

---

## RESPONSIVE BASICS

---

# LEARNING OBJECTIVES

- Describe responsive design.
- Apply media queries to achieve a responsive layout.

---

## REVIEW QUESTIONS

---

- Which file is in charge of adding the Flexbox magic to our pages - HTML or CSS?
- What should our HTML structure look like to set up a Flexbox?
- How do we initiate Flexbox? Name the property. Which element would we set this property on (the flex container or flex items)?
- Which property can we use to determine the space between flex items? Which element would we set this property on (the flex container or flex items)?
- How can we determine how much width each column should take up?
- What are the advantages and disadvantages of a class-based grid system (like we created last class)
- Look up flex-wrap. What does this property do?

---

## PARENT WRAPPER (FLEX CONTAINER)

---

Setting **display: flex** on an element turns it into a flex container and gives us access to these properties:

- justify-content
- align-items
- flex-direction
- flex-wrap

Setting **display: flex** on an element turns any direct children into flex items. Here are some properties we can set on the individual children:

- flex-basis
- order

---

**RESPONSIVE BASICS**

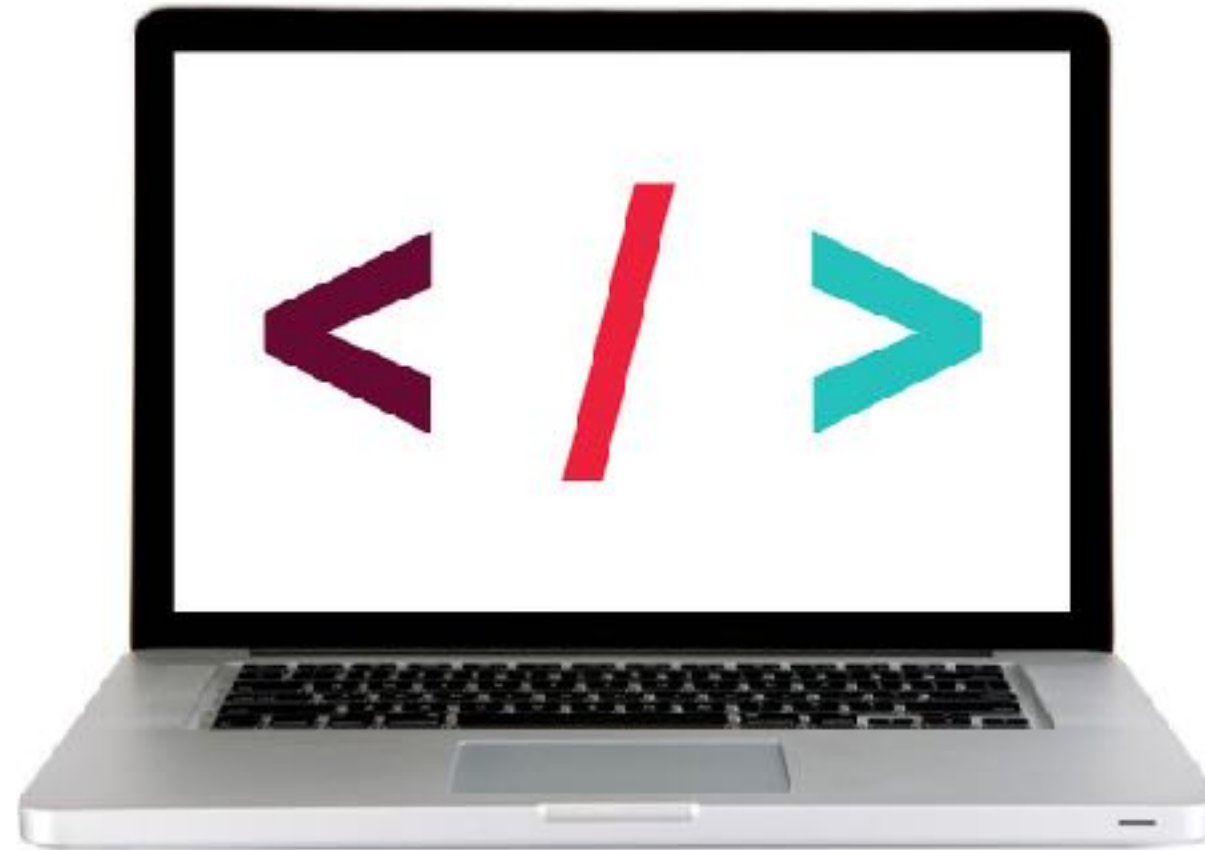
---

# RESPONSIVE — LAYOUT DESIGN

---

## LET'S TAKE A CLOSER LOOK

---



<https://generalassemb.ly/>

---

# RESPONSIVE DESIGN

---

“Day by day, the number of devices, platforms, and browsers that need to work with your site grows. Responsive web design represents a fundamental shift in how we’ll build websites for the decade to come.”

- Jeffrey Veen

---

# RESPONSIVE DESIGN

---



# RESPONSIVE DESIGN

---





---

# RESPONSIVE LAYOUT

---

- ▶ Uses percentages for widths
- ▶ Use media queries to control design and content as it scales down or up with the browser or device



---

# WITH BREAKPOINTS VS. WITHOUT BREAKPOINTS

---

With Breakpoints



Without Breakpoints

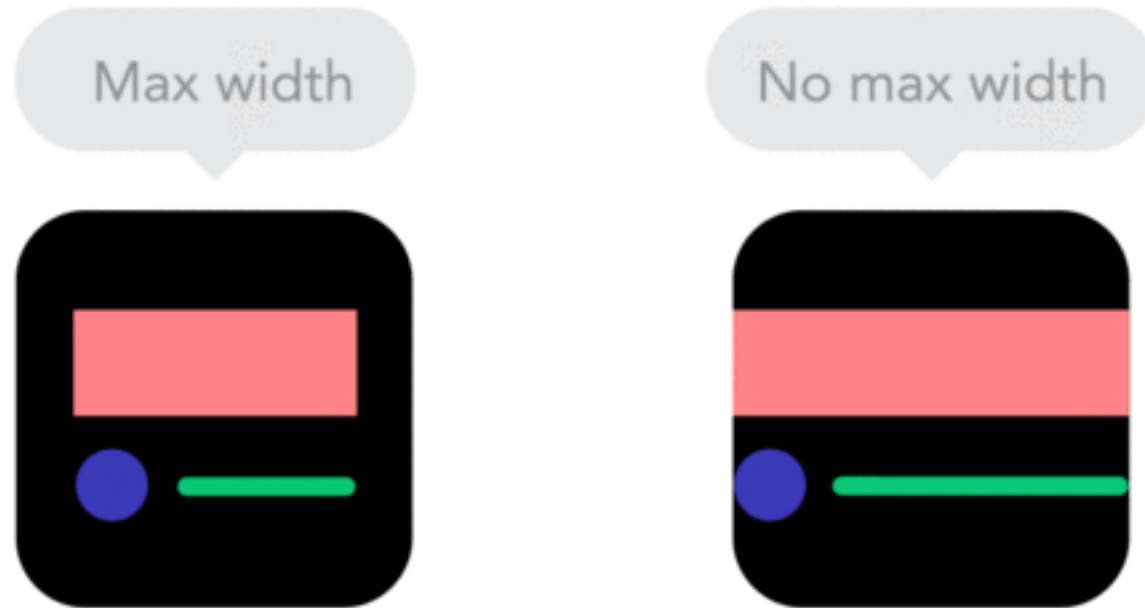


Gif credit: [Fast Company](#)

---

# MAX-WIDTH — A HELPFUL TOOL FOR LAYOUT

---



Gif credit: [Fast Company](#)

---

## RESPONSIVE BASICS

---

# RESPONSIVE — MEDIA QUERIES

---

## VIEWPORT META TAG — AN IMPORTANT NOTE!!

---

- The viewport meta tag controls how a webpage is displayed on a mobile device.
- Without the tag, mobile devices will assume you want the full desktop experience and will set the viewport width at 980px (iOS)

### DEVICE-WIDTH

- This tells the browser “My Website adapts to your width”

### INITIAL-SCALE

- Sets the initial zoom level and prevents default zooming

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

---

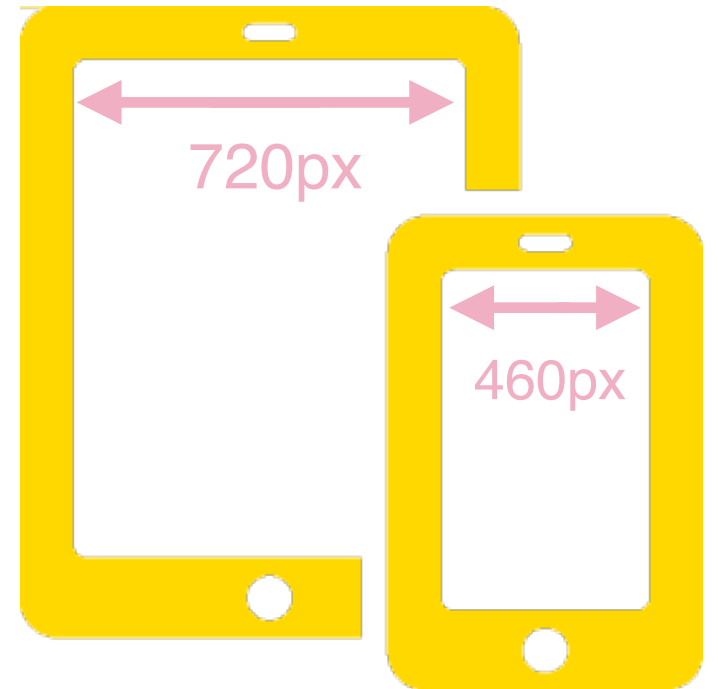
# MEDIA QUERIES

---

Media queries allow us to add or change styles for elements based on:

- screen size (width or height of browser window)
- device orientation (portrait or landscape)
- display density (retina or non-retina)

With media queries, we can allow most of our styles to remain the same, while we make small tweaks for specific formats.



---

# ADDING MEDIA QUERIES

---


```
@media screen and (max-width: 600px) {  
  section {  
    flex-basis: 50%;  
  }  
}
```

---

# MEDIA QUERIES — SYNTAX


---

## STYLES FOR COMPUTER SCREENS



```
@media screen and (max-width: 600px) {  
  section {  
    flex-basis: 50%;  
  }  
}
```

## STYLES FOR PRINT



```
@media print {  
  /* All your print styles go here */  
  header, footer, nav {  
    display: none;  
  }  
}
```



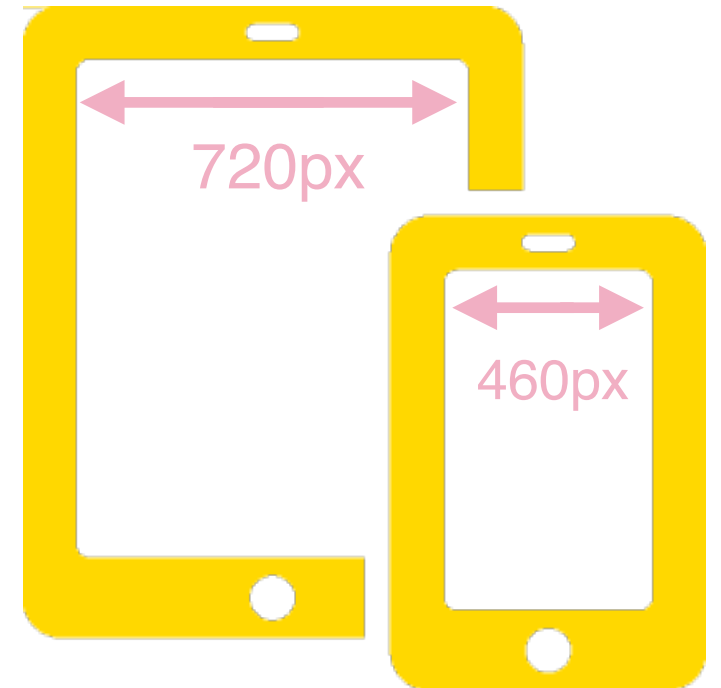
# MEDIA QUERIES — SYNTAX

## MAX-WIDTH: SCREENS 600PX OR LESS

```
@media screen and (max-width: 600px) {  
  section {  
    flex-basis: 50%;  
  }  
}
```

## MIN-WIDTH: SCREENS 600PX OR MORE

```
@media screen and (min-width: 600px) {  
  section {  
    flex-basis: 50%;  
  }  
}
```



---

# MEDIA QUERIES — ORDER

---

1. Place your media queries at the bottom of your stylesheet so they can override code that comes before them.
2. If you have multiple media queries, they should go from largest to smallest (max-width: 600 would be placed above max-width: 400)

```
@media screen and (max-width: 600px) {  
  section {  
    flex-basis: 50%;  
  }  
}
```

```
@media screen and (max-width: 400px) {  
  section {  
    flex-basis: 100%;  
  }  
}
```

---

# TURN AND CHAT

---

- What are media queries? What can we change inside media queries?
- Where should we add media queries? Why is this order important?
- I have the two following media queries. How should these be rearranged? Can anything be removed?

```
@media screen and (max-width: 400px) {  
  .hamburger {  
    display: block;  
  }  
  section {  
    flex-basis: 100%;  
  }  
}
```

```
@media screen and (max-width: 600px) {  
  .hamburger {  
    display: block;  
  }  
  section {  
    flex-basis: 50%;  
  }  
}
```

---

**FEWD**

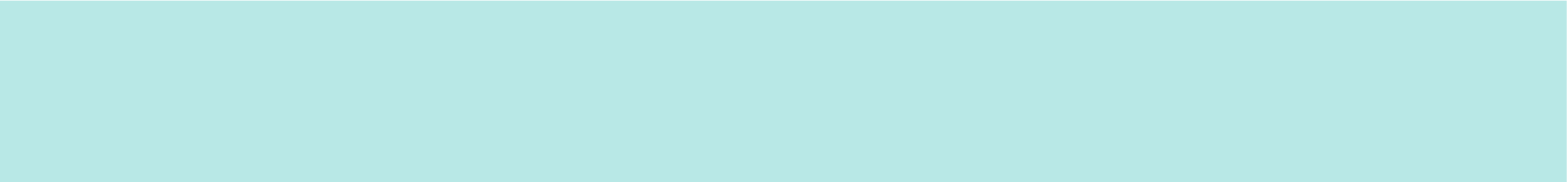
---

# GRID SYSTEMS — RESPONSIVE

---

**ROWS**

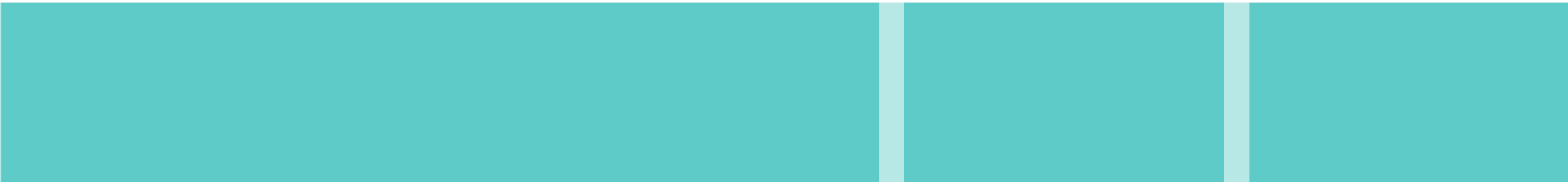
---



---

# COLUMNS

---



---

# ROWS

---

Since all our rows will be set to **display: flex;** we can store this property in a class and apply it to any element to turn it into a row.

```
.row {  
  display: flex;  
  justify-content: space-between;  
  flex-wrap: wrap;  
  margin-bottom: 20px;  
}
```

---

# COLUMNS

---

We can create classes for the different options for column widths. We can then apply these classes to any flex item to set its width.

```
.col-1-4 {  
  flex-basis: 25%;  
}
```

```
.col-1-3 {  
  flex-basis: 33%;  
}
```

```
.col-1-2 {  
  flex-basis: 50%;  
}
```



---

## COLUMNS — RESPONSIVE

---

We can then create classes for different screen sizes in a media query:

```
@media screen and (max-width: 970px) {  
  .col-md-1-4 {  
    flex-basis: 24%;  
  }  
  .col-md-1-2 {  
    flex-basis: 49%;  
  }  
  .col-md-full {  
    flex-basis: 100%;  
  }  
}
```

---

## COLUMNS — RESPONSIVE

---

We can then create classes for different screen sizes in a media query:

```
@media screen and (max-width: 750px) {  
  .col-sm-full {  
    flex-basis: 100%;  
  }  
}
```

---

## COLUMNS — RESPONSIVE

---

And finally, we can add any classes that apply to our HTML. We can add multiple classes to one element with a space-separated list:

```
<section class="col-1-4 col-md-1-2 col-sm-full">Content</section>
```

---

**RESPONSIVE BASICS**

---

**MORE RESOURCES**

---

## MORE RESOURCES — THIS IS RESPONSIVE

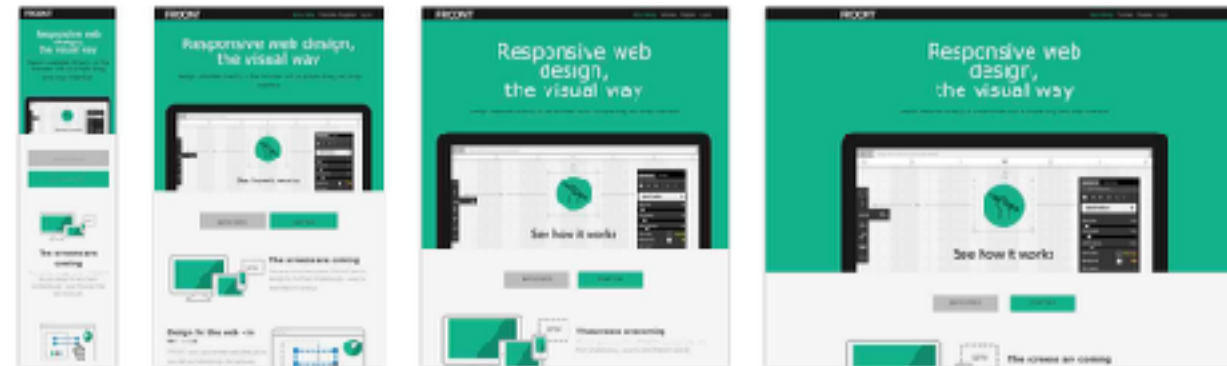
---



# MORE RESOURCES — MEDIA QUERIES



FROONT



International Polar Foundation



---

## MORE RESOURCES — MEDIA QUERIES

---



---

## RESPONSIVE BASICS

---

# LEARNING OBJECTIVES

- Describe responsive design.
- Apply media queries to achieve a responsive layout.