WEB DESIGN

W6 - RESPONSIVE DESIGN

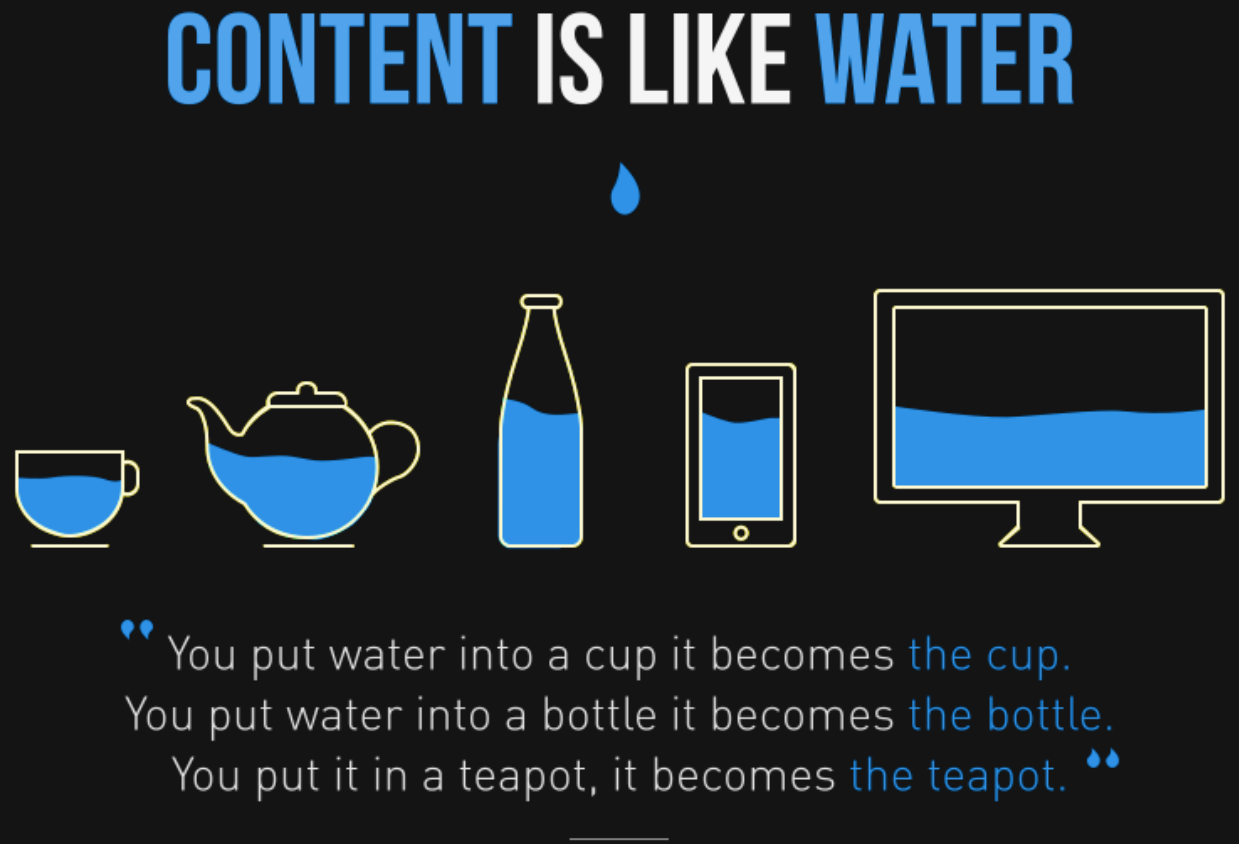
CADT
IDT

Course Objectives

- ✓ Be able to understand what is **RESPONSIVE DESIGN**
- ✓ Be able to use **MEDIA QUERIES**
- ✓ Understand the concept of **BREAK POINT**
- ✓ Understand **CLAMP FUNCTION**
- ✓ Select the right **CSS UNITS** for responsive design

Think of the website's content like **water**.

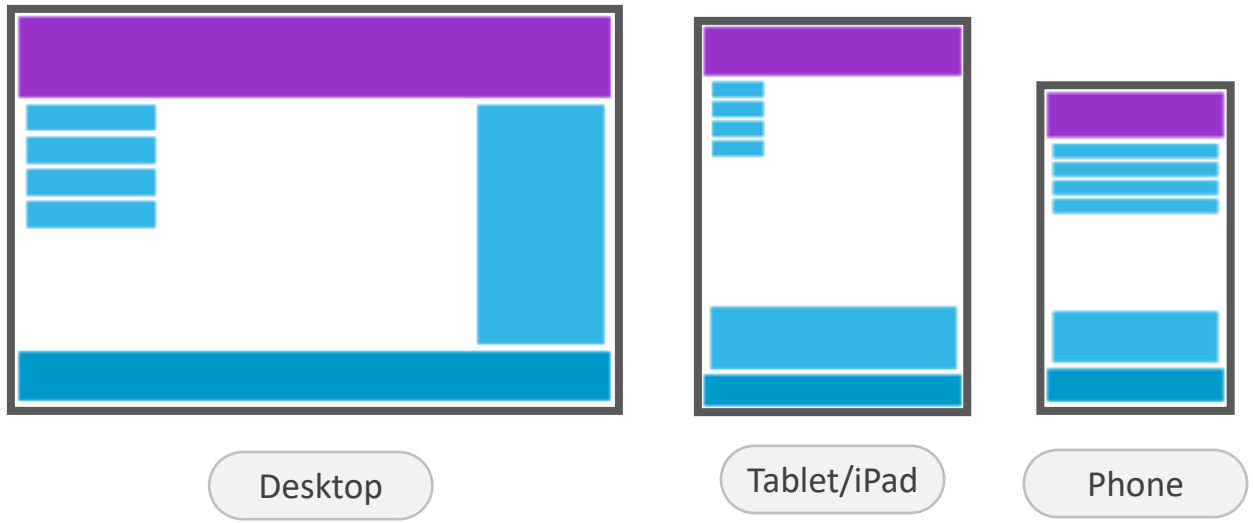
we design the website so its content will fit well in all kinds and sizes of **containers**



EXPLAIN

What is Responsive Design?

Responsive Design aims to design a websites that adapt to different screen sizes.



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

EXPLAIN

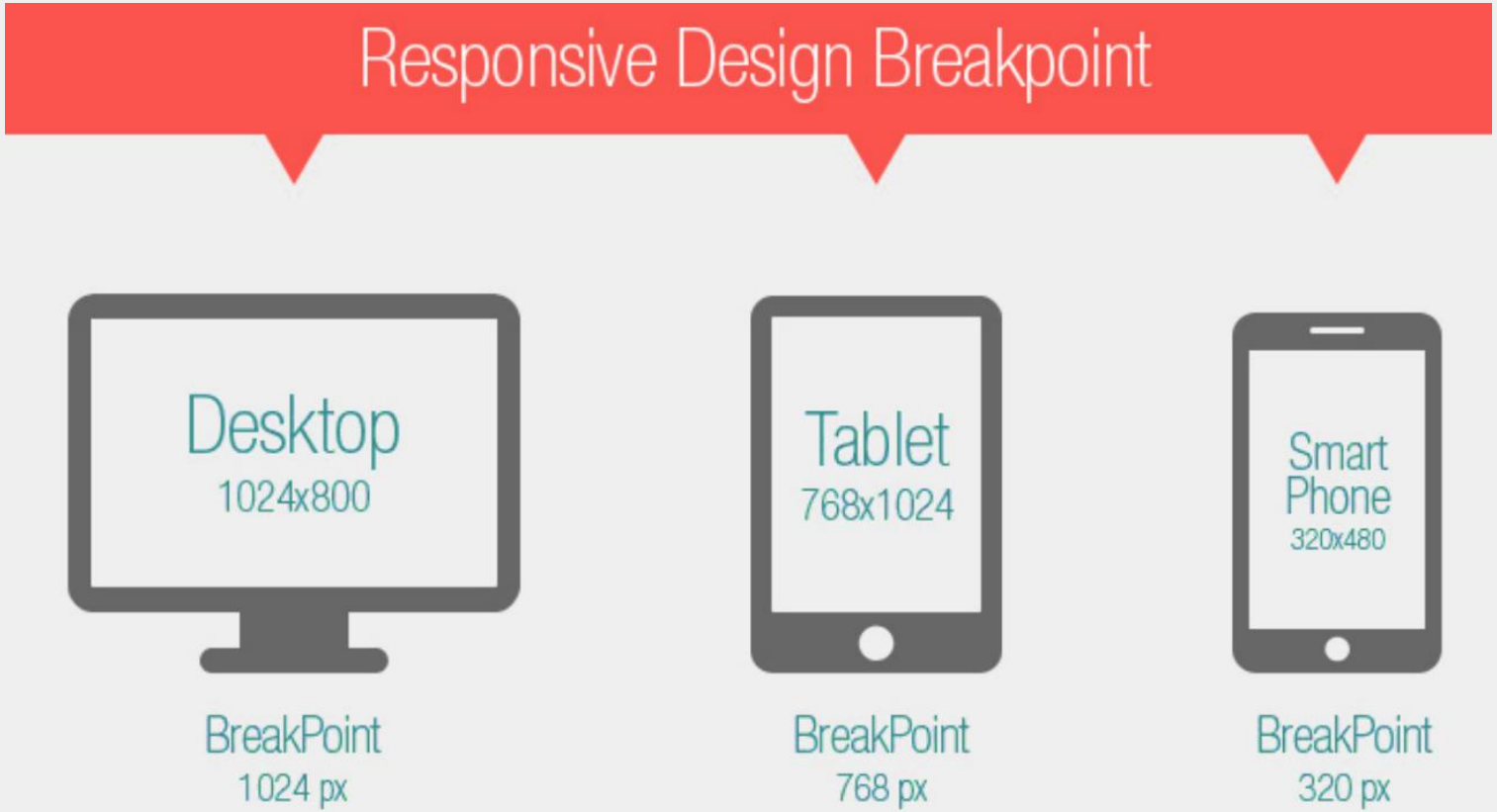
RESPONSIVE DESIGN
ensures that a website adapts to different screen sizes

NON-RESPONSIVE DESIGN
doesn't adapt to different screen sizes or devices.

EXPLAIN

Breakpoint

A breakpoint is a screen width at which a **website changes** its layout or styling to stay responsive.



EXPLAIN

What is @media ?

A CSS rule that applies styles only when specific device or screen conditions (like width) are met.

```
@media (min-width: 600px) and (max-width: 900px) {
      body {
        background-color: lightblue;
      },
}
```

- **Media Types**

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

- **Condition composition**

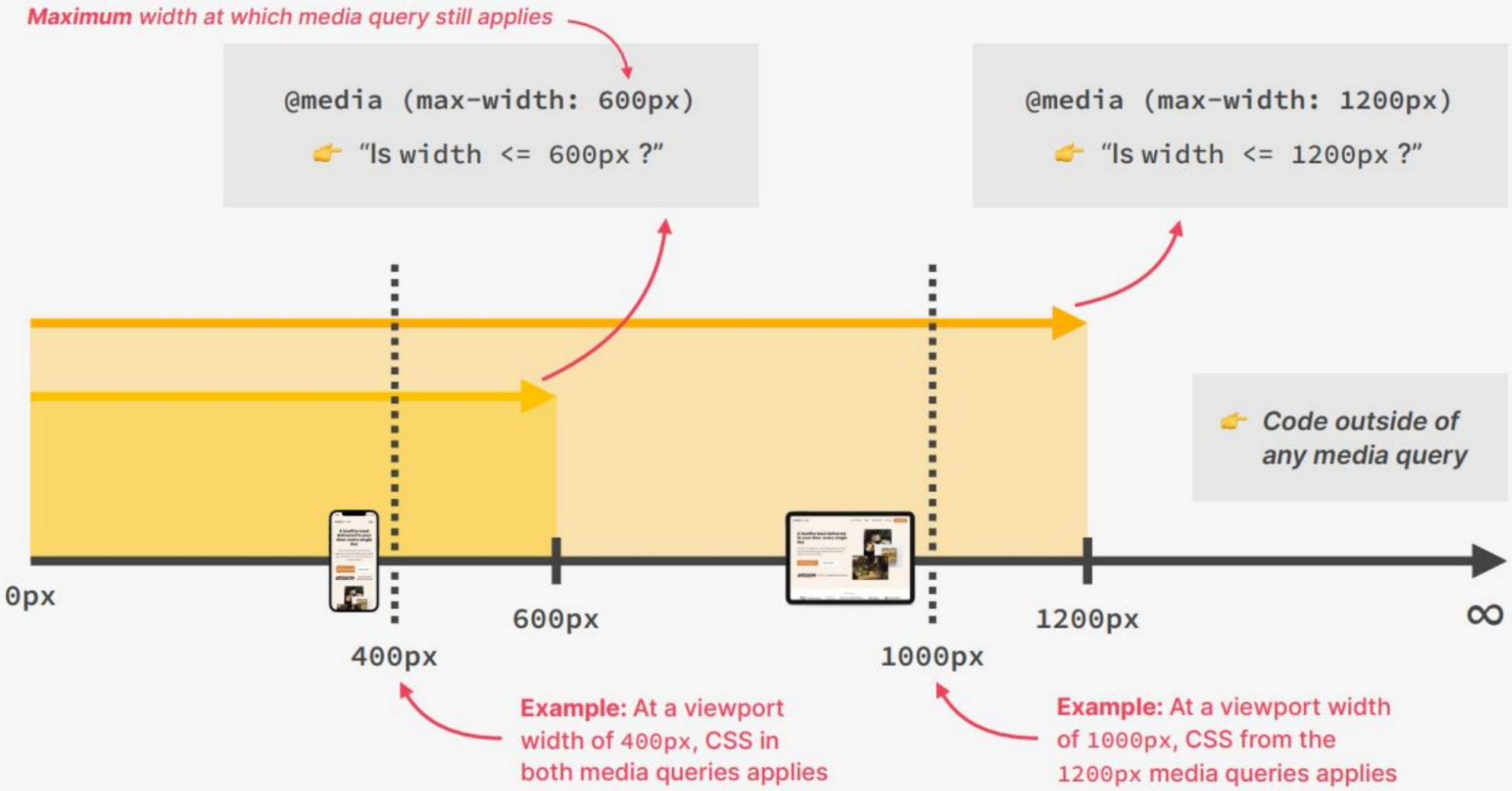
not	Apply style ONLY if the condition is NOT met
only	Apply style ONLY if the condition is met

- **Expressions: Width and Height Conditions**

min-width	viewport width >= specified value
max-width	viewport width <= specified value
min-height	viewport height >= specified value
max-height	viewport height <= specified value

EXPLAIN

How Media Queries Work?

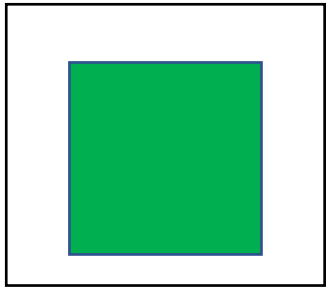


ACTIVITY

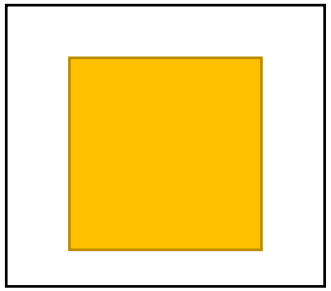
Activity 1

✓ Change colors according to screen width

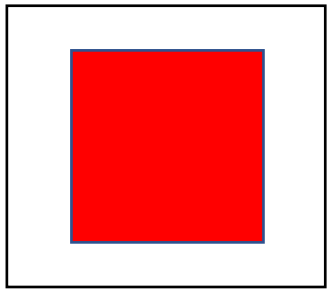
By default



Breakpoint 768px



Breakpoint 414 px



ANSWER

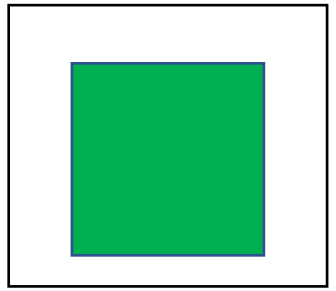
Activity 1

START CODE

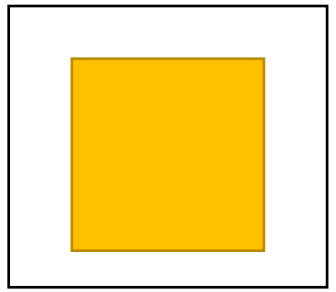
✓ Change colors according to screen width

```
.box {
    width: 100%;
    height: 200px;
    background: green;        /* Default */
}

/* Breakpoint ≤ 768px */
@media (max-width: 768px) {
    .box {
        background: yellow;
    }
}

/* Breakpoint ≤ 414px */
@media (max-width: 414px) {
    .box {
        background: red;
    }
}
```

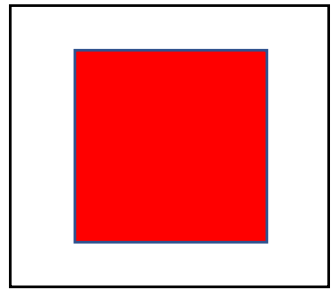
By default



Breakpoint 768px



Breakpoint 414 px

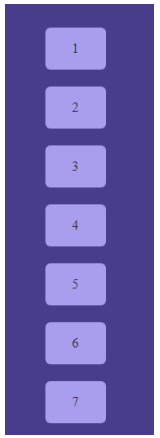


EXPLAIN

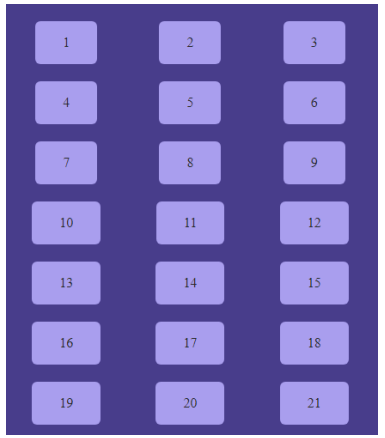
Media Queries with **Grids**

```css
.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);
  }
}
```

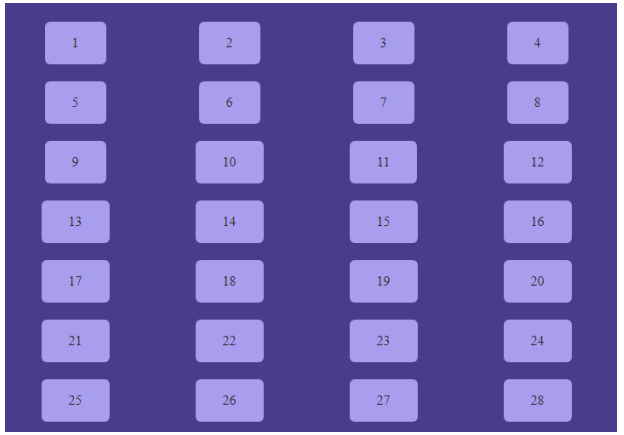
250px > screen < 480px



481px > screen < 768px



769px > screen < 1024px



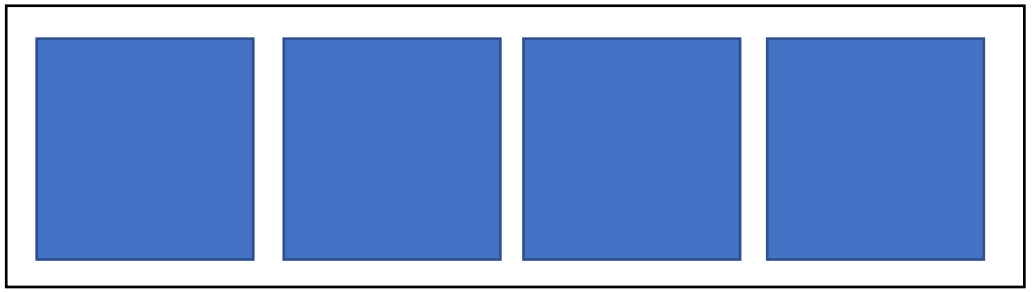
ACTIVITY

Activity 2

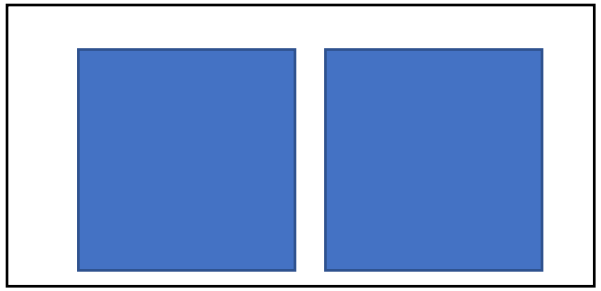
START CODE

- ✓ Reduce the number of columns to fit the smaller screens

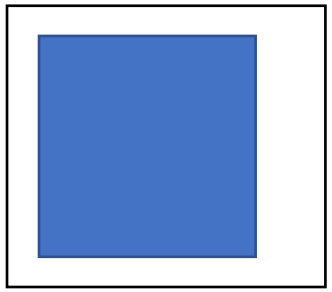
Breakpoint 1024px



Breakpoint 768px



Breakpoint 320px



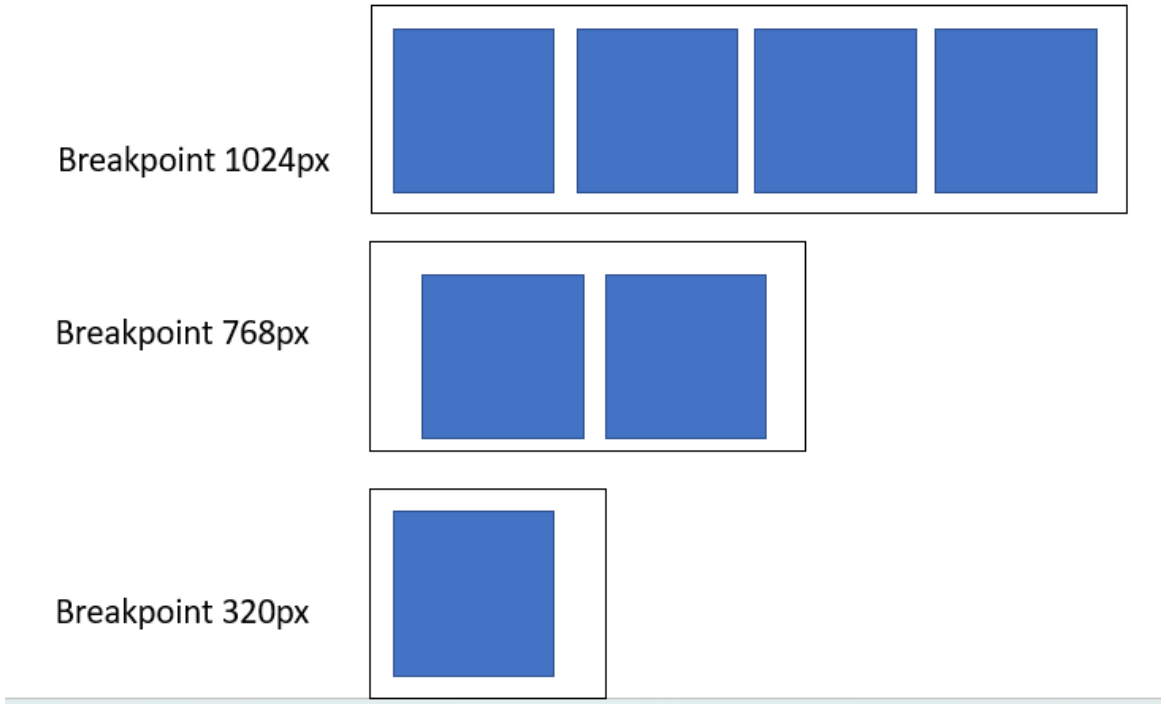
Adapt the
grid-template-columns !

ANSWER

Activity 2

✓ Reduce the number of columns to fit the smaller screens

```
.grid {
     display: grid;
     grid-template-columns: repeat(4, 1fr);
     gap: 16px;
     padding: 20px;
   }

   /* ≤ 768px → 2 columns */
   @media (max-width: 768px) {
     .grid {
       grid-template-columns: repeat(2, 1fr);
     }
   }

   /* ≤ 320px → 1 column */
   @media (max-width: 320px) {
     .grid {
       grid-template-columns: 1fr;
     }
   }
```

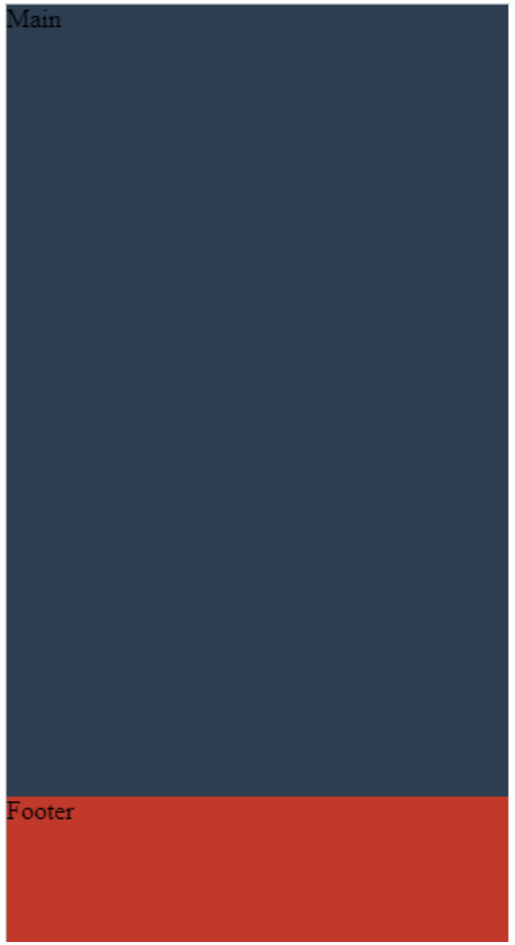


EXPLAIN

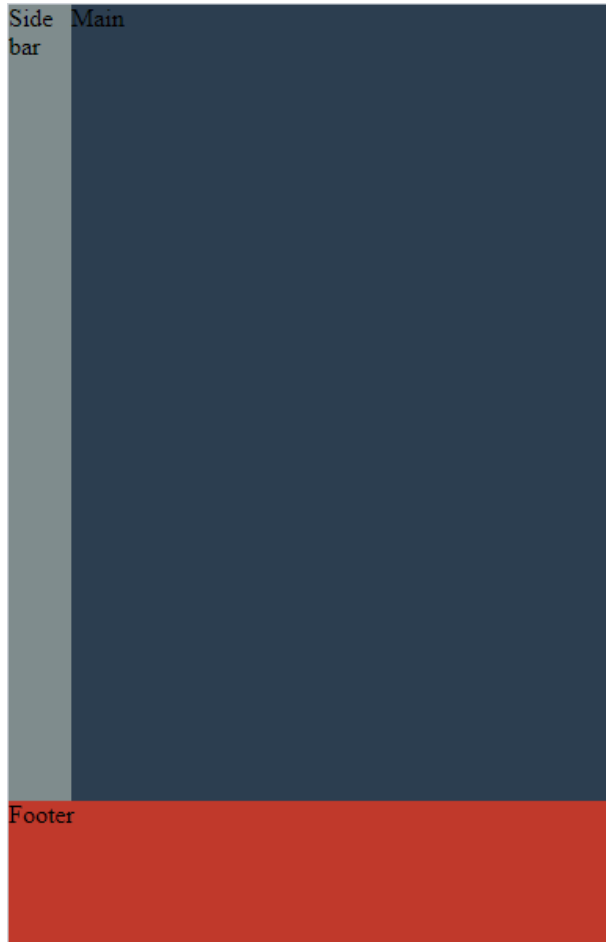
Media Queries with **Grid Areas**

```css
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

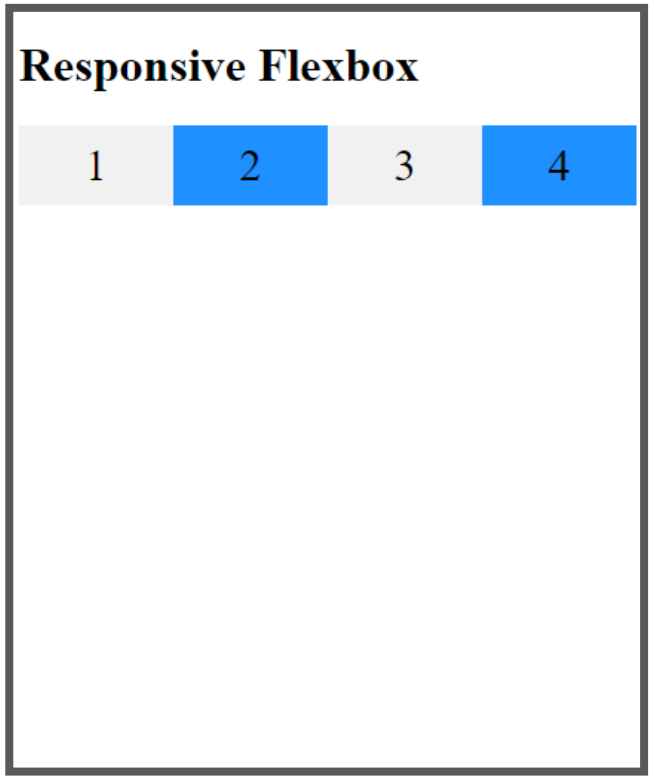


EXPLAIN

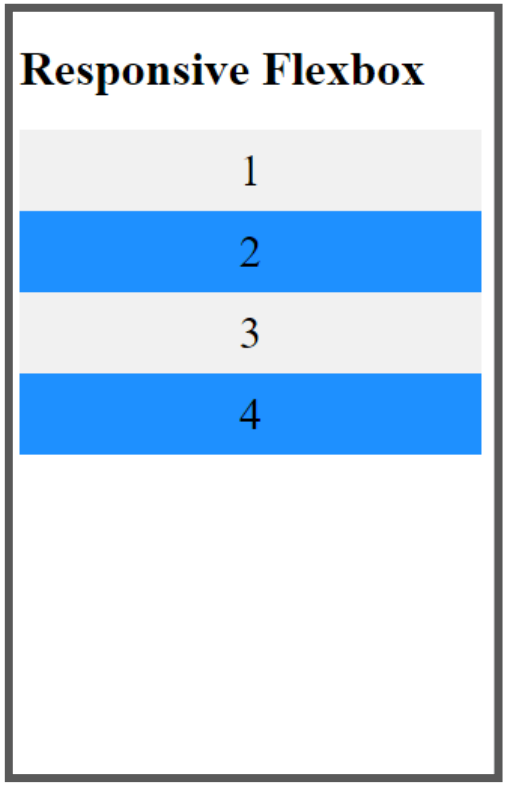
Media Queries with **Flex**

```css
.flex-container {
    display: flex;
    flex-direction: row;
}

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

- default



- max-width: 400px



EXPLAIN

Clamp()

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

```
calmp(minimumValue, preferedValue, 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 2rem

box:

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

EXPLAIN

7 UNITS RULES for responsive design

1. Use rem for text
Keeps font sizes consistent and scalable.

2. Use rem for spacing
Margins and padding scale nicely with the design.

3. Use % and fr for layout
Good for flexible widths in grids and flexbox.

4. Use vw / vh only when something must follow the screen size
Example: full-screen sections or big hero text.

5. Use px only when the value must never change
Example: borders (1px), hairlines.

6. Use clamp() to prevent text from becoming too small or too large
Example:
font-size: clamp(1rem, 2vw, 2rem);

7. Avoid em except inside components
It can grow unexpectedly.

QUIZ !

- ✓ The teacher starts clicking through the questions.
- ✓ The students answer the questions within the given time.
- ✓ At the end of the quiz, a leaderboard with the results for the entire group will be generated.

ARE YOU READY ?

Q1

What will be the color if the screen width is 200 px ?

```
<html>
    <body>
        <h1>Hello PNC !</h1>
        <style>
            @media (max-width: 300px) {
                body {
                    background-color: blue;
                }
            }
            @media (min-width: 301px) {
                body {
                    background-color: red;
                }
            }
        </style>
    </body>
</html>
```

The Answer is **A**

B

Q2

What will be the color if the screen width is 100 px ?

```
<html>
    <body>
        <h1>Hello PNC !</h1>
        <style>
            @media (max-width: 300px) and (min-width: 200px) {
                body {
                    background-color: red;
                }
            }
            @media (min-width: 301px) {
                body {
                    background-color: blue;
                }
            }
        </style>
    </body>
</html>
```

A

B

Hello PNC !

The Answer is C

Q3

What will be the color if the screen width is 210 px ?

```
<html>
    <body>
        <h1>Hello PNC !</h1>
        <style>
            @media (max-width: 200px) {
                body {
                    background-color: red;
                }
            }
            @media (max-width: 300px) and (min-width: 201px) {
                body {
                    background-color: blue;
                }
            }
            @media (min-width: 301px) {
                body {
                    background-color: green;
                }
        </style>
    </body>
</html>
```

Hello PNC !

A

Hello PNC !

The Answer is B

Hello PNC !

C

Hello PNC !

D

Q4

How the div will be dislplay if the screen width is 400 px ?

```
<html>
    <body>
        <div>
            <h1>Hello everyone!</h1>
        </div>
        <div>
            <h1>how are you?</h1>
        </div>
        <style>
            div {
                display: inline-block;
                border: 1px solid peru;
            }
            @media (max-width: 400px) {
                div {
                    width: 95%;
                }
            }
            @media (min-width: 401px) {
                div {
                    width: 45%;
                }
            }
        </style>
    </body>
</html>
```

Hello everyone!	how are you?

The Answer is **A**

Hello everyone!
how are you?

B

Q5

What style will be applied when viewing on a screen width of 500px?

```
.box {
  background: blue;
}

@media (max-width: 600px) {
  .box {
    background: red;
  }
}
```

A. Always blue

B. Always red

C. Red at 600px and below

D. Blue at 600px and below

Q6

What background color will `.card` have at 900px width?

```
.card {
  background: green;
}

@media (min-width: 1000px) {
  .card {
    background: orange;
  }
}
```

A. Green

B. Orange

C. Both colors

D. No background

Q7

What font size is actually rendered if current viewport with is 1000px?

```
.text {
  font-size: min(20px, 5vw);
}
```

Viewport width = 1000px → 5vw = ???

A. 50px

B. 20px

C. 25px

D. 5px

Q8

What is the computed width taken if the viewport width is 1200px?

```
.box {
  width: max(200px, 10vw);
}
```

Viewport width = 1200px → 10vw = ???

A. 120px

B. 200px

C. 320px

D. 100px

Q9

What is the output font size of the heading if the viewport with is 400px?

```
h1 {
  font-size: clamp(16px, 5vw, 32px);
}
```

Viewport width = 400px → 5vw = 20px

A. 16px

B. 20px

C. 32px

D. 40px

Q10

What color will the paragraph be at 1200px width?

```
p {
  color: black;
}

@media (max-width: 1000px) {
  p {
    color: red;
  }
}

@media (min-width: 1200px) {
  p {
    color: blue;
  }
}
```

A. Black

B. Red

C. Blue

D. Purple

Q11

What happens at width 768px?

```
div {
  padding: 20px;
}

@media (min-width: 768px) and (max-width: 1024px) {
  div {
    padding: 40px;
  }
}
```

A. padding = 20px

B. padding = 40px

C. padding = 0

D. padding = 60px

Q12

What is the computed width?

```
.container {
  width: clamp(300px, 50%, 600px);
}
```

Viewport width = 1000px → 50% = ???

A. 300px

B. 500px

C. 600px

D. 800px

Q13

At 400px screen width, what is the .menu display property?

```
.menu {
  display: block;
}

@media (max-width: 500px) {
  .menu {
    display: none;
  }
}
```

A. block

B. flex

C. inline

D. none

TODO AFTER

LEARN MORE !!

✓ **Read** more about Responsive Design
https://www.w3schools.com/html/html_responsive.asp

✓ **Practice** more on Responsive design
https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design