

CSS Variables Review

CSS Custom Properties (CSS Variables)

- **Definition:** CSS custom properties, also known as CSS variables, are entities defined by CSS authors that contain specific values to be reused throughout a document. They are a powerful feature that allows for more efficient, maintainable, and flexible stylesheets. Custom properties are particularly useful in creating themeable designs. You can define a set of properties for different themes:

```
:root {  
  --bg-color: white;  
  --text-color: black;  
}
```

```
.dark-theme {  
  --bg-color: #333;  
  --text-color: white;  
}
```

```
body {  
  background-color: var(--bg-color);  
  color: var(--text-color);  
}
```

The @property Rule

- **Definition:** The @property rule is a powerful CSS feature that allows developers to define custom properties with greater control over their behavior, including how they animate and their initial values.

```
@property --property-name {  
  syntax: '<type>';
```

```
inherits: true | false;  
initial-value: <value>;  
}
```

- **--property-name:** This is the name of the custom property you're defining. Like all custom properties, it must start with two dashes. --property-name can be things like <color>, <length>, <number>, <percentage>, or more complex types.
- **syntax:** This defines the type of the property.
- **inherits:** This specifies whether the property should inherit its value from its parent element.
- **initial-value:** This sets the default value of the property.
- **Gradient Example Using the @property Rule:** This example creates a gradient that smoothly animates when the element is hovered over.

```
<div class="gradient-box"></div>
```

```
@property --gradient-angle {  
  syntax: "<angle>";  
  inherits: false;  
  initial-value: 0deg;  
}
```

```
.gradient-box {  
  width: 100px;  
  height: 100px;  
  background: linear-gradient(var(--gradient-angle), red, blue);  
  transition: --gradient-angle 0.5s;  
}
```

```
.gradient-box:hover {
```

```
--gradient-angle: 90deg;  
}
```

- **Fallbacks:** When using the custom property, you can provide a fallback value using the `var()` function, just as you would with standard custom properties:

```
.button {  
  background-color: var(--main-color, #3498db);  
}
```