

# Chapter 07

## Introduction to **Web Design**



## **Responsive Web Design**

# Content

---

- Introduction to responsive web design
- Media Queries
- CSS Min, Max, Clamp method

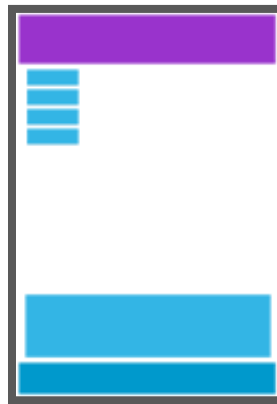
# Introduction to responsive web design

## ☐ Responsive Web Design

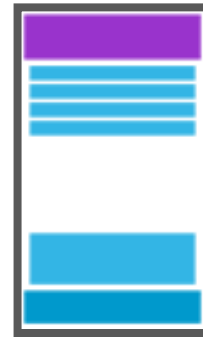
Responsive web design makes your web page look good on all devices



Desktop



Tablet/iPad



Phone



Designing for the best Experience  
for all Users' using device platforms



# Media Queries

## ❑ What is @media ?

This let you target specific screen sizes and specify code to be run on the devices alone.

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

### ▪ Media Types

all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.

### ▪ Expressions: Width and Height Conditions

min-width	if the viewport width is equal to or wider than the specified value
max-width	if the viewport width is equal to or narrower than the specified value
min-height	if the viewport height is equal to or taller than the specified value
max-height	if the viewport height is equal to or shorter than the specified value

- “not” Keyword: apply styles when a specific condition is not met
- “only” keyword is used to ensure that styles are applied only if the specified condition is met

# Media Queries

## □ Examples of applying responsive styles

```
body {  
  background-color: pink;  
}  
  
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}  
  
@media screen and (min-width: 820px) {  
  body {  
    background-color: yellow;  
  }  
}
```

- default style

**Resize the browser window to see the effect!**

The media query will only apply if the media type is screen and the viewport is 480px wide or wider.

- min-width: 480px

**Resize the browser window to see the effect!**

The media query will only apply if the media type is screen and the viewport is 480px wide or wider.

- min-width: 820px

**Resize the browser window to see the effect!**

The media query will only apply if the media type is screen and the viewport is 480px wide or wider.

# Media Queries

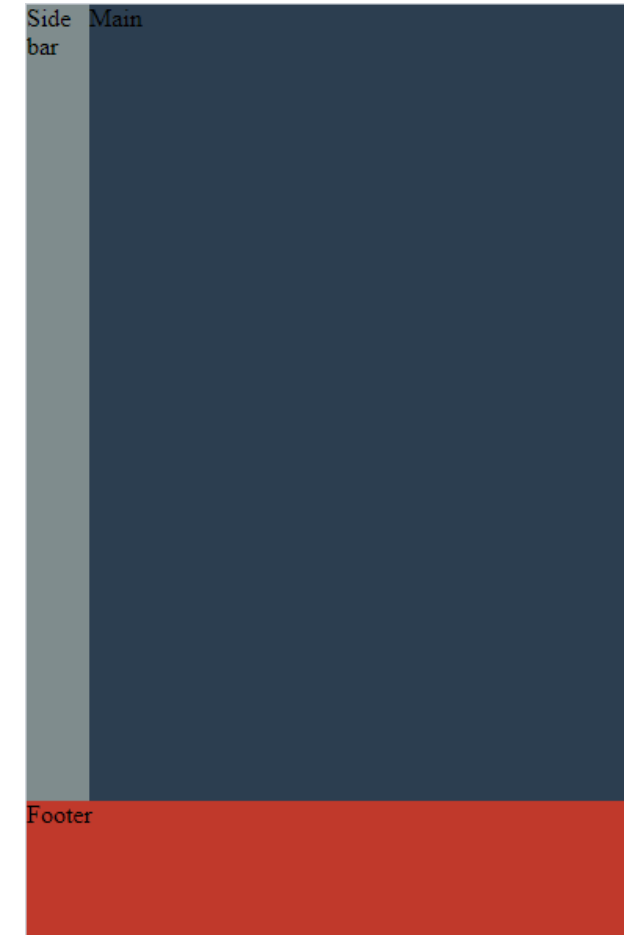
## ❑ Redefining Grid Areas with Media Queries

```
body {  
  display: grid;  
  min-height: 100vh;  
  min-width: 100%;  
  grid-template-rows: 1fr 90px;  
  grid-template-columns: 1fr;  
  grid-template-areas: "content"  
                      "footer"  
}  
  
@media only screen and (min-width: 600px) {  
  body {  
    grid-template-columns: 120px 1fr;  
    grid-template-areas:  
      "sidebar content"  
      "footer footer";  
  }  
}
```

### ■ default



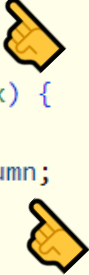
### ■ min-width: 600px



# Media Queries

## ❑ Redefining **Flex** with Media Queries

```
.flex-container {  
  display: flex;  
  flex-direction: row;  
}  
  
@media (max-width: 400px) {  
  .flex-container {  
    flex-direction: column;  
  }  
}
```



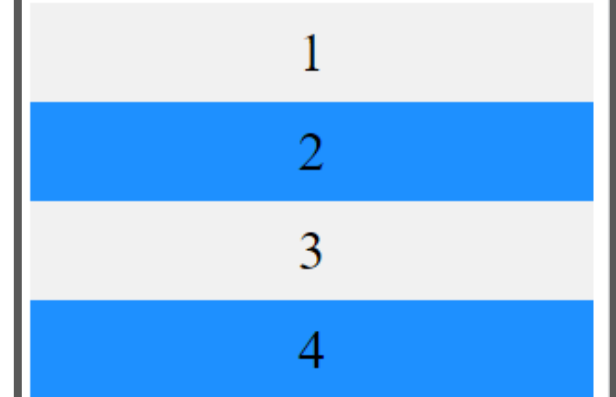
### ▪ default

#### Responsive Flexbox



### ▪ max-width: 400px

#### Responsive Flexbox



# Media Queries

## ❑ Redefining Grid with Media Queries

```
<html lang="en">
<head></head>
<body>
  <div class="container">
    <div class="elements">1</div>
    <div class="elements">2</div>
    <div class="elements">3</div>
    .
    .
    .
    <div class="elements">49</div>
    <div class="elements">50</div>
  </div>
</body>
</html>
```

HTML

```
.container{
  display: grid;
  grid-template-columns: repeat(5,1fr);
  place-items: center;
  gap: 20px 30px;
}

/* Mobile View */
@media screen and (min-width:250px) and (max-width:480px) {
  .container{
    grid-template-columns: repeat(1,1fr);
  }
}

/* Tablet View */
@media screen and (min-width:481px) and (max-width:768px) {
  .container{
    grid-template-columns: repeat(3,1fr);
  }
}

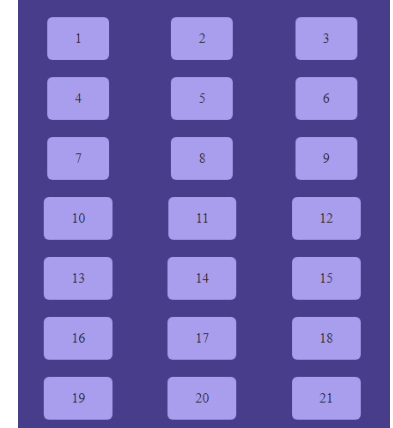
/* Laptop View */
@media screen and (min-width:769px) and (max-width:1024px) {
  .container{
    grid-template-columns: repeat(4,1fr);
  }
}
```

CSS

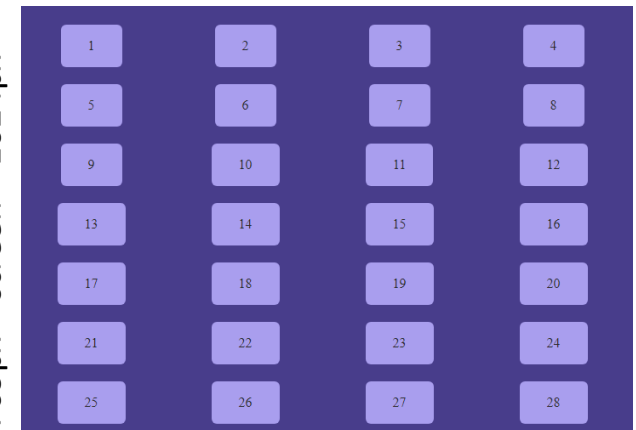
250px > screen < 480px



481px > screen < 768px



769px > screen < 1024px





# CSS Min, Max, Clamp function

## ❏ Min() Method

`min()` function in CSS is used to extract the minimum value from a set of comma-separated values

```
min(value1, value2);  
min(value1, min(value2, min(value3, value4)));
```

- It returns the smallest value among the expressions.
- It can be used to set values for properties like width, height, font-size, and more

```
width: min(300px, 50%);
```

```
<!DOCTYPE html>  
<html>  
  
<head>  
  <style>  
    #div1 {  
      background-color: yellow;  
      height: 100px;  
      width: min(50%, 300px);  
    }  
  </style>  
</head>  
<body>  
  <h1>The min() Function</h1>  
  <div id="div1">Some text...</div>  
</body>  
</html>
```



When 50% is smaller than 300px

When 300px is smaller than 50%

### The min() Function

Some text...

### The min() Function

Some text...

# CSS Min, Max, Clamp function

## ❑ Max() Method

The `max()` function in CSS is used to set the maximum of the numbers given. It is used to return the largest value from a set of comma-separated values

```
max(value1, value2...);
```

- This function returns the largest value from a set of comma-separated values.
- It can be used to set values for properties like `width`, `height`, `font-size`, and more

```
width: max(300px, 50%);
```

```
<head>
  <style>
    #div1 {
      background-color: yellow;
      height: 100px;
      width: max(50%, 300px);
    }
  </style>
</head>
<body>
  <h1>The max() Function</h1>
  <div id="div1">Some text...</div>
</body>
</html>
```



When 50% is bigger than 300px

### The max() Function

Some text...



### The max() Function

Some text...



When 300px is smaller than 50%

# CSS Min, Max, Clamp function

## ❑ Clamp() Method

The clamp() method is used to clamp the value between an upper and lower bound

```
clamp(minimumValue, preferredValue, maximumValue);
```

- **Preferred value:** becomes useful when it is between the minimum and maximum value
- **Minimum value:** comes in handy when the preferred value is smaller minimum value
- **Maximum value:** comes in handy when the preferred value is more than the maximum value

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1 {
      font-size: clamp(2rem, 4vw, 4rem);
      color: #eb4034;
    }
    .box {
      width: clamp(150px, 50%, 400px);
      height: 8rem;
      background: #5f76e8;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Welcome To GFG</h1>
    <div class="box"></div>
  </div>
</body>
</html>
```

### h1:

- Dynamic size 4vm
- Never smaller than 2rem
- Never bigger than 4rem

### box:

- Dynamic size 50%
- Never smaller than 150px
- Never bigger than 400px

Welcome To GFG



# Good luck 🍀

## References

1. <https://www.educative.io/> (The Complete Advanced Guide to CSS)
2. [https://www.w3schools.com/css/css3\\_flexbox\\_responsive.asp](https://www.w3schools.com/css/css3_flexbox_responsive.asp)
3. <https://www.geeksforgeeks.org/css-clamp-method/>
4. <https://www.geeksforgeeks.org/css-max-function/>
5. <https://www.geeksforgeeks.org/css-min-function/>