

# Media Query

## Assignment

**Q1. Create a simple webpage with a header, a main content section, and a footer. Apply a media query that changes the background color of the header to blue when the screen width is less than 600px.**

**Ans:-** <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Responsive Header</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<header><h1>Responsive Header</h1></header>

<main>Main Content</main>

<footer>Footer</footer>

</body>

</html>

**Style.css**

```
*{  
  margin: 0;
```

```
    box-sizing: border-box;
    padding: 0;
}

header{
    padding: 20px;
    background-color: #dddddd;
    font-size: 1em;
    text-align: center;
}
```

```
@media screen and (max-width: 600px) {
    header{
        background-color: #333333;
        color: #ffffff;
    }
}
```

**Q2. Create an image gallery with three images in a row. Use media queries to adjust the layout to two images in a row for screens smaller than 800px and one image in a row for screens smaller than 500px.**

**Ans:-** <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

```
<meta name="viewport" content="width=device-width, initial-
scale=1.0" />

<title>Document</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<div class="image-gallery">





</div>

</body>

</html>
```

## **Style.css**

```
.image-gallery {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}

.image-gallery img {
  width: 30%;
  margin: 10px;
}

@media screen and (max-width: 800px) {
  .image-gallery img {
```

```
width: 45%;  
}  
}  
@media screen and (max-width: 500px) {  
  .image-gallery img {  
    width: 100%;  
  }  
}
```

**Q3. Create a navigation bar with five links. Apply a media query that converts the navigation into a vertical list for screens smaller than 768px and also changes the background color to gray.**

**Ans:-** <!DOCTYPE html>

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

```
  <title>Responsive Nav</title>
```

```
  <link rel="stylesheet" href="./style.css">
```

```
</head>
```

```
<body>
```

```
  <nav>
```

```
    <ul>
```

```
      <li>Home</li>
```

```
      <li>About</li>
```

```
      <li>Services</li>
```

```
        <li>Portfolio</li>
        <li>Contact</li>
    </ul>
</nav>
</body>
</html>
```

### **Style.css:-**

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

nav{
    background-color: #212121;
    color: #ffffffde;
    font-size: 1.2em;
    padding: 20px;
}

nav ul{
    display: flex;
    justify-content: space-around;
}
```

```
nav ul li{  
    list-style: none;  
}
```

```
@media screen and (max-width: 768px){
```

```
    nav{  
        background-color: #757575;  
    }
```

```
    nav ul{  
        flex-direction: column;  
        align-items: center;  
    }
```

```
    nav ul li{  
        margin-block-end: 1.2em;  
    }
```

```
}
```

**Q4. Create a grid based layout containing four grid items which arranges them as the screen size changes.**

**Ans:-** <!DOCTYPE html>

<html lang="en">

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Responsive Grid</title>
  <link rel="stylesheet" href="./style.css">
</head>
<body>
  <div class="container">
    <div class="box">Box1</div>
    <div class="box">Box2</div>
    <div class="box">Box3</div>
    <div class="box">Box4</div>
  </div>
</body>
</html>
```

### **Style.css**

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

.container{
```

```
display: grid;
justify-content: center;
width: 100%;
grid-template-columns: repeat(3, 1fr);
grid-auto-rows: 200px;
gap: 10px;
}
```

```
.box{
    text-align: center;
}
```

```
.container :nth-child(1){
    background-color: red;
}
.container :nth-child(2){
    background-color: green;
}
.container :nth-child(3){
    background-color: blue;
}
.container :nth-child(4){
    background-color: orange;
}
```



```
@media screen and (max-width: 800px) {  
  .container{  
    grid-template-columns: repeat(2, 1fr);  
  }  
}
```

```
@media screen and (max-width: 500px) {  
  .container{  
    grid-template-columns: repeat(1, 1fr);  
  }  
}
```

**Q5. Create the below given layout using the flexbox in css, which should adapt itself on mobile screen as per given.**

**Ans:-** <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Flex Responsive</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<div class="container">

<div class="box1">Box1</div>

```
<div class="main">
  <div class="box2">Box2</div>
  <div class="box3">Box3</div>
</div>
<div class="box4">Box4</div>
</div>
</body>
</html>
```

### **Style.css**

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

.container {
  display: flex;
  background-color: #eeeeee;
  min-height: 100vh;
}

.main {
  flex-basis: 90%;
  display: flex;
  flex-direction: column;
```

```
}
```

```
.box1 {  
    flex-basis: 5%;  
    background-color: #212121;  
    color: #ffffff;  
    text-align: center;  
}
```

```
.box2 {  
    flex-basis: 50%;  
    text-align: center;  
    background-color: rgb(37, 194, 152);  
}
```

```
.box3 {  
    flex-basis: 50%;  
    text-align: center;  
    background-color: rgb(246, 164, 11);  
}
```

```
.box4 {  
    flex-basis: 5%;  
    background-color: #212121;  
    color: #ffffff;
```

```
    text-align: center;
}
```

```
.box1, .box2, .box3, .box4 {
    padding: 20px;
}
```

```
@media screen and (max-width: 768px) {
    .container{
        flex-direction: column;
    }
}
```

```
.box4{
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
}
}
```

**COMPLETE**