

The background features a dark blue gradient with three glowing, translucent 3D torus shapes. One large ring is positioned on the left side, another smaller one is at the bottom left corner, and a third smaller one is located on the right side.

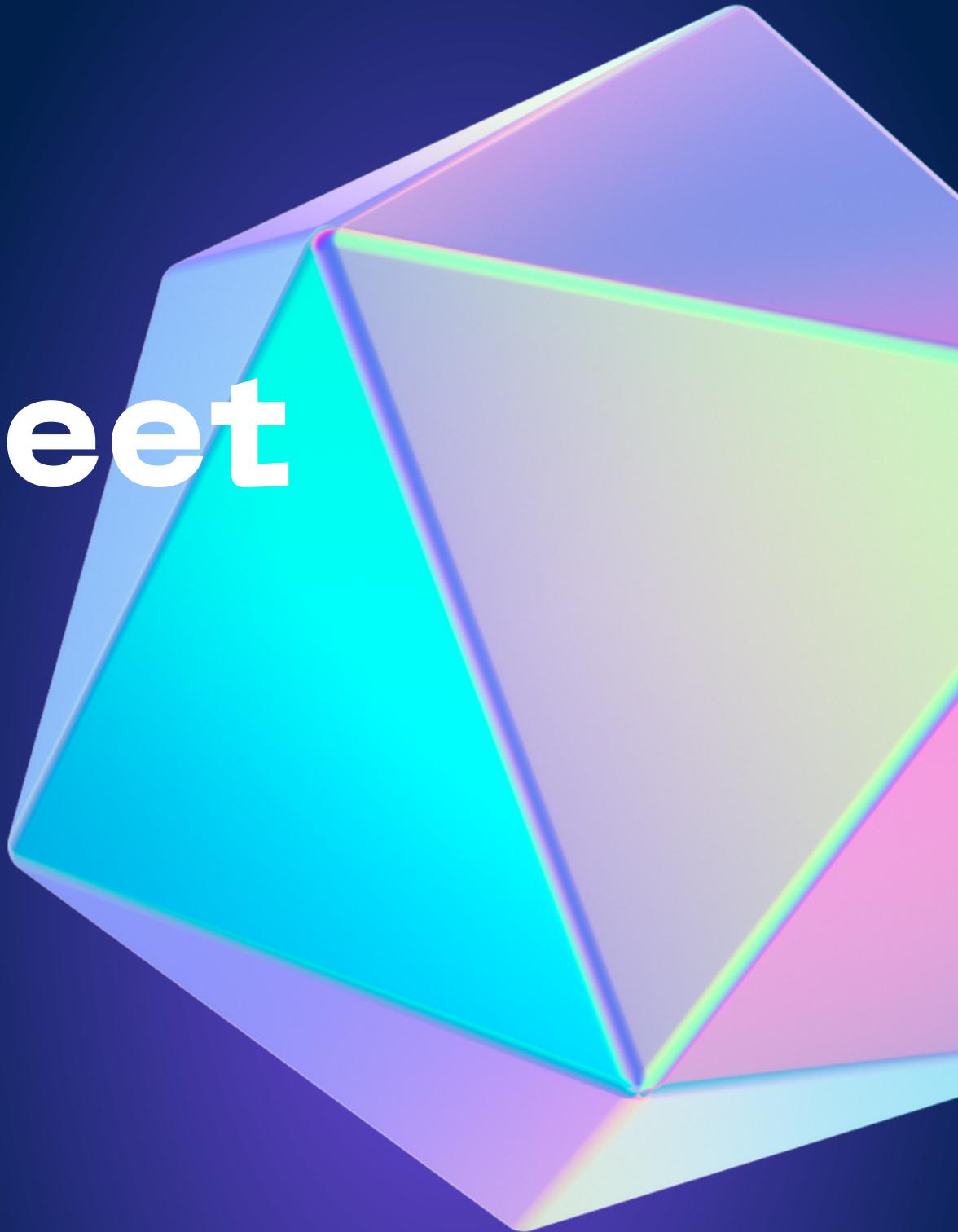
CSS

Rohan Chhetry



Cascading Style Sheet

CSS describes how HTML elements are to be displayed on screen, paper, or in other media



Adding CSS

index.html

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

```
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

Internal

External

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

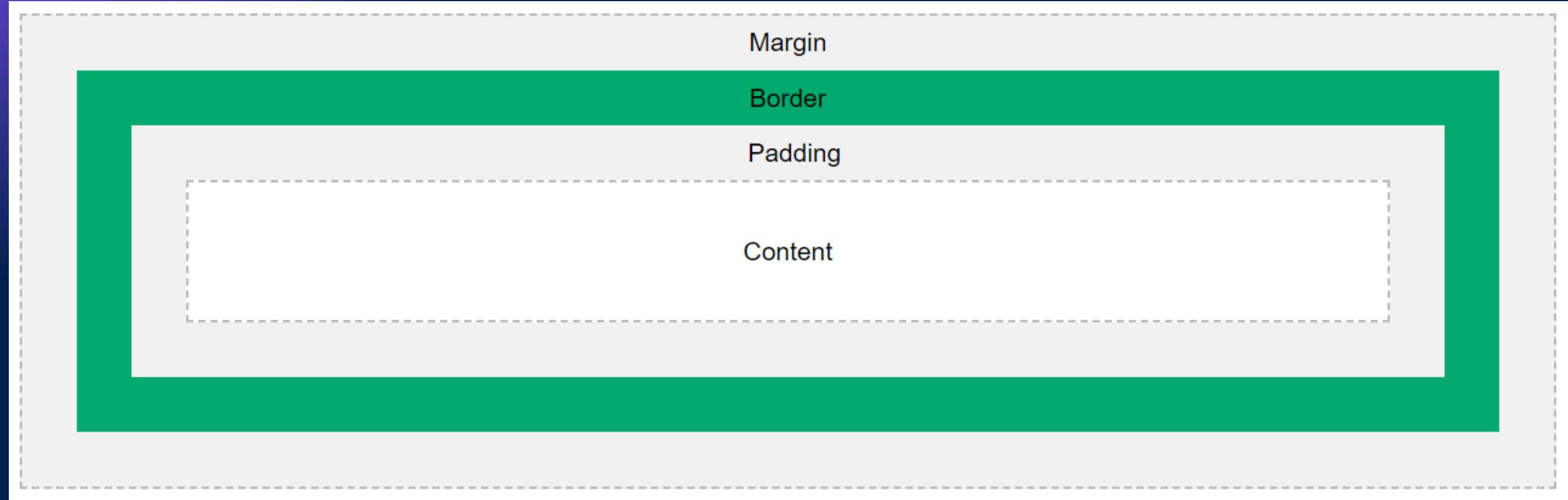
Inline

Selector

- CSS rules consist of a selector, property, and value.
- Selectors target HTML elements to apply styles

```
/* Selector: Targeting all paragraphs */  
p {  
  
    /* Property: Styling text color */  
    color: blue;  
  
    /* Property: Adding padding */  
    padding: 10px;  
}
```

Box Model

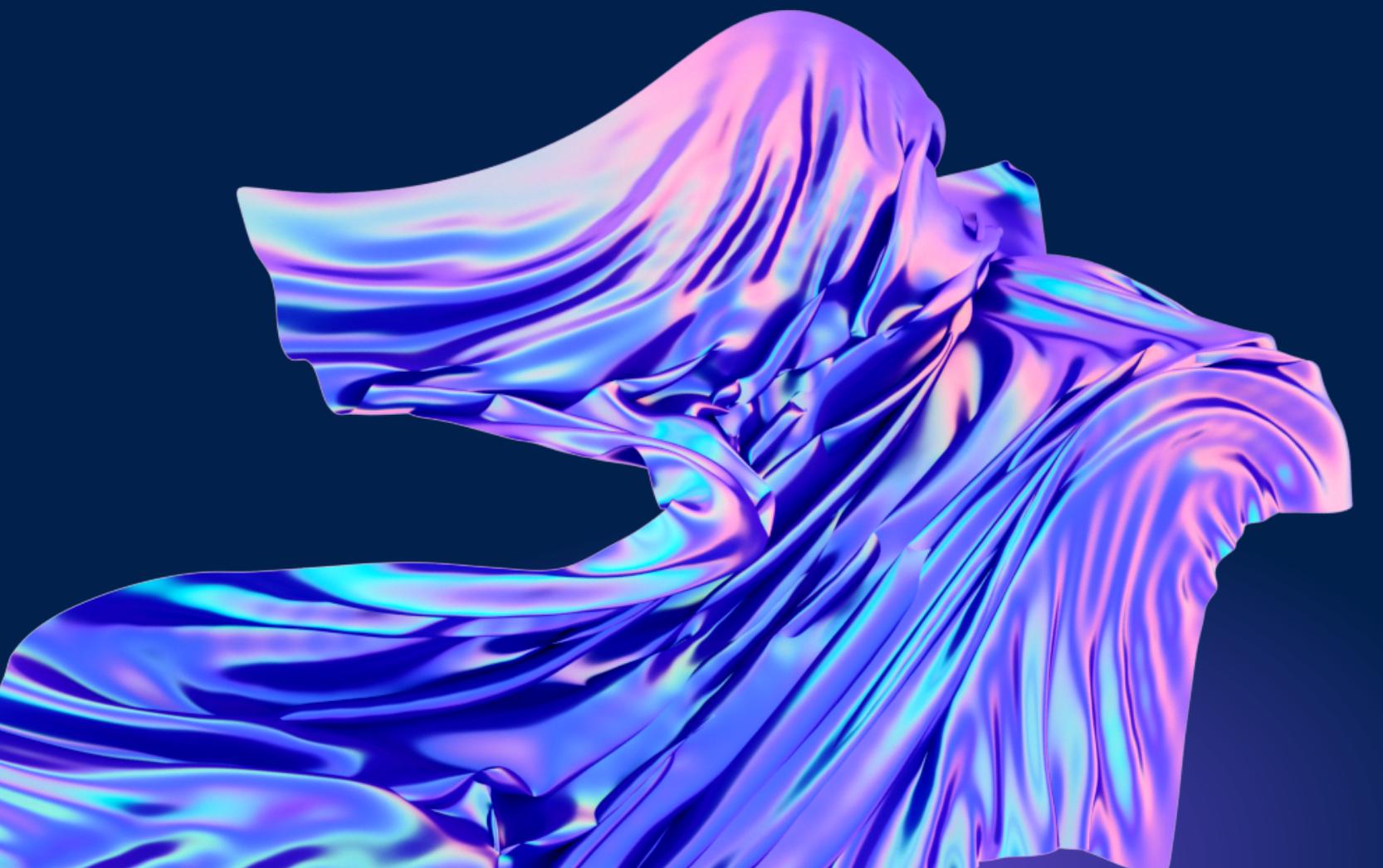


- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

```
/* Applying box model properties to a div */
.box {
  width: 200px;
  padding: 20px;
  border: 2px solid black;
  margin: 10px;
}
```

Color and Background

- Use various color representations (names, hex, RGB).
- Set background properties for elements.



```
/* Styling a header with color and  
background */  
h1 {  
    color: red;  
    color: rgb(52, 152, 219);  
    background: #ecf0f1;  
}
```



Typography

- Control font properties like family, size, weight, and style.
- Adjust text alignment and spacing

```
/* Styling paragraphs with different  
typography properties */
```

```
p {  
    font-family: 'Arial', sans-serif;  
    font-size: 16px;  
    font-weight: bold;  
    text-align: center;  
    letter-spacing: 2px;  
}
```

Layout

- Understand the normal flow, flexbox, and grid layout models.
- Position elements using relative, absolute, and fixed positioning

```
.box-model-example {  
    width: 200px;  
    height: 100px;  
    padding: 20px;  
    margin: 10px;  
    border: 1px solid #333;  
}
```

Layout

```
.flex-container {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
.flex-item {  
  flex: 1;  
  padding: 10px;  
  border: 1px solid #333;  
}
```

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3,  
  1fr);  
  gap: 10px;  
}
```

```
.grid-item {  
  padding: 20px;  
  border: 1px solid #333;  
}
```

Responsive Design

Use media queries to apply styles based on screen characteristics

```
/* Hide navigation on small screens */  
@media only screen and (max-width:  
600px) {  
  nav {  
    display: none;  
  }  
}  
}
```

```
/* Apply styles for screens with a  
width between 600px and 1200px */  
@media only screen and (min-width:  
600px) and (max-width: 1200px) {  
  body {  
    font-size: 16px;  
  }  
}
```

Transitions and Animations

- Apply transitions for smooth state changes.
- Use keyframes for animations

```
/* Adding a transition to a button */  
button {  
    transition: background 0.3s ease-in-out;  
}
```

Questions

Create a simple HTML page and apply styles to it using each of the three methods: inline, internal, and external stylesheets

- **Build a webpage with three sections, each with a different background color using hex, RGB, and color names.**
- **Style a navigation bar with a gradient background using the linear-gradient property.**

- **Create a webpage with different font sizes for headings, paragraphs, and links.**
- **Implement a Google Font on your webpage and demonstrate how to apply it to specific elements.**

- **Build a simple box layout with a header, sidebar, content area, and footer.**
- **Design a navigation menu using Flexbox and center the items both horizontally and vertically.**
- **Create a grid layout for a photo gallery with a responsive number of columns based on the screen size.**

- **Develop a basic two-column layout that becomes a single column on smaller screens using media queries.**
- **Apply styles to hide and show different elements based on the screen width.**

- **Add a smooth transition effect to a button when hovered over.**
- **Create a simple CSS animation to make an element move or fade in.**



Grouping Tags in Css

- Can use various methods to group styles and apply them to multiple elements.
- Grouping styles helps you write cleaner and more maintainable code



Selector & Element

```
.h1, h2, h3 {  
    color: blue;  
    font-family: Arial, sans-serif;  
}
```

```
.container {  
    width: 100%;  
    margin: 0 auto;  
}  
  
.header, .footer {  
    background-color: #eee;  
    padding: 10px;  
}
```

Class & Descendant

```
.primary-btn, .secondary-btn {  
    display: inline-block;  
    padding: 8px 16px;  
    text-decoration: none;  
}
```

```
article p {  
    line-height: 1.5;  
}
```

Universal Selector & Media Queries

```
* {  
margin: 0;  
padding: 0;  
box-sizing: border-box;  
}
```

```
@media screen and (max-width: 600px) {  
body {  
font-size: 14px;  
}  
.header {  
display: none;  
}  
}
```

Tableless Website Sample

- Deployment:
https://0000blaze.github.io/Tableless_webpage/
- Code:
https://github.com/0000Blaze/Tableless_webpage