

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## 1. What is CSS and what does it stand for?

### What is it?

What is CSS and what does it stand for? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is CSS and what does it stand for? in creating scalable, maintainable design systems.
2. Applied What is CSS and what does it stand for? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 2. Explain the purpose of CSS in web design.

### What is it?

Explain the purpose of CSS in web design. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain the purpose of CSS in web design. in creating scalable, maintainable design systems.
2. Applied Explain the purpose of CSS in web design. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 3. What are the different types of CSS?

### What is it?

What are the different types of CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are the different types of CSS? in creating scalable, maintainable design systems.
2. Applied What are the different types of CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 4. How do you link CSS to an HTML document?

### What is it?

How do you link CSS to an HTML document? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Real-world Scenarios

1. Used How do you link CSS to an HTML document? in creating scalable, maintainable design systems.
2. Applied How do you link CSS to an HTML document? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 5. What is the syntax of a CSS rule?

### What is it?

What is the syntax of a CSS rule? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used What is the syntax of a CSS rule? in creating scalable, maintainable design systems.
2. Applied What is the syntax of a CSS rule? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 6. What are selectors in CSS? Give examples.

### What is it?

What are selectors in CSS? Give examples. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are selectors in CSS? Give examples. in creating scalable, maintainable design systems.
2. Applied What are selectors in CSS? Give examples. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 7. Difference between class selector and id selector.

### What is it?

Difference between class selector and id selector. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Difference between class selector and id selector. in creating scalable, maintainable design systems.
2. Applied Difference between class selector and id selector. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 8. What is the universal selector?

### What is it?

What is the universal selector? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is the universal selector? in creating scalable, maintainable design systems.
2. Applied What is the universal selector? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## 9. How do you apply multiple classes to an element?

### What is it?

How do you apply multiple classes to an element? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you apply multiple classes to an element? in creating scalable, maintainable design systems.
2. Applied How do you apply multiple classes to an element? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 10. What is specificity in CSS?

### What is it?

What is specificity in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is specificity in CSS? in creating scalable, maintainable design systems.
2. Applied What is specificity in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 11. How does the cascade work in CSS?

### What is it?

How does the cascade work in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How does the cascade work in CSS? in creating scalable, maintainable design systems.
2. Applied How does the cascade work in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 12. What are pseudo-classes in CSS?

### What is it?

What are pseudo-classes in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

1. Used What are pseudo-classes in CSS? in creating scalable, maintainable design systems.
2. Applied What are pseudo-classes in CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 13. What are pseudo-elements in CSS?

### What is it?

What are pseudo-elements in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are pseudo-elements in CSS? in creating scalable, maintainable design systems.
2. Applied What are pseudo-elements in CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.



# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 14. Difference between relative and absolute positioning.

### What is it?

Difference between relative and absolute positioning. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Difference between relative and absolute positioning. in creating scalable, maintainable design systems.
2. Applied Difference between relative and absolute positioning. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 15. Explain fixed and sticky positioning.

### What is it?

Explain fixed and sticky positioning. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain fixed and sticky positioning. in creating scalable, maintainable design systems.
2. Applied Explain fixed and sticky positioning. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
background-color: #f8f9fa;
margin: 0;
padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 16. What is the box model in CSS?

### What is it?

What is the box model in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is the box model in CSS? in creating scalable, maintainable design systems.
2. Applied What is the box model in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 17. How do you center an element using CSS?

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## What is it?

How do you center an element using CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used How do you center an element using CSS? in creating scalable, maintainable design systems.
2. Applied How do you center an element using CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 18. What is z-index in CSS?

### What is it?

What is z-index in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is z-index in CSS? in creating scalable, maintainable design systems.
2. Applied What is z-index in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
margin: 0 auto;
padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 19. Difference between em, rem, %, px units.

### What is it?

Difference between em, rem, %, px units. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Difference between em, rem, %, px units. in creating scalable, maintainable design systems.
2. Applied Difference between em, rem, %, px units. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 20. How do you make a website responsive with CSS?

### What is it?

How do you make a website responsive with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you make a website responsive with CSS? in creating scalable, maintainable design systems.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

2. Applied How do you make a website responsive with CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 21. Explain media queries and how they work.

### What is it?

Explain media queries and how they work. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain media queries and how they work. in creating scalable, maintainable design systems.
2. Applied Explain media queries and how they work. to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 22. What is Flexbox in CSS?

### What is it?

What is Flexbox in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is Flexbox in CSS? in creating scalable, maintainable design systems.
2. Applied What is Flexbox in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 23. How does justify-content and align-items work?

### What is it?

How does justify-content and align-items work? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How does justify-content and align-items work? in creating scalable, maintainable design systems.
2. Applied How does justify-content and align-items work? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
font-family: Arial, sans-serif;
background-color: #f8f9fa;
margin: 0;
padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 24. Explain the difference between Flexbox and Grid.

### What is it?

Explain the difference between Flexbox and Grid. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain the difference between Flexbox and Grid. in creating scalable, maintainable design systems.
2. Applied Explain the difference between Flexbox and Grid. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## 25. How to create a grid layout using CSS Grid.

### What is it?

How to create a grid layout using CSS Grid. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How to create a grid layout using CSS Grid. in creating scalable, maintainable design systems.
2. Applied How to create a grid layout using CSS Grid. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 26. What are CSS variables?

### What is it?

What are CSS variables? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are CSS variables? in creating scalable, maintainable design systems.
2. Applied What are CSS variables? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```



# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 27. How to declare and use custom properties (variables).

### What is it?

How to declare and use custom properties (variables). is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How to declare and use custom properties (variables). in creating scalable, maintainable design systems.
2. Applied How to declare and use custom properties (variables). to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 28. What are combinators in CSS?

### What is it?

What are combinators in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Real-world Scenarios

1. Used What are combinators in CSS? in creating scalable, maintainable design systems.
2. Applied What are combinators in CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 29. Difference between adjacent and general sibling selectors.

### What is it?

Difference between adjacent and general sibling selectors. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used Difference between adjacent and general sibling selectors. in creating scalable, maintainable design systems.
2. Applied Difference between adjacent and general sibling selectors. to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 30. What is the difference between visibility: hidden and display: none?

### What is it?

What is the difference between visibility: hidden and display: none? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is the difference between visibility: hidden and display: none? in creating scalable, maintainable design systems.
2. Applied What is the difference between visibility: hidden and display: none? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 31. How do you create animations with CSS?

### What is it?

How do you create animations with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you create animations with CSS? in creating scalable, maintainable design systems.
2. Applied How do you create animations with CSS? to build complex web application interfaces with responsive behavior.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 32. Explain keyframes in CSS animations.

### What is it?

Explain keyframes in CSS animations. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain keyframes in CSS animations. in creating scalable, maintainable design systems.
2. Applied Explain keyframes in CSS animations. to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

- Optimize for performance by minimizing CSS size.

## 33. What is transition in CSS?

### What is it?

What is transition in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is transition in CSS? in creating scalable, maintainable design systems.
2. Applied What is transition in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 34. Difference between animation and transition.

### What is it?

Difference between animation and transition. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Difference between animation and transition. in creating scalable, maintainable design systems.
2. Applied Difference between animation and transition. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 35. What is opacity in CSS?

### What is it?

What is opacity in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is opacity in CSS? in creating scalable, maintainable design systems.
2. Applied What is opacity in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 36. How do you apply background images with CSS?

### What is it?

How do you apply background images with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Real-world Scenarios

1. Used How do you apply background images with CSS? in creating scalable, maintainable design systems.
2. Applied How do you apply background images with CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 37. How do you create gradients with CSS?

### What is it?

How do you create gradients with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used How do you create gradients with CSS? in creating scalable, maintainable design systems.
2. Applied How do you create gradients with CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 38. What are CSS shadows and how do you apply them?

### What is it?

What are CSS shadows and how do you apply them? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are CSS shadows and how do you apply them? in creating scalable, maintainable design systems.
2. Applied What are CSS shadows and how do you apply them? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 39. Difference between inline, inline-block and block elements.

### What is it?

Difference between inline, inline-block and block elements. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Difference between inline, inline-block and block elements. in creating scalable, maintainable design systems.
2. Applied Difference between inline, inline-block and block elements. to build complex web application interfaces with responsive behavior.

### End-to-End Solution



# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 40. How do you hide elements in CSS?

### What is it?

How do you hide elements in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you hide elements in CSS? in creating scalable, maintainable design systems.
2. Applied How do you hide elements in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## 41. Explain the concept of responsive typography.

### What is it?

Explain the concept of responsive typography. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain the concept of responsive typography. in creating scalable, maintainable design systems.
2. Applied Explain the concept of responsive typography. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 42. What is a media feature in media queries?

### What is it?

What is a media feature in media queries? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is a media feature in media queries? in creating scalable, maintainable design systems.
2. Applied What is a media feature in media queries? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 43. How do you use CSS frameworks?

### What is it?

How do you use CSS frameworks? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you use CSS frameworks? in creating scalable, maintainable design systems.
2. Applied How do you use CSS frameworks? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 44. Explain the concept of a reset or normalize CSS.

### What is it?

Explain the concept of a reset or normalize CSS. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Real-world Scenarios

1. Used Explain the concept of a reset or normalize CSS. in creating scalable, maintainable design systems.
2. Applied Explain the concept of a reset or normalize CSS. to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 45. What are CSS pre-processors?

### What is it?

What are CSS pre-processors? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used What are CSS pre-processors? in creating scalable, maintainable design systems.
2. Applied What are CSS pre-processors? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 46. Difference between SCSS and SASS.

### What is it?

Difference between SCSS and SASS. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Difference between SCSS and SASS. in creating scalable, maintainable design systems.
2. Applied Difference between SCSS and SASS. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 47. What is a mixin in SASS?

### What is it?

What is a mixin in SASS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is a mixin in SASS? in creating scalable, maintainable design systems.
2. Applied What is a mixin in SASS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
font-family: Arial, sans-serif;
background-color: #f8f9fa;
margin: 0;
padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 48. Explain nesting in SASS.

### What is it?

Explain nesting in SASS. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain nesting in SASS. in creating scalable, maintainable design systems.
2. Applied Explain nesting in SASS. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 49. How does extend/inheritance work in SASS?

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## What is it?

How does extend/inheritance work in SASS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used How does extend/inheritance work in SASS? in creating scalable, maintainable design systems.
2. Applied How does extend/inheritance work in SASS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 50. What is a CSS sprite?

### What is it?

What is a CSS sprite? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is a CSS sprite? in creating scalable, maintainable design systems.
2. Applied What is a CSS sprite? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
margin: 0 auto;
padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 51. How do you optimize CSS for performance?

### What is it?

How do you optimize CSS for performance? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you optimize CSS for performance? in creating scalable, maintainable design systems.
2. Applied How do you optimize CSS for performance? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 52. What is critical CSS and why is it important?

### What is it?

What is critical CSS and why is it important? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is critical CSS and why is it important? in creating scalable, maintainable design systems.



# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

2. Applied What is critical CSS and why is it important? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 53. How do you minify CSS?

### What is it?

How do you minify CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you minify CSS? in creating scalable, maintainable design systems.
2. Applied How do you minify CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 54. What is BEM (Block Element Modifier) methodology?

### What is it?

What is BEM (Block Element Modifier) methodology? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is BEM (Block Element Modifier) methodology? in creating scalable, maintainable design systems.
2. Applied What is BEM (Block Element Modifier) methodology? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 55. Explain OOCSS (Object Oriented CSS).

### What is it?

Explain OOCSS (Object Oriented CSS). is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain OOCSS (Object Oriented CSS). in creating scalable, maintainable design systems.
2. Applied Explain OOCSS (Object Oriented CSS). to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
font-family: Arial, sans-serif;
background-color: #f8f9fa;
margin: 0;
padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 56. What is SMACSS (Scalable and Modular Architecture for CSS)?

### What is it?

What is SMACSS (Scalable and Modular Architecture for CSS)? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is SMACSS (Scalable and Modular Architecture for CSS)? in creating scalable, maintainable design systems.
2. Applied What is SMACSS (Scalable and Modular Architecture for CSS)? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## 57. Difference between inline styles and external stylesheets.

### What is it?

Difference between inline styles and external stylesheets. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Difference between inline styles and external stylesheets. in creating scalable, maintainable design systems.
2. Applied Difference between inline styles and external stylesheets. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 58. How do you prevent style conflicts?

### What is it?

How do you prevent style conflicts? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you prevent style conflicts? in creating scalable, maintainable design systems.
2. Applied How do you prevent style conflicts? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 59. How do you override CSS rules?

### What is it?

How do you override CSS rules? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you override CSS rules? in creating scalable, maintainable design systems.
2. Applied How do you override CSS rules? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 60. What is important declaration in CSS?

### What is it?

What is important declaration in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

1. Used What is important declaration in CSS? in creating scalable, maintainable design systems.
2. Applied What is important declaration in CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 61. How to create a sticky footer with CSS?

### What is it?

How to create a sticky footer with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How to create a sticky footer with CSS? in creating scalable, maintainable design systems.
2. Applied How to create a sticky footer with CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 62. What is vh, vw units in CSS?

### What is it?

What is vh, vw units in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is vh, vw units in CSS? in creating scalable, maintainable design systems.
2. Applied What is vh, vw units in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 63. Explain aspect-ratio in CSS.

### What is it?

Explain aspect-ratio in CSS. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain aspect-ratio in CSS. in creating scalable, maintainable design systems.
2. Applied Explain aspect-ratio in CSS. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
margin: 0;
padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 64. How to style forms and inputs with CSS?

### What is it?

How to style forms and inputs with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How to style forms and inputs with CSS? in creating scalable, maintainable design systems.
2. Applied How to style forms and inputs with CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 65. What is :nth-child and :nth-of-type selector?



# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## What is it?

What is :nth-child and :nth-of-type selector? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used What is :nth-child and :nth-of-type selector? in creating scalable, maintainable design systems.
2. Applied What is :nth-child and :nth-of-type selector? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 66. Explain attribute selectors in CSS.

### What is it?

Explain attribute selectors in CSS. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain attribute selectors in CSS. in creating scalable, maintainable design systems.
2. Applied Explain attribute selectors in CSS. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
margin: 0 auto;
padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 67. How do you create tooltips with CSS?

### What is it?

How do you create tooltips with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you create tooltips with CSS? in creating scalable, maintainable design systems.
2. Applied How do you create tooltips with CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 68. How do you create dropdown menus with CSS?

### What is it?

How do you create dropdown menus with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you create dropdown menus with CSS? in creating scalable, maintainable design systems.
2. Applied How do you create dropdown menus with CSS? to build complex web application interfaces with responsive

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 69. How to use hover effects effectively?

### What is it?

How to use hover effects effectively? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How to use hover effects effectively? in creating scalable, maintainable design systems.
2. Applied How to use hover effects effectively? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 70. Explain the difference between max-width and min-width.

### What is it?

Explain the difference between max-width and min-width. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain the difference between max-width and min-width. in creating scalable, maintainable design systems.
2. Applied Explain the difference between max-width and min-width. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 71. What is clamp() function in CSS?

### What is it?

What is clamp() function in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is clamp() function in CSS? in creating scalable, maintainable design systems.
2. Applied What is clamp() function in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
margin: 0;
padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 72. How do you use CSS filter effects?

### What is it?

How do you use CSS filter effects? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you use CSS filter effects? in creating scalable, maintainable design systems.
2. Applied How do you use CSS filter effects? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 73. Explain backdrop-filter property in CSS.

### What is it?

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

Explain backdrop-filter property in CSS. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used Explain backdrop-filter property in CSS. in creating scalable, maintainable design systems.
2. Applied Explain backdrop-filter property in CSS. to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 74. How do you create responsive images with CSS?

### What is it?

How do you create responsive images with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used How do you create responsive images with CSS? in creating scalable, maintainable design systems.
2. Applied How do you create responsive images with CSS? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 75. What is a media object pattern in CSS?

### What is it?

What is a media object pattern in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is a media object pattern in CSS? in creating scalable, maintainable design systems.
2. Applied What is a media object pattern in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 76. How do you debug CSS issues?

### What is it?

How do you debug CSS issues? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you debug CSS issues? in creating scalable, maintainable design systems.
2. Applied How do you debug CSS issues? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 77. What are best practices for writing maintainable CSS?

### What is it?

What are best practices for writing maintainable CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are best practices for writing maintainable CSS? in creating scalable, maintainable design systems.
2. Applied What are best practices for writing maintainable CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.



# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

- Optimize for performance by minimizing CSS size.

## 78. Explain layered background images.

### What is it?

Explain layered background images. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain layered background images. in creating scalable, maintainable design systems.
2. Applied Explain layered background images. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 79. How do you create parallax effects with CSS?

### What is it?

How do you create parallax effects with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you create parallax effects with CSS? in creating scalable, maintainable design systems.
2. Applied How do you create parallax effects with CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 80. What is grid-template-areas in CSS Grid?

### What is it?

What is grid-template-areas in CSS Grid? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is grid-template-areas in CSS Grid? in creating scalable, maintainable design systems.
2. Applied What is grid-template-areas in CSS Grid? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 81. How do you use object-fit and object-position?

### What is it?

How do you use object-fit and object-position? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Real-world Scenarios

1. Used How do you use object-fit and object-position? in creating scalable, maintainable design systems.
2. Applied How do you use object-fit and object-position? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 82. How do you create custom scrollbars?

### What is it?

How do you create custom scrollbars? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used How do you create custom scrollbars? in creating scalable, maintainable design systems.
2. Applied How do you create custom scrollbars? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 83. What are web fonts and how do you use them?

### What is it?

What are web fonts and how do you use them? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are web fonts and how do you use them? in creating scalable, maintainable design systems.
2. Applied What are web fonts and how do you use them? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 84. Explain @font-face rule in CSS.

### What is it?

Explain @font-face rule in CSS. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain @font-face rule in CSS. in creating scalable, maintainable design systems.
2. Applied Explain @font-face rule in CSS. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 85. What is the difference between serif and sans-serif fonts?

### What is it?

What is the difference between serif and sans-serif fonts? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is the difference between serif and sans-serif fonts? in creating scalable, maintainable design systems.
2. Applied What is the difference between serif and sans-serif fonts? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## 86. How do you preload fonts for performance?

### What is it?

How do you preload fonts for performance? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you preload fonts for performance? in creating scalable, maintainable design systems.
2. Applied How do you preload fonts for performance? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 87. What is variable font in CSS?

### What is it?

What is variable font in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is variable font in CSS? in creating scalable, maintainable design systems.
2. Applied What is variable font in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 88. What are pseudo-elements used for decorative purposes?

### What is it?

What are pseudo-elements used for decorative purposes? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What are pseudo-elements used for decorative purposes? in creating scalable, maintainable design systems.
2. Applied What are pseudo-elements used for decorative purposes? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 89. Explain counter-increment and counter-reset properties.

### What is it?

Explain counter-increment and counter-reset properties. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Real-world Scenarios

1. Used Explain counter-increment and counter-reset properties. in creating scalable, maintainable design systems.
2. Applied Explain counter-increment and counter-reset properties. to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 90. How do you use CSS clip-path?

### What is it?

How do you use CSS clip-path? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

## Real-world Scenarios

1. Used How do you use CSS clip-path? in creating scalable, maintainable design systems.
2. Applied How do you use CSS clip-path? to build complex web application interfaces with responsive behavior.

## End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```



# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 91. What is calc() function in CSS?

### What is it?

What is calc() function in CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is calc() function in CSS? in creating scalable, maintainable design systems.
2. Applied What is calc() function in CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 92. How do you implement dark mode with CSS?

### What is it?

How do you implement dark mode with CSS? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you implement dark mode with CSS? in creating scalable, maintainable design systems.
2. Applied How do you implement dark mode with CSS? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 93. What is prefers-color-scheme media query?

### What is it?

What is prefers-color-scheme media query? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is prefers-color-scheme media query? in creating scalable, maintainable design systems.
2. Applied What is prefers-color-scheme media query? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

## 94. How do you create responsive cards using Flexbox?

### What is it?

How do you create responsive cards using Flexbox? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used How do you create responsive cards using Flexbox? in creating scalable, maintainable design systems.
2. Applied How do you create responsive cards using Flexbox? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}
```

### Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 95. What is :not() selector used for?

### What is it?

What is :not() selector used for? is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used What is :not() selector used for? in creating scalable, maintainable design systems.
2. Applied What is :not() selector used for? to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */
body {
  font-family: Arial, sans-serif;
  background-color: #f8f9fa;
  margin: 0;
  padding: 0;
}
```

# Drupal 9/10 & Drupal 10+ Exam Preparation Guide

```
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.

## 96. Explain differences between CSS Grid and Flexbox in practical usage.

### What is it?

Explain differences between CSS Grid and Flexbox in practical usage. is a critical skill in modern frontend development, enabling developers to style and layout websites responsively and accessibly.

### Real-world Scenarios

1. Used Explain differences between CSS Grid and Flexbox in practical usage. in creating scalable, maintainable design systems.
2. Applied Explain differences between CSS Grid and Flexbox in practical usage. to build complex web application interfaces with responsive behavior.

### End-to-End Solution

```
/* Example CSS rule */  
body {  
  font-family: Arial, sans-serif;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 0;  
}  
  
.container {  
  max-width: 1200px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

## Best Practices

- Use semantic class names.
- Avoid using !important excessively.
- Minimize specificity conflicts.
- Organize CSS using methodologies like BEM.
- Optimize for performance by minimizing CSS size.