bold;
}
```



# GROUPING SELECTOR

A grouping selector is used to select a group of elements. For example, the selector p, div would select all p and div elements on a web page.

CSS

```
p, div {
  color: black;
  font-size: 14px;
}
```

# CSS COMBINATORS

Make relationships between the selectors

- **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 ( ~ )

# CSS CASCADE

The cascade is the conflict where multiple CSS rules apply to an HTML element.

CSS

```
button {
  color: red;
}

button {
  color: blue;
}
```

# SPECIFICITY CALCULATIONS

Specificity describes the weight of a selector and any declarations associated with it.

| Selector type | Example | Specificity |
|---|---|---|
| Universal selector | `*` | 0,0,0,0 |
| Combinator | `+` | |
| Element identifier | `div` | 0,0,0,1 |
| Pseudo-element identifier | `::first-line` | |
| Class identifier | `.warning` | 0,0,1,0 |
| Pseudo-class identifier | `:hover` | |
| Attribute identifier | `[type="checkbox"]` | |
| ID identifier | `#content` | 0,1,0,0 |
| Inline `style` attribute | `style="color: red;"` | 1,0,0,0 |

# BOX MODEL

The CSS box model is **essentially a box that wraps around every HTML element.** It consists of: margins, borders, padding, and the actual content.

# VALUES IN CSS

- Integers and real numbers
- Lengths & font values
- Percentages
- URLs and URIs
- Counters
- Colors
- Keywords
- Unsupported values
- Functions

# MEASUREMENT UNITS

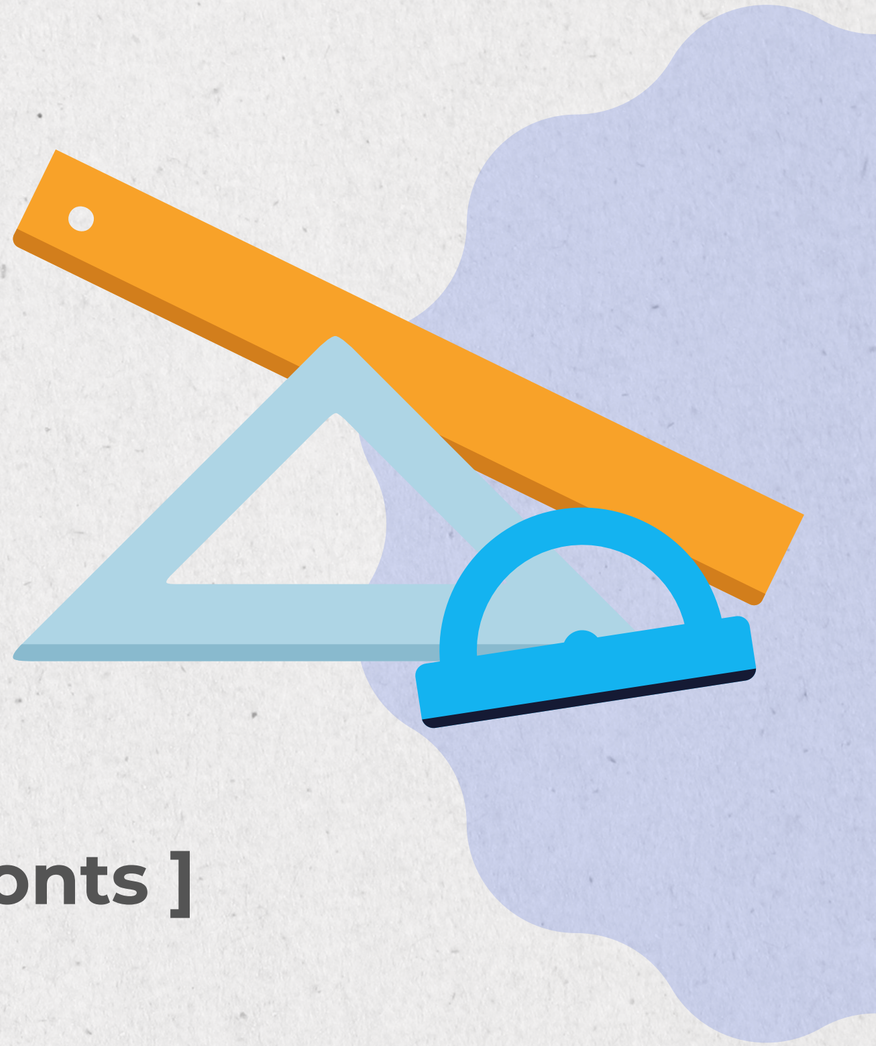
## (LENGTHS & FONT VALUES)

Relative measurement units

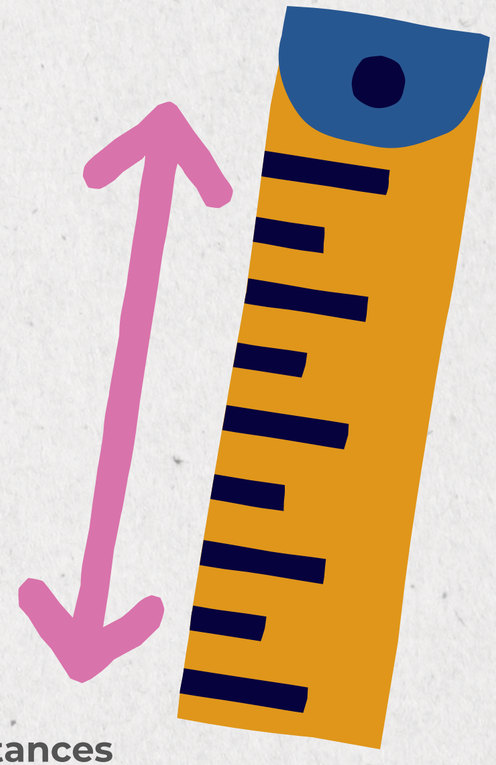
Absolute measurement units

# Relative measurement units

- vw - viewport width [ length ]
- vh - viewport height [ length ]
- em - equal measurement unit [ fonts ]
- rem - root equal measurement unit [ fonts ]
- % - percentage [length]



# Absolute measurement units



- cm [ length ]
- mm [ length ]
- in - inch [ length ]
- pc -

  [ length ] parsec (The parsec (symbol: pc) is a unit of length used to measure the large distances to astronomical objects outside the Solar System)
- pt -

  [ length ] Percentage point (Percentage point (pt), a unit for the arithmetic difference of two percentages)
- px - pixel [ length & font ]

# CSS POSITIONING

CSS positioning is a way to control the position of elements on a web page. It allows you to move elements around, make them stay in place, and overlap other elements.

# TYPES OF CSS POSITIONING

There are 5 types of CSS Positions

- **Static**
- **Relative**
- **Fixed**
- **Absolute**
- **Sticky**

# STATIC

- The default positioning method
- Elements are positioned according to the flow of the document
- No top, bottom, left, or right properties are used

# RELATIVE

- Elements are positioned relative to their original position
- The top, bottom, left, and right properties can be used to move the element around

# FIXED

- Elements are positioned relative to the viewport
- They stay in the same place even when the page is scrolled

# ABSOLUTE

- Elements are positioned relative to the nearest positioned ancestor
- The top, bottom, left, and right properties can be used to move the element around

# STICKY

- Elements are positioned like relative elements until they reach the top or bottom of the viewport
- Then, they are positioned like fixed elements

# THANK YOU!

**Kavindu Samarasinghe**
B.Sc in Computer Science,
G.Dip in Software Engineering

IJSE - INSTITUTE OF SOFTWARE ENGINEERING

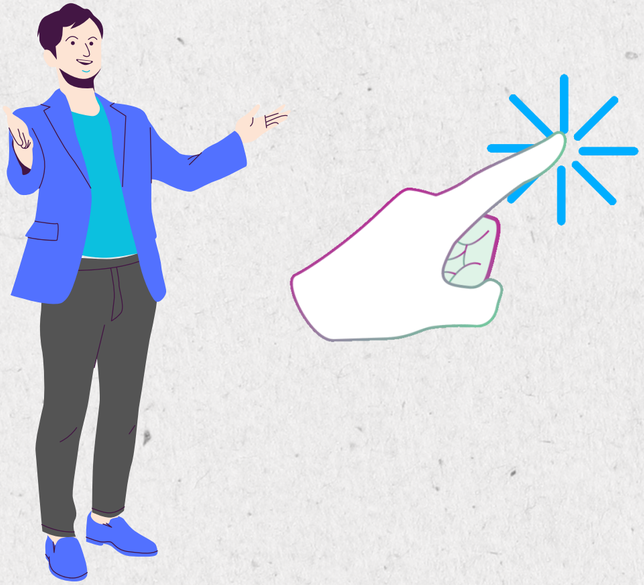
# CSS

## CSS TRANSITION AND ANIMATION

**Kavindu Samarasinghe**
B.Sc in Computer Science,
G.Dip in Software Engineering

# CSS TRANSITION AND ANIMATION

Transition



Animation

# WHAT IS TRANSITION?

CSS transitions smoothly change the style of an element over a specified duration when a certain condition is met, such as hovering over or clicking on the element.

```
button {
    background-color: #007bff;
    color: white;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #0056b3;
}
```

# WHAT IS ANIMATION?

CSS animations allow for dynamic movement and style changes of an element by defining keyframes that specify the intermediate styles at various points during the animation's duration.

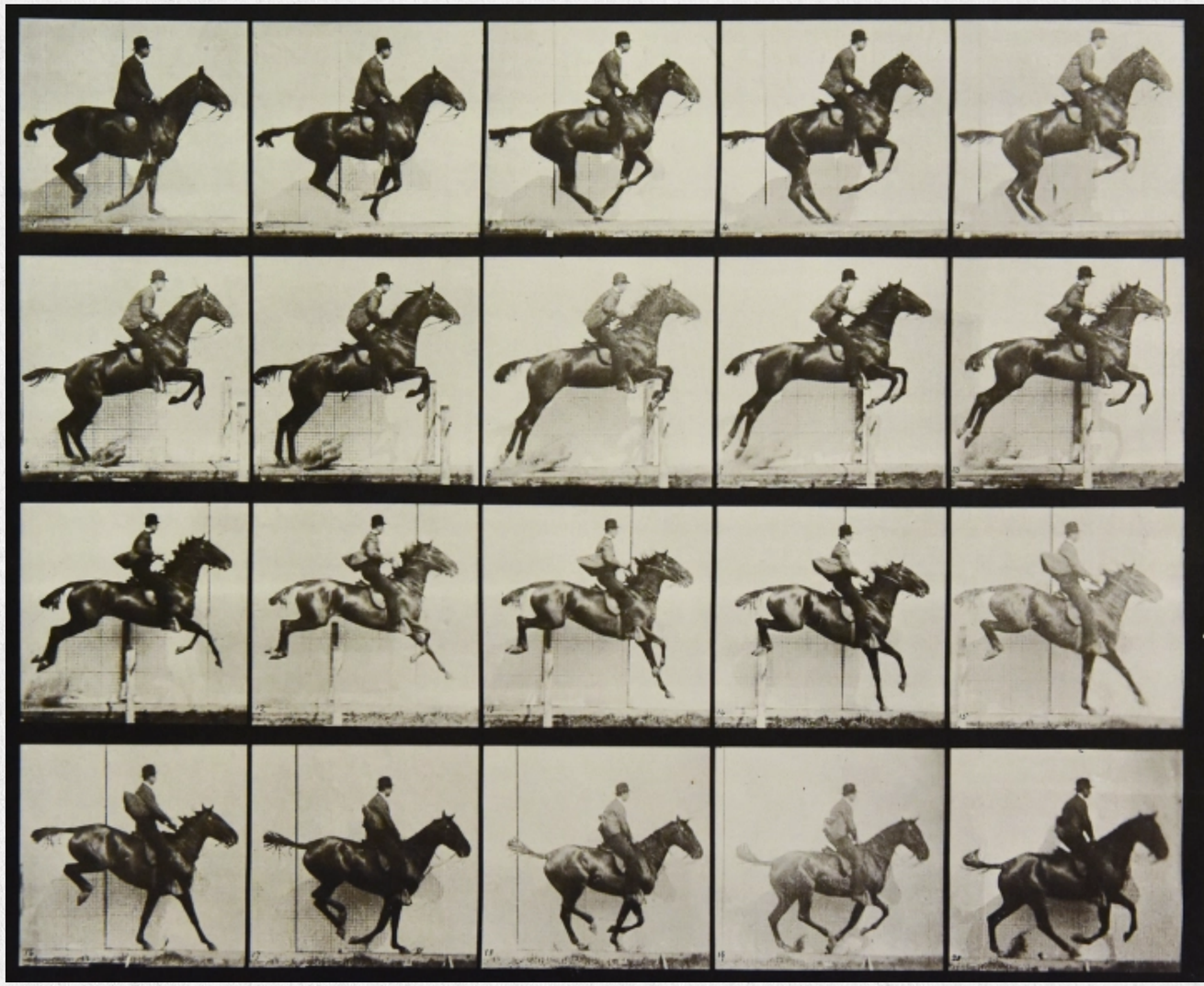
```
@keyframes spin {
    from {
        transform: rotate(0deg);
    }
    to {
        transform: rotate(360deg);
    }
}

.spinner {
    width: 50px;
    height: 50px;
    background-color: #007bff;
    animation: spin 2s linear infinite;
}
```

# KEY FRAMES

Keyframes are used to specify the styles that an element should have at various points in time during the animation's duration.

Keyframes allow you to define the intermediate steps of an animation, indicating how the element's properties should change over time.



# ZOETROPE

Zoetrope Animation Toy Of A Galloping Horse | Zoetrope
Share
www.Zoetrope.org
Watch on YouTube

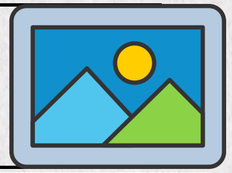
# MODERN ZOETROPE

# @keyframes

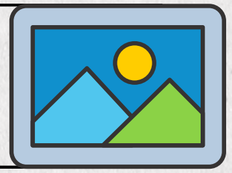
@keyframes is a rule that defines the animation behavior at various points in time. It's used to create animations by specifying the styles that an element should have at different stages of the animation's duration.

```
@keyframes spin {
    from {
        transform: rotate(0deg);
    }
    to {
        transform: rotate(360deg);
    }
}
```

**Frame 01**



**Frame 02**



# Various animation-related properties

1. **animation-name:** Specifies the name of the animation defined using @keyframes.
2. **animation-duration:** Sets the duration over which the animation should occur, measured in seconds (s) or milliseconds (ms).
3. **animation-timing-function:** Defines the timing function that controls the pace of the animation's progression, such as ease, linear, ease-in, ease-out, ease-in-out, or custom cubic-bezier functions.
4. **animation-delay:** Specifies the delay before the animation starts, allowing you to stagger animations.
5. **animation-iteration-count:** Determines the number of times the animation should repeat, or whether it should repeat indefinitely using the value infinite.
6. **animation-direction:** Sets the direction of the animation playback, such as normal, reverse, alternate, or alternate-reverse.
7. **animation-fill-mode:** Specifies how the animation's styles are applied before and after the animation plays, such as none, forwards, backwards, or both.
8. **animation-play-state:** Controls whether the animation is running or paused, with values running or paused.

# animation property

The animation property in CSS is indeed a shorthand property that combines several individual animation-related properties into one concise declaration.

```
.element {
    animation: fadeInOut 3s ease-in-out 1s infinite alternate both running;
}
```

animation-name

animation-duration

animation-timing-function

animation-delay

animation-iteration-count

animation-direction

animation-fill-mode

animation-play-state

# THANK YOU!

**Kavindu Samarasinghe**
B.Sc in Computer Science,
G.Dip in Software Engineering

Introduction to

# Bootstrap

# | What is Bootstrap |

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

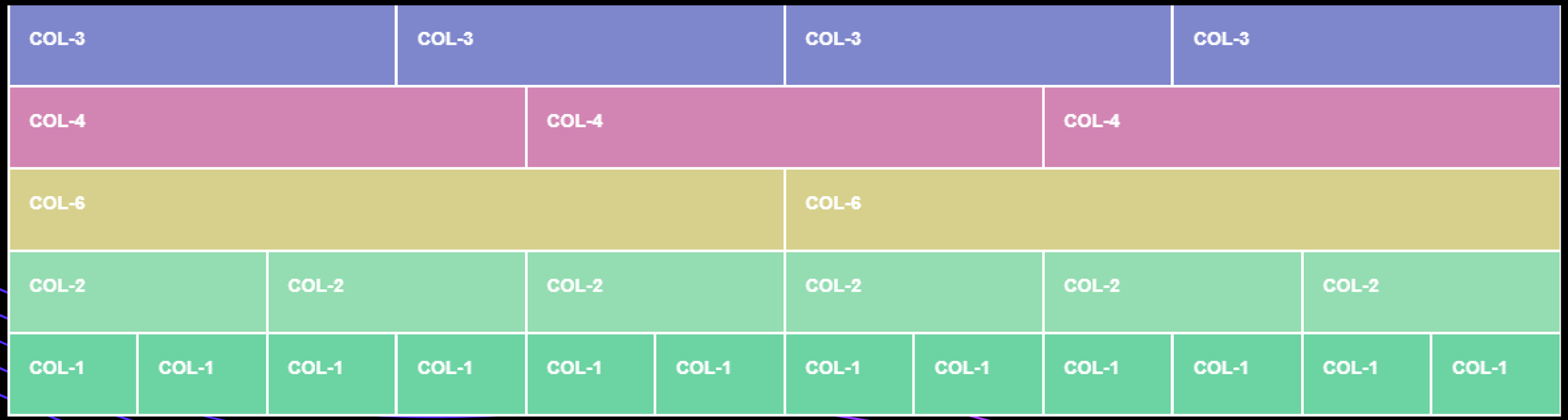
# | How to Use |

A basic understanding of HTML is required to start learning Bootstrap. Some familiarity with how CSS works (CSS Selectors and Visual Rules) would be helpful,

1. Download Bootstrap from getbootstrap.com
2. Include Bootstrap from a CDN

# | Bootstrap Grid |

The Bootstrap grid system has four classes: xs (<576px wide) sm ( >576px wide) md (for >768px wide) lg ( >992px wide) xl ( >1200px wide) XXl (For >1400px wide)



# | Viewport |

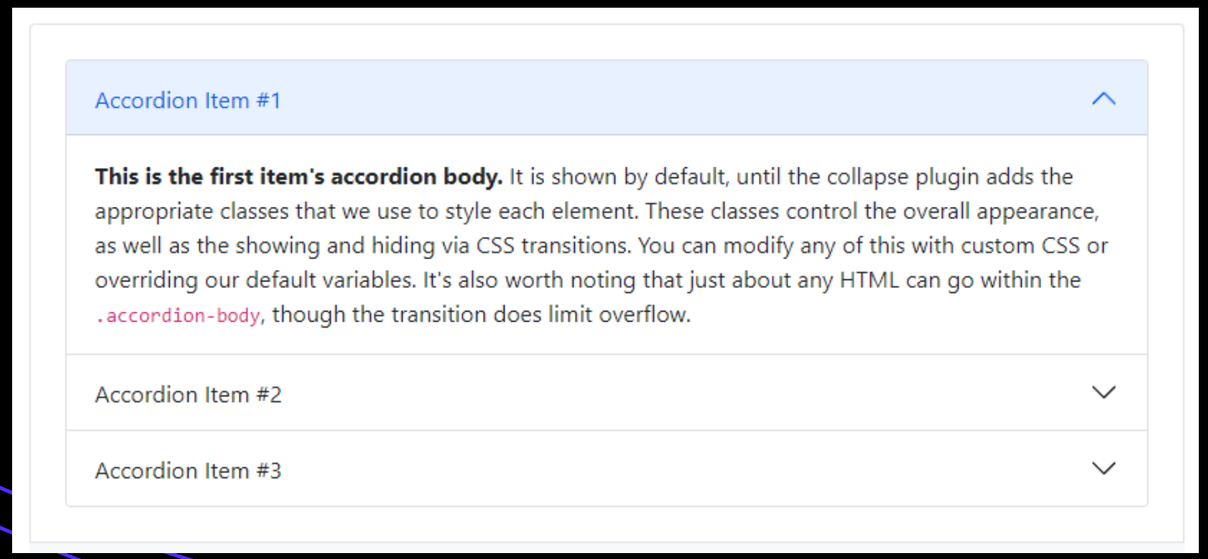
The viewport is the user's visible area of a web page.
The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

# | Components |

Many more reusable components built to provide iconography, dropdowns, input groups, navigation, alerts, and much more.

## Accordion




```
<div class="accordion" id="accordionExample">
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingOne">
      <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#co
        Accordion Item #1
      </button>
    </h2>
    <div id="collapseOne" class="accordion-collapse collapse show" aria-labelledby="headingOne" da
      <div class="accordion-body">
        <strong>This is the first item's accordion body.</strong> It is shown by default, until th
      </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingTwo">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-t
        Accordion Item #2
      </button>
    </h2>
    <div id="collapseTwo" class="accordion-collapse collapse" aria-labelledby="headingTwo" data-bs
      <div class="accordion-body">
        <strong>This is the second item's accordion body.</strong> It is hidden by default, until
      </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingThree">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-t
        Accordion Item #3
      </button>
    </h2>
    <div id="collapseThree" class="accordion-collapse collapse" aria-labelledby="headingThree" dat
      <div class="accordion-body">
        <strong>This is the third item's accordion body.</strong> It is hidden by default, until t
      </div>
    </div>
  </div>
</div>
```

# | Components |

## Alerts

A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

A simple danger alert—check it out!

A simple warning alert—check it out!

A simple info alert—check it out!

A simple light alert—check it out!

A simple dark alert—check it out!

```html
<div class="alert alert-primary" role="alert">
  A simple primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
  A simple success alert—check it out!
</div>
<div class="alert alert-danger" role="alert">
  A simple danger alert—check it out!
</div>
<div class="alert alert-warning" role="alert">
  A simple warning alert—check it out!
</div>
<div class="alert alert-info" role="alert">
  A simple info alert—check it out!
</div>
<div class="alert alert-light" role="alert">
  A simple light alert—check it out!
</div>
<div class="alert alert-dark" role="alert">
  A simple dark alert—check it out!
</div>
```

# | Components |

- Accordion
- Alerts
- Badge
- Breadcrumb
- Buttons
- Button group
- Card
- Carousel
- Close button
- Collapse
- Dropdowns
- List group
- Modal
- Navs & tabs
- Navbar
- Offcanvas
- Pagination
- Popovers
- Progress
- Scrollspy
- Spinners
- Toasts
- Tooltips

# | Advantages of Bootstrap |

1. Fewer Cross browser bugs
2. A consistent framework that supports major of all browsers and CSS compatibility fixes
3. Lightweight and customizable
4. Responsive structures and styles
5. Several JavaScript plugins using the jQuery
6. Good documentation and community support
7. Loads of free and professional templates, WordPress themes and plugins
8. Great grid system

# THANK YOU!