

Web Fundamentals



Session-07

Responsiveness

Agenda



- 01 What is a responsive Design ?
- 02 Responsive design Elements
- 03 Mobile Friendly Website
- 04 CSS Frameworks(Bootstrap etc)
- 05 Bootstrap deep dive
- 06 Hands-On, Summary & Resources

Why a website need to be responsive?



Why a website need to be responsive?

People use tablets and smartphones for different purposes

90%

of people
use multiple
screens
sequentially



81%

Browse the
Internet



67%

Shop
Online



46%

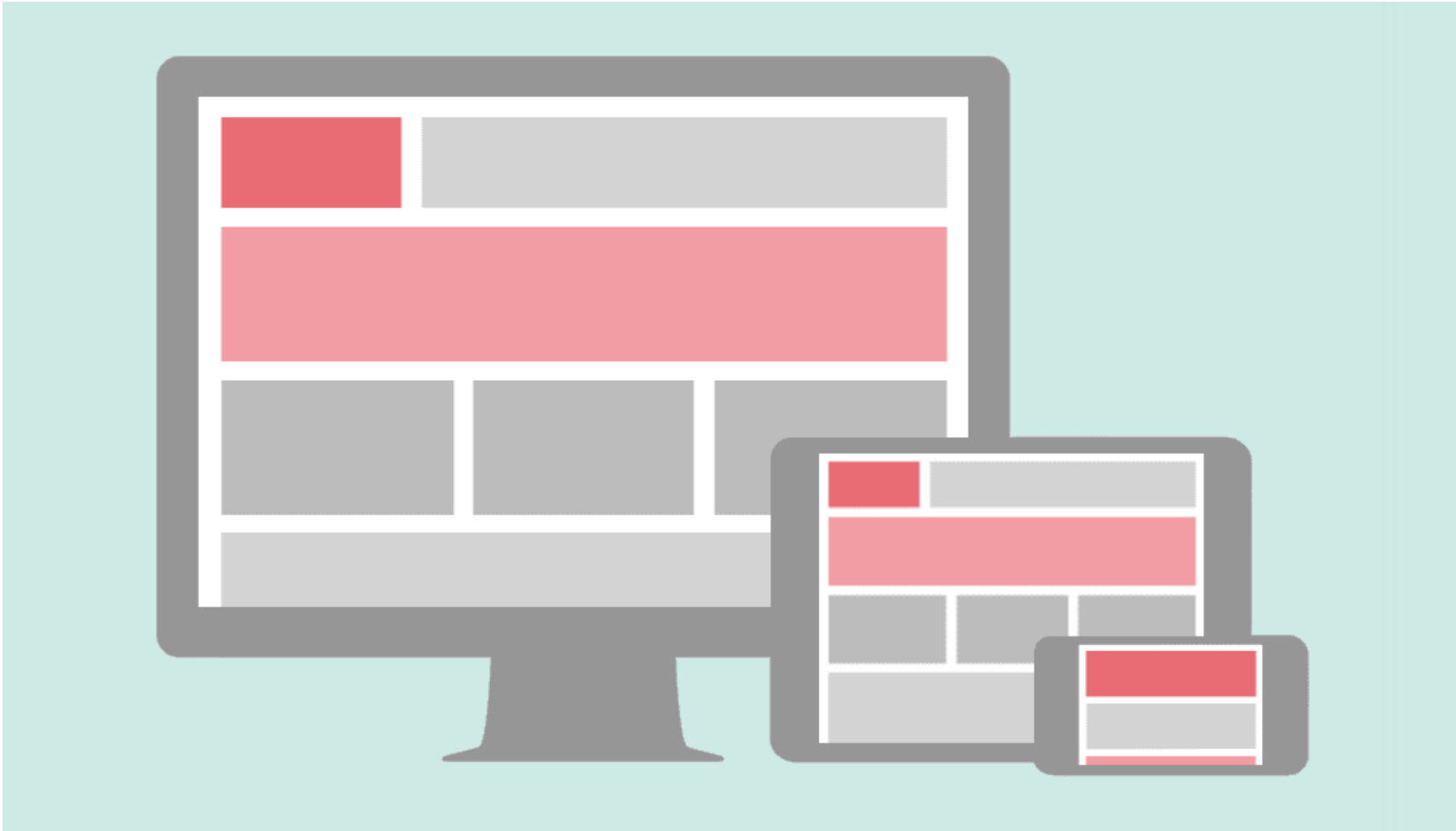
Manage
Finances



43%

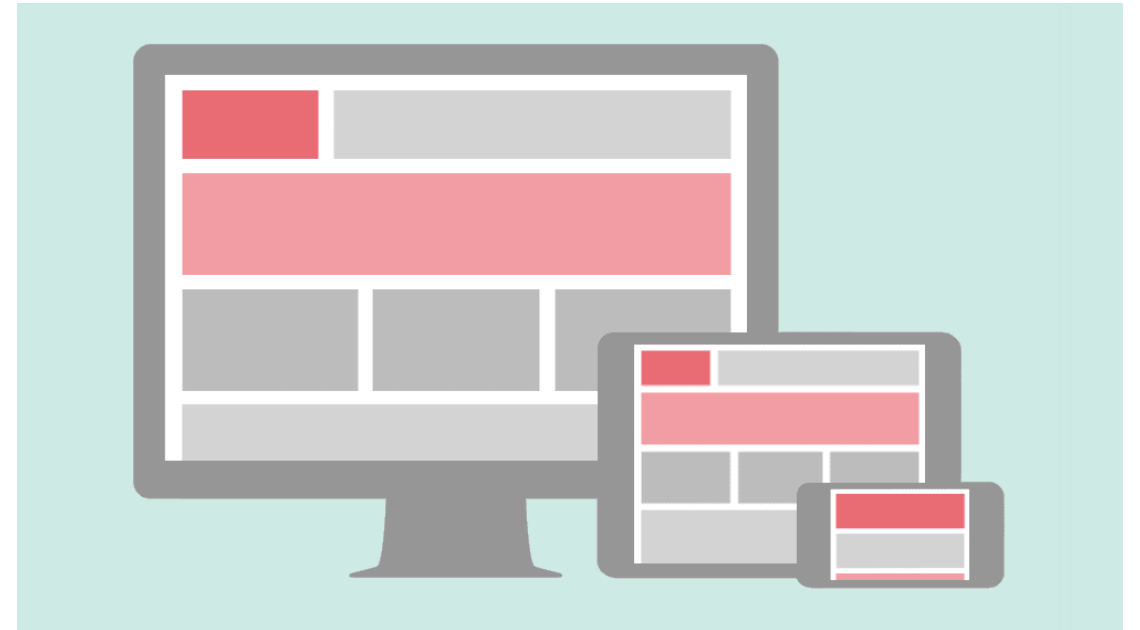
Plan
Vacations

HOW DO MARKETERS ENGAGE USERS ON SO MANY SCREENS?

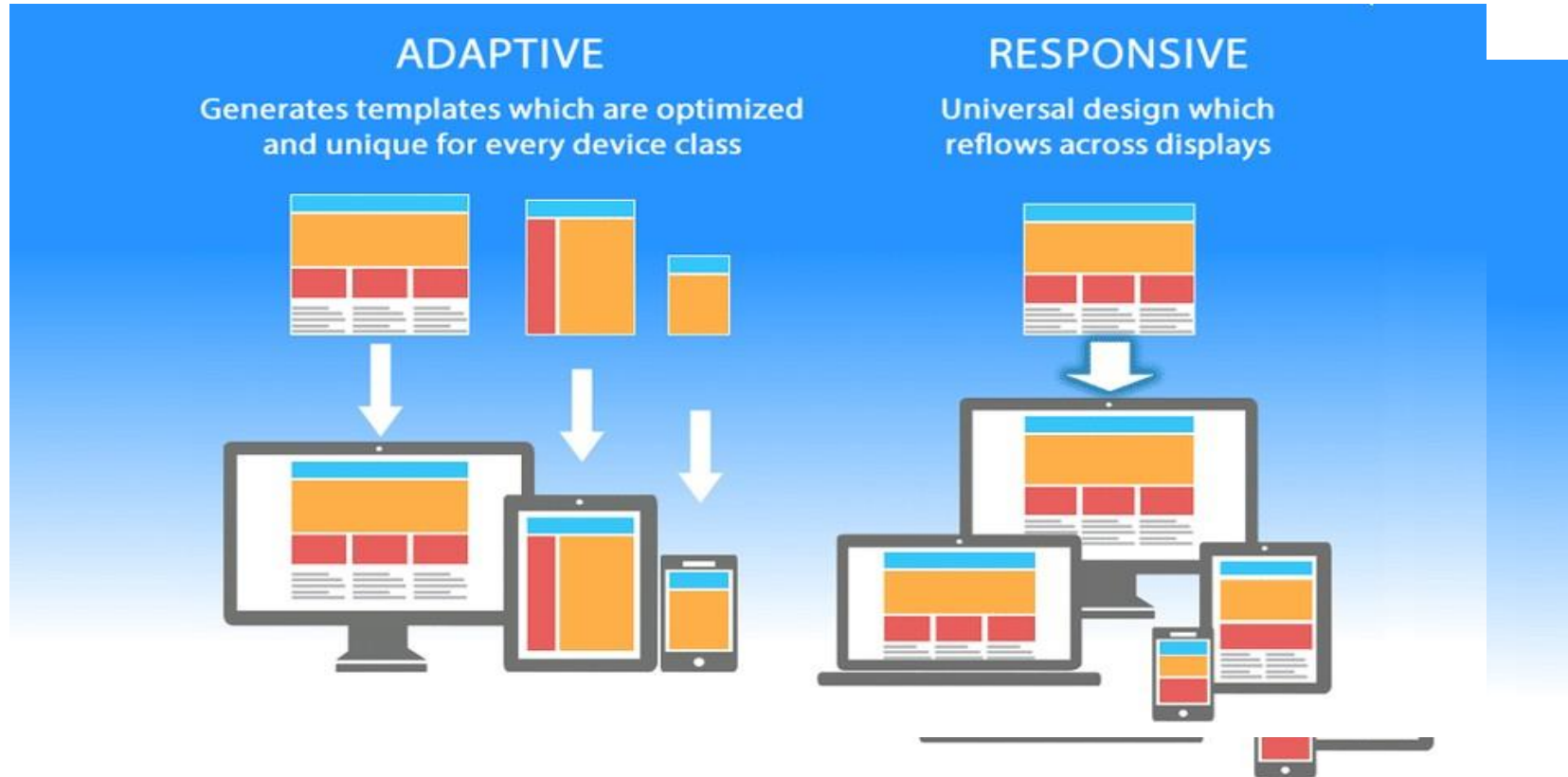


RESPONSIVE DESIGN

- Separate layouts of your content and design to different devices depending on screen size.
- CSS code will control the layout and render it differently based on screen size



Responsive Vs Adaptive



What are the disadvantages of adaptive?

Why Responsiveness?



Responsiveness SEO Advantages

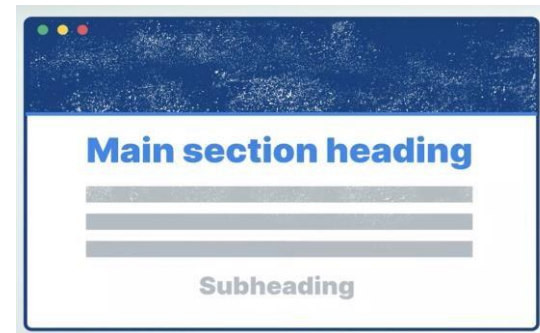


**“Having A Mobile Responsive Website Is Not
Just Another Option**

it’s A Requirement!”

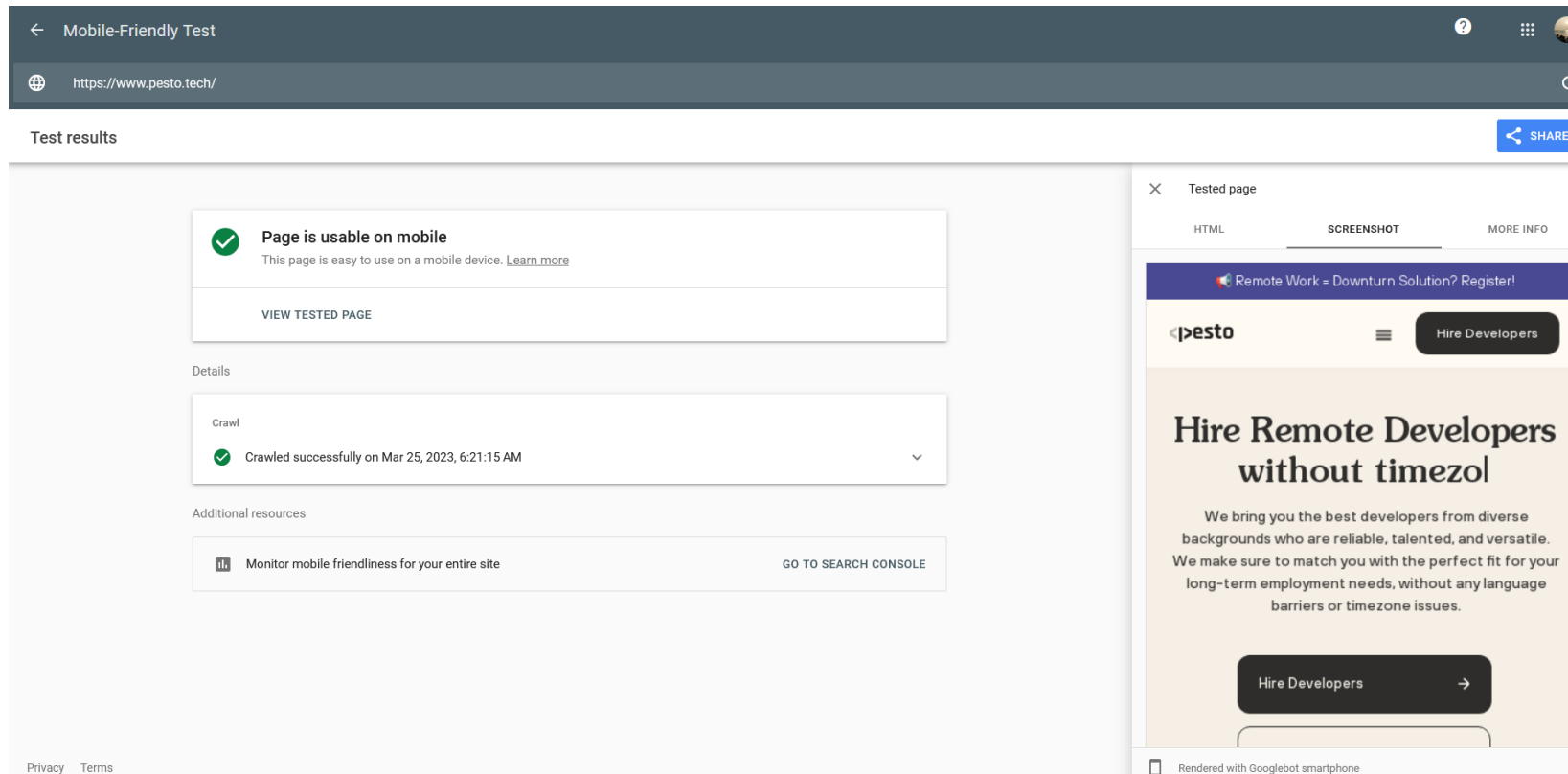
What makes a website mobile friendly?

- Watch out for popups
- Forget the mouse
- Ask fewer questions in forms
- Shorter paragraphs
- Speed things up



Mobile Friendly test Demo

1. <https://search.google.com/test/mobile-friendly>
2. Developer Tools



The screenshot displays the Google Mobile-Friendly Test interface. At the top, the browser address bar shows the URL <https://www.pesto.tech/>. Below the address bar, the page title is "Mobile-Friendly Test". The main content area is titled "Test results" and features a green checkmark icon next to the heading "Page is usable on mobile". A subtext states, "This page is easy to use on a mobile device. [Learn more](#)". A button labeled "VIEW TESTED PAGE" is positioned below this section. Under the "Details" heading, a "Crawl" section shows a green checkmark and the text "Crawled successfully on Mar 25, 2023, 6:21:15 AM". In the "Additional resources" section, there is a link to "Monitor mobile friendliness for your entire site" with a "GO TO SEARCH CONSOLE" button. On the right side, a "Tested page" panel shows a mobile preview of the pesto.tech website. The preview includes the pesto logo, a "Hire Developers" button, and the main heading "Hire Remote Developers without timezol". The text below the heading reads: "We bring you the best developers from diverse backgrounds who are reliable, talented, and versatile. We make sure to match you with the perfect fit for your long-term employment needs, without any language barriers or timezone issues." A "Hire Developers" button with a right arrow is visible at the bottom of the preview. The bottom of the page shows "Privacy" and "Terms" links.

The How Part



Responsiveness Design Elements

- HTML & CSS
- Breakpoints
- Fluid layout
- Flex box / Grid layout

Choose Breakpoints you need to target

Desktop



@media screen and
(min-width: 1024px)
{...}

Tablet



@media screen and
(min-width: 768px) and
(max-width: 1023px)
{...}

Smartphone



@media screen and
(max-width: 767px)
{...}

Media Queries for any Typical breakpoint

```
/* Extra small devices (phones, 600px and down) */  
@media only screen and (max-width: 600px) {...}
```

```
/* Small devices (portrait tablets and large phones, 600px and up) */  
@media only screen and (min-width: 600px) {...}
```

```
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {...}
```

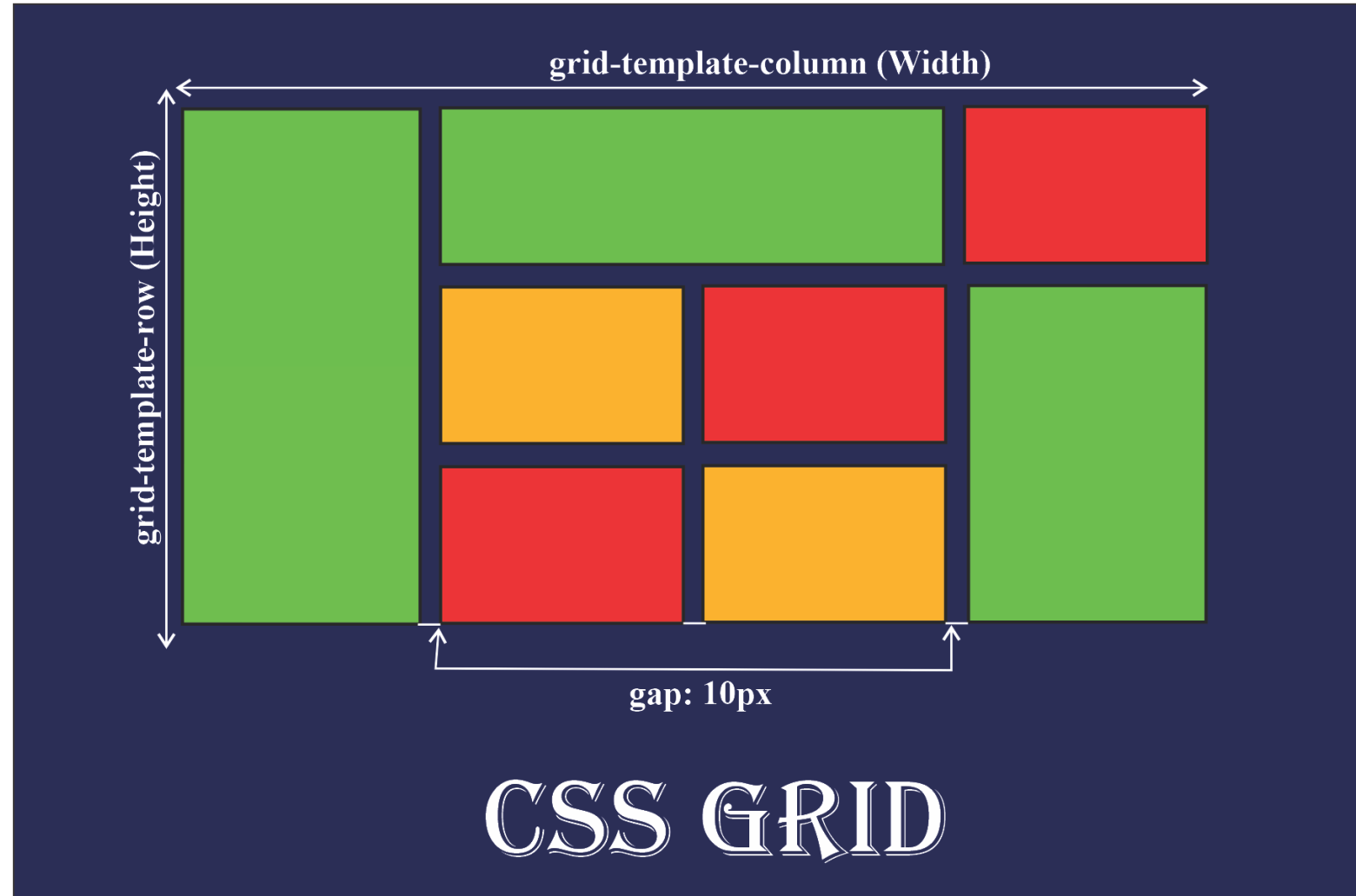
```
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {...}
```

```
/* Extra large devices (large laptops and desktops, 1200px and up) */  
@media only screen and (min-width: 1200px) {...}
```


Different Layout Techniques

- Grid Layout
- Fluid Layout
- Flex Layout

Grid Layout



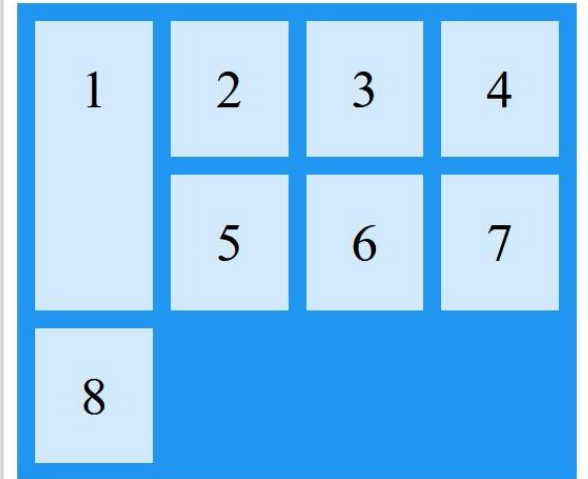
Container based, 2D technique layout

Grid Layout: Example

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-row-start: 1;
  grid-row-end: 3;
}
```



You can refer to line numbers when placing grid items.

Fluid Layout

Fluid layout

width: 63%

width: 32%

A fluid layout relies on dynamic values like a **percentage of the viewport width**.

Fluid Layout: Example

```
.sidebar {  
  width: 20%;  
  /* Sidebar width is set to 20% of the container width */  
  float: left;  
  /* Sidebar is floated to the left */  
  background-color: #f2f2f2;  
  /* Example background color */  
}  
  
.content {  
  width: 80%;  
  /* Content width is set to 80% of the container width */  
  float: left;  
  /* Content is floated to the left */  
  padding: 20px;  
  /* Example padding for content */  
}
```

Flex Layout

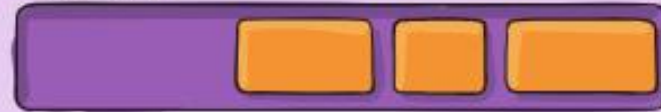
```
.parent {  
  display: flex;  
  flex-flow: row wrap; /* OK elements, go as far as you can on o  
}
```

justify-content

flex-start



flex-end



center



space-between



space-around



space-evenly



Flex Vs Grid

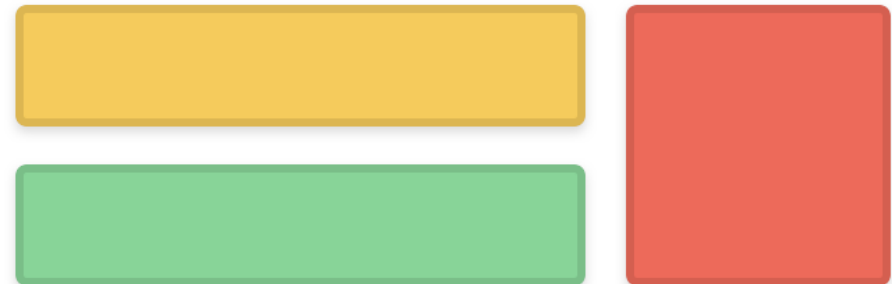
Flexbox

One-dimensional layout



Grid

Multi-dimensional layout



Responsiveness Concepts

- Meta tag
- Media Query
- Relative units
- Flexible Typography
- Responsive images
- Bitmap vs vector images for icons.

Meta Tag

To **signal to browsers that your page** adapts to all devices, add a meta tag to the head of the document:

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

A media query is a fundamental part of CSS3 that lets you render content to adapt to different **factors like screen size** or resolution.

Media Queries

```
article {  
  width: 50%;  
}
```

PREFERRED

```
@media only screen and (max-width: 600px) {  
  article {  
    width: 200px;  
  }  
}
```

SMALL SCREENS

```
@media only screen and (min-width: 1200px) {  
  article {  
    width: 800px;  
  }  
}
```

LARGE SCREENS

A media query is a fundamental part of CSS3 that lets you render content to adapt to different **factors like screen size** or resolution.

Media Queries

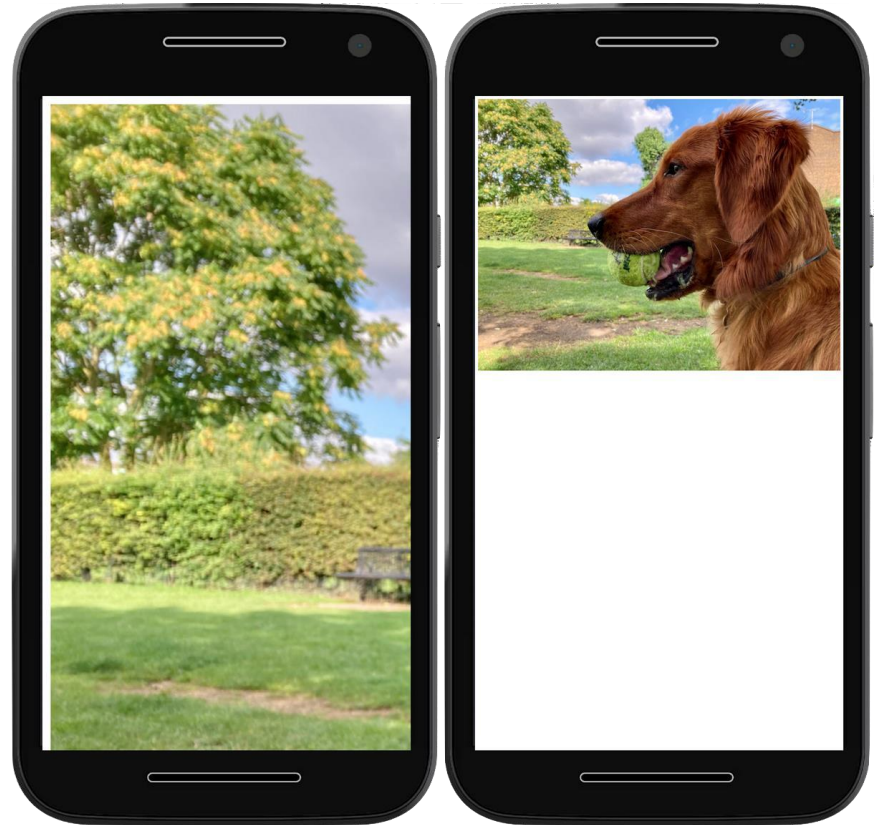
```
@media all and (orientation: landscape) {  
  /* Target device in landscape mode */  
}  
  
@media all and (orientation: portrait) {  
  /* Target device in portrait mode */  
}
```

Relative Units

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

```
img {  
  max-inline-size: 100%;  
  block-size: auto;  
}
```

Responsive Images



The % unit approximates to a single percentage of the width or height of the viewport and makes sure the image remains in proportion to the screen.

I don't want to do all this, Can I get readymade??



CSS Frameworks

CSS Frameworks

CSS frameworks **offer generic functionalities** that can be overridden.

CSS frameworks **save the time**

CSS Frameworks



Top CSS Frameworks



Drilldown on Bootstrap

To start using Bootstrap:

- Go to GetBootstrap.com.
- Download the most current version (Bootstrap current version: 5.1).
- Unpack into a directory and upload to your server.

That's it. You now "have Bootstrap."

Add reference in head section

```
<html>
```

```
<h
```

```
<link rel="stylesheet" a href="css/bootstrap.css"/>|  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<!--
```

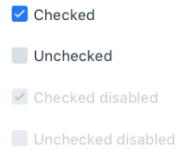
```
-->
```

Use existing classes- Demo

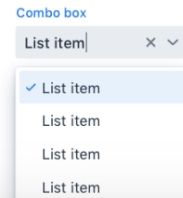
HTML

```
<div class="col-lg-6">
  <div class="card alert">
    <h4>Default</h4>
    <div class="multi-select">
      <select id="pre-selected-options" multiple="multiple">
        <option value="elem_1" selected="">elem 1</option>
        <option value="elem_2">elem 2</option>
        <option value="elem_3">elem 3</option>
        <option value="elem_4">elem 4</option>
        <option value="elem_5">elem 5</option>
        <option value="elem_6">elem 6</option>
        <option value="elem_7">elem 7</option>
        <option value="elem_8">elem 8</option>
        <option value="elem_9">elem 9</option>
        <option value="elem_10">elem 10</option>
        <option value="elem_11">elem 11</option>
        <option value="elem_12">elem 12</option>
      </select>
    </div>
  </div>
</div>
```

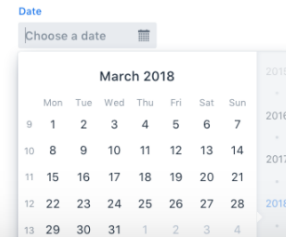
Ready-made custom components



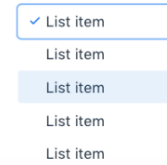
Checkbox
Latest v2.2.6



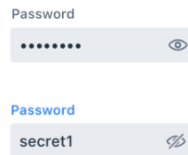
Combo Box
Latest v4.2.2



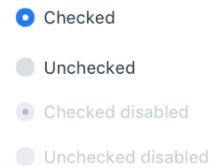
Date Picker
Latest v3.3.2



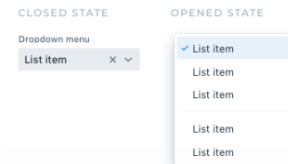
List Box
Latest v1.1.0



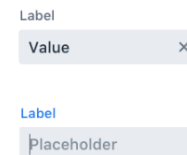
Password Field
Latest v-



Radio Button
Latest v



Select
Latest v



Text Field
Latest v1.1.0

Support for modern frameworks

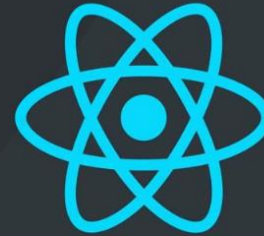


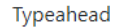
Angular

+



Bootstrap





B B B B B B B B

Disadvantages

- It is extraordinarily heavy
- Expectation to learn and adapt
- Customization is a pain

“If you want to learn to swim,
jump into the water.”

–Bruce Lee



Q & A