10 Examples HTML CSS Web Design Guide

CSS Centering:

HTML:

```
<html>
  <body>
    <div class="centered-div">
      Centered Text
    </div>
```

```
  </body>
</html>
```

```
CSS:
.centered-div {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
```

Explanation: This code centers the text inside a div both vertically and horizontally by using display: flex;, justify-content: center;, and align-items: center;. The height: 100vh; ensures that the div takes up the full height of the viewport.

CSS Gradient Background:

```
HTML:
<html>
  <body>
    <div class="gradient-div">
      Gradient Background
    </div>
```

```
  </body>
</html>
```

CSS:

```
.gradient-div {
  background: linear-gradient(to bottom right, #ff416c, #ff4b2b);
  height: 100vh;
}
```

Explanation: This code creates a gradient background on a div using the background property with the linear-gradient value. The to bottom right specifies the direction of the gradient. The hex codes represent the start and end colors of the gradient.

CSS Image Hover Effect:

HTML:

```
<html>
  <body>
    <div class="image-container">
      <img src="image.jpg" alt="Image">
      <div class="overlay">
        <div class="overlay-text">Overlay Text</div>
```

```
      </div>
    </div>
  </body>
</html>
```

CSS:

```
.image-container {
  position: relative;
}

.overlay {
  position: absolute;
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  opacity: 0;
  transition: .5s ease;
  background-color: rgba(0,0,0,0.5);
}

.image-container:hover .overlay {
  opacity: 1;
}
```

```
.overlay-text {
  color: white;
  font-size: 20px;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  text-align: center;
}
```

Explanation: This code creates an overlay effect on an image when the mouse hovers over it. The .image-container div is set to position: relative; to allow the overlay to be positioned absolutely within it. The .overlay div is initially hidden with opacity: 0; and transitions to an opacity of 1 when the .image-container div is hovered over. The rgba(0,0,0,0.5) background-color creates a semi-transparent black overlay. The .overlay-text is positioned absolutely in the center of the image and is only displayed when the overlay is visible.

CSS Flexbox Layout:

HTML:

```
<html>
  <body>
    <div class="flex-container">
      <div class="flex-item">Item 1</div>
      <div class="flex-item">Item 2</div>
      <div class="flex-item">Item 3</div>
    </div>
  </body>
</html>
```

CSS:

```
.flex-container {
  display: flex;
  justify-content: space-around;
  align-items: center;
  height: 100vh;
}

.flex-item {
  background-color: #4CAF50;
  color: white;
  padding: 20px;
  font-size: 20px;
}
```

CSS Responsive Layout:

HTML:

```
<html>
  <body>
    <div class="responsive-container">
      <div class="responsive-item">Item 1</div>
      <div class="responsive-item">Item 2</div>
      <div class="responsive-item">Item 3</div>
    </div>
  </body>
</html>
```

CSS:

```
.responsive-container {
  display: flex;
  flex-wrap: wrap;
}

.responsive-item {
  flex-basis: calc(33.33% - 20px);
  margin: 10px;
  background-color: #4CAF50;
  color: white;
```

```
  padding: 20px;
  font-size: 20px;
}
```

CSS Box Shadow:

HTML:

```
<html>
  <body>
    <div class="box-shadow-div">
      Box Shadow
    </div>
  </body>
</html>
```

CSS:

```
.box-shadow-div {
  width: 200px;
  height: 200px;
  background-color: white;
  box-shadow: 0px 0px 10px 5px rgba(0, 0, 0, 0.5);
}
```

CSS Hover Transition:

HTML:

```
<html>
  <body>
    <button class="hover-transition-btn">Hover
Me</button>
  </body>
</html>
```

CSS:

```
.hover-transition-btn {
  background-color: #4CAF50;
  color: white;
  border: none;
  padding: 10px 20px;
  font-size: 16px;
  transition: background-color 0.5s ease;
}

.hover-transition-btn:hover {
  background-color: #3e8e41;
}
```

CSS Dropdown Menu:

HTML:

```
<html>
  <body>
    <div class="dropdown-menu">
      <button class="dropbtn">Dropdown</button>
      <div class="dropdown-content">
        <a href="#">Link 1</a>
        <a href="#">Link 2</a>
        <a href="#">Link 3</a>
      </div>
    </div>
  </body>
</html>
```

CSS:

```
.dropbtn {
  background-color: #4CAF50;
  color: white;
  padding: 10px;
  font-size: 16px;
  border: none;
}
```

```
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 10px;
  text-decoration: none;
  display: block;
}

.dropdown:hover .dropdown-content {
  display: block;
}
```

CSS Sticky Header:

HTML:

```
<html>
  <body>
    <div class="header">
      <h1>My Website</h1>
      <p>A sticky header</p>
    </div>
    <div class="content">
      <h2>Content</h2>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
    </div>
  </body>
</html>
```

CSS:

```
.header {
  background-color: #f1f1f1;
  padding: 20px;
  position: sticky;
  top: 0;
}

.content {
```

```
  padding: 20px;
}
```

CSS Grid Layout:

HTML:

```
<html>
  <body>
    <div class="grid-container">
      <div class="item1">1</div>
      <div class="item2">2</div>
      <div class="item3">3</div>
      <div class="item4">4</div>
      <div class="item5">5</div>
      <div class="item6">6</div>
      <div class="item7">7</div>
      <div class="item8">8</div>
      <div class="item9">9</div>
    </div>
  </body>
</html>
```

CSS:

```
.grid-container {
  display: grid;
```

```
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: repeat(3, 100px);
  grid-gap: 10px;
}

.grid-container > div {
  background-color: #2196F3;
  color: white;
  font-size: 30px;
  display: flex;
  justify-content: center;
  align-items: center;
}

.item1 {
  grid-row: 1 / 3;
  grid-column: 1 / 3;
}

.item2 {
  grid-row: 1 / 2;
  grid-column: 3 / 4;
}
```

```
.item3 {
  grid-row: 2 / 3;
  grid-column: 3 / 4;
}

.item4 {
  grid-row: 3 / 4;
  grid-column: 1 / 2;
}

.item5 {
  grid-row: 3 / 4;
  grid-column: 2 / 3;
}

.item6 {
  grid-row: 3 / 4;
  grid-column: 3 / 4;
}

.item7 {
  grid-row: 1 / 2;
  grid-column: 2 / 3;
}
```

```
.item8 {
  grid-row: 2 / 3;
  grid-column: 2 / 3;
}

.item9 {
  grid-row: 1 / 2;
  grid-column: 3 / 4;
}
```

In this example, we have a container with the class grid-container that contains 9 child elements with classes item1 through item9. The grid-template-columns and grid-template-rows properties define the number and size of the columns and rows in the grid. The grid-gap property sets the spacing between grid items. The child elements are positioned within the grid using the grid-row and grid-column properties.