

Advanced CSS: Comprehensive Guide



Advanced CSS: Comprehensive Guide

Advanced CSS Concepts

Advanced Selectors

1. Attribute Selectors

2. Pseudo-classes

3. Pseudo-elements

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

4. Combinators	4
CSS Variables	4
Responsive Design	5
Media Queries	5
Viewport Units	5
Advanced Layout Techniques	5
CSS Grid	5
Flexbox	6
CSS Effects	6
Transitions	6
Clip Path	7
Exercises	7
Exercise 1: Advanced Selectors	7
Solution:	7
Exercise 2: Responsive Box	7
Solution:	7
Multiple-Choice Questions	8
Question 1:	8
Question 2:	8
Question 3:	8
Advanced Example	9
Create a Profile Card with Advanced CSS	9

This guide is designed to teach advanced CSS concepts to help you build responsive, interactive, and aesthetically pleasing websites. We'll explore advanced selectors, layout techniques, animations, and best practices.

Advanced CSS Concepts

1. **Advanced Selectors**
 - Attribute Selectors
 - Pseudo-classes
 - Pseudo-elements
 - Combinators
2. **CSS Variables**
 - Defining and using CSS custom properties.
3. **Responsive Design**
 - Media Queries
 - Viewport Units
 - CSS Grid and Flexbox

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

4. **Advanced Layout Techniques**

- Flexbox and Grid for complex layouts
- Positioning and layering with z-index

5. **CSS Effects**

- Transitions and Animations
- 3D Transforms
- Clip Path and Masking

6. **Best Practices**

- CSS Architecture (e.g., BEM methodology)
- Writing maintainable and reusable CSS
- Debugging and optimizing CSS performance

Advanced Selectors

1. Attribute Selectors

Attribute selectors allow you to style elements based on their attributes and attribute values.

```
<input type="text" placeholder="Enter text" />
<input type="password" placeholder="Enter password" />
<style>
  input[type="text"] {
    border: 2px solid blue;
  }
  input[type="password"] {
    border: 2px solid red;
  }
</style>
```

2. Pseudo-classes

```
<a href="#">Normal Link</a>
<a href="#" class="visited">Visited Link</a>
<style>
  a:hover {
    color: green;
  }
  a:visited {
    color: purple;
  }
</style>
```

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

3. Pseudo-elements

```
<p>Learn CSS Pseudo-elements.</p>
<style>
  p::first-line {
    font-weight: bold;
    color: blue;
  }
  p::after {
    content: " [Read more]";
    color: gray;
    font-style: italic;
  }
</style>
```

4. Combinators

```
<div>
  <p>Paragraph 1</p>
  <span>Span inside div</span>
</div>
<style>
  div > p {
    color: red;
  }
  div span {
    color: blue;
  }
</style>
```

CSS Variables

CSS Variables, or custom properties, allow you to define reusable values.

```
<div class="box"></div>
<style>
  :root {
    --main-color: teal;
    --secondary-color: coral;
    --box-size: 100px;
  }
</style>
```

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

```

}
.box {
  width: var(--box-size);
  height: var(--box-size);
  background-color: var(--main-color);
  border: 2px solid var(--secondary-color);
}
</style>

```

Responsive Design

Media Queries

```

<div class="responsive-box"></div>
<style>
  .responsive-box {
    width: 300px;
    height: 300px;
    background: lightblue;
  }
  @media (max-width: 600px) {
    .responsive-box {
      background: coral;
    }
  }
</style>

```

Viewport Units

```

<div class="full-height-box"></div>
<style>
  .full-height-box {
    width: 100vw; /* 100% of the viewport width */
    height: 100vh; /* 100% of the viewport height */
    background: lavender;
  }
</style>

```

Advanced Layout Techniques

CSS Grid

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
<style>
  .grid-container {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 10px;
  }
</style>
```

Flexbox

```
<div class="flex-container">
  <div>Item 1</div>
  <div>Item 2</div>
  <div>Item 3</div>
</div>
<style>
  .flex-container {
    display: flex;
    justify-content: space-around;
  }
</style>
```

CSS Effects

Transitions

```
<div class="hover-box"></div>
<style>
  .hover-box {
    width: 100px;
    height: 100px;
    background: pink;
    transition: transform 0.3s;
  }
  .hover-box:hover {
```

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

```
        transform: scale(1.2);
    }
</style>
```

Clip Path

```
<div class="clipped-box"></div>
<style>
    .clipped-box {
        width: 200px;
        height: 200px;
        background: orange;
        clip-path: circle(50%);
    }
</style>
```

Exercises

Exercise 1: Advanced Selectors

Style all input fields of type email with a green border. Highlight the input when it is focused.

Solution:

```
<input type="email" placeholder="Enter email" />
<style>
    input[type="email"] {
        border: 2px solid green;
        padding: 5px;
    }
    input[type="email"]:focus {
        outline: none;
        border-color: blue;
    }
</style>
```

Exercise 2: Responsive Box

Create a box that changes its background color based on the screen width (less than 500px: red, 500px or more: blue).

Solution:

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

```
<div class="responsive-box"></div>
<style>
  .responsive-box {
    width: 100px;
    height: 100px;
    background: blue;
  }
  @media (max-width: 500px) {
    .responsive-box {
      background: red;
    }
  }
</style>
```

Multiple-Choice Questions

Question 1:

What does `:root` represent in CSS?

1. The first element in the DOM.
2. The root element of the document, typically `<html>`.
3. A CSS pseudo-class for selecting parent elements.
4. None of the above.

Answer: 2. The root element of the document, typically `<html>`.

Question 2:

Which CSS property is used to define reusable variables?

1. `@variables`
2. `--custom`
3. `var()`
4. Custom properties using `--`.

Answer: 4. Custom properties using `--`.

Question 3:

What does `clip-path` do?

Learn more HTML, CSS, JavaScript Web Development at <https://basescripts.com/> Laurence Svekis

1. Clips elements inside a grid container.
2. Defines a clipping region to show part of an element.
3. Animates the element's border.
4. None of the above.

Answer: 2. Defines a clipping region to show part of an element.

Advanced Example

Create a Profile Card with Advanced CSS

```
<div class="profile-card">
  
  <h2>John Doe</h2>
  <p>Web Developer</p>
</div>
<style>
.profile-card {
  width: 250px;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 10px;
  text-align: center;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  transition: transform 0.3s;
}
.profile-card:hover {
  transform: translateY(-10px);
}
.profile-card img {
  border-radius: 50%;
  margin-bottom: 10px;
}
</style>
```