IRON
HACK

Responsive Web Design

Summary

In this episode...

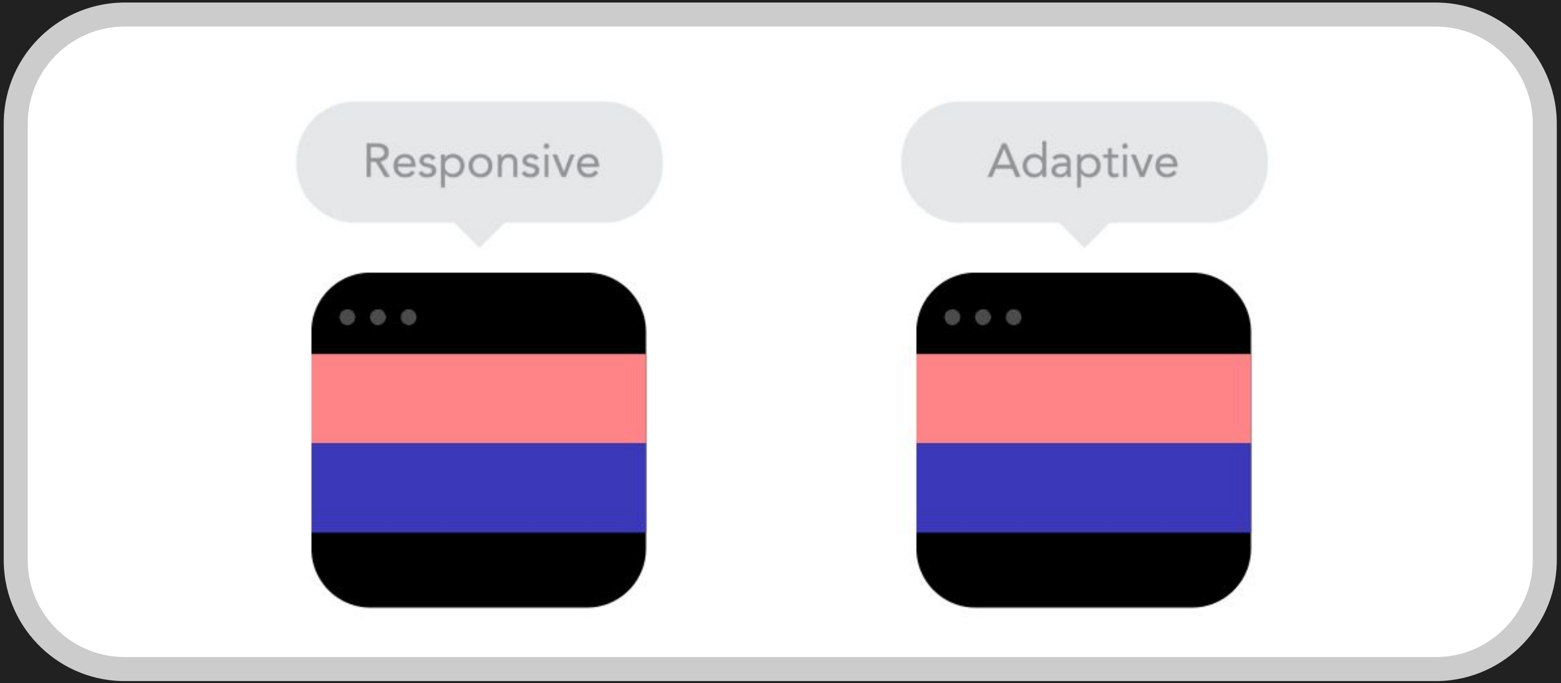
- What is Responsive Design
- The Grid
- Media Queries
- Fluid grids

What is Responsive Design?

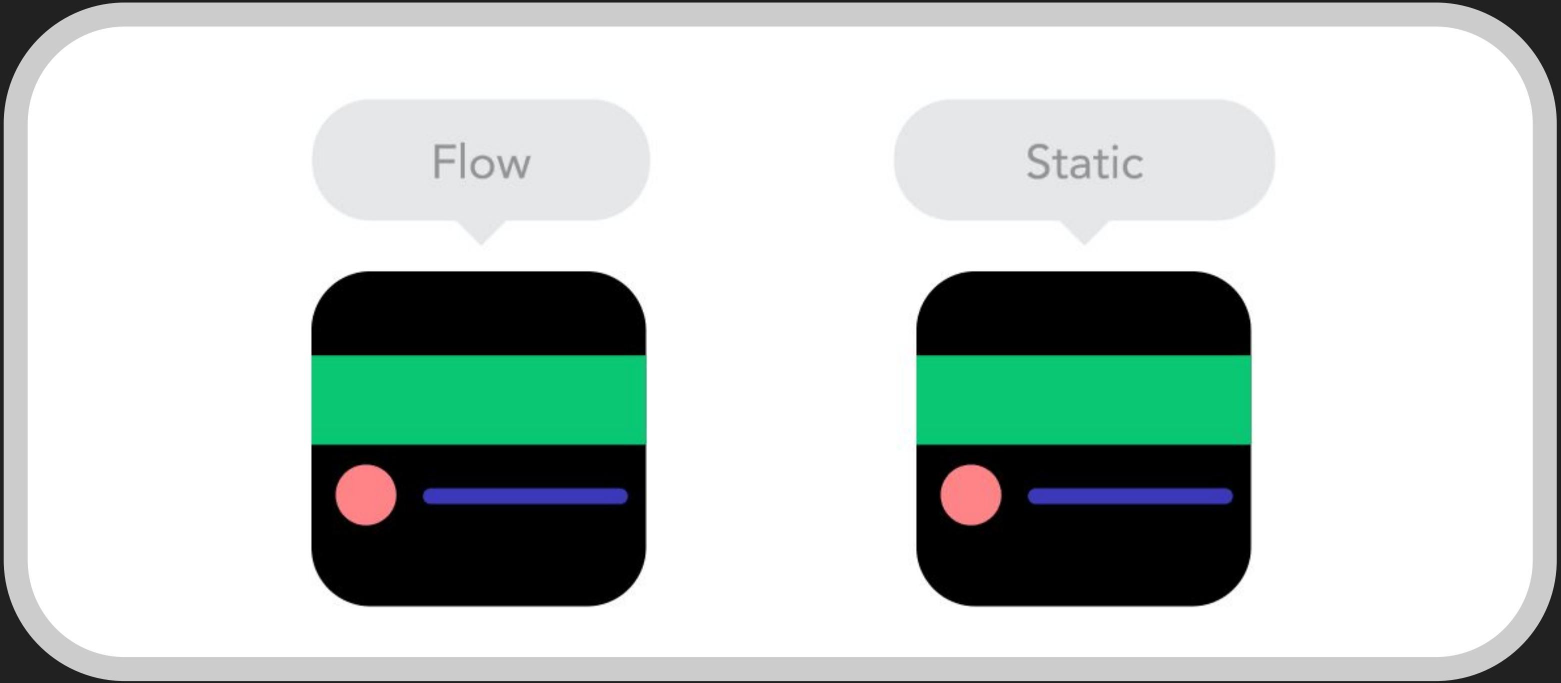
Responsive Web Design is the practice of building a website suitable to work on every device and every screen size, no matter how large or small, mobile or desktop.

Responsive web design is focused around providing an intuitive and gratifying experience for everyone.

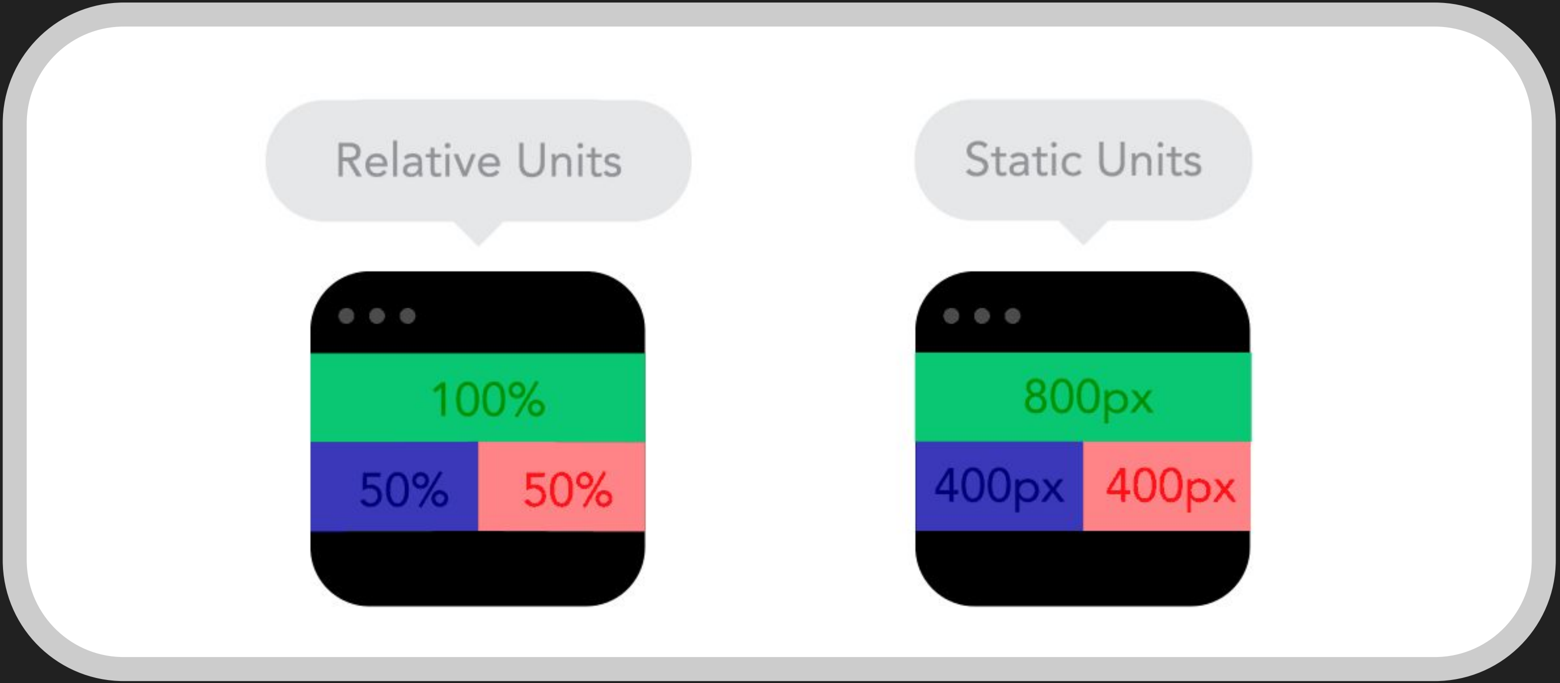
Principles of Responsive Design



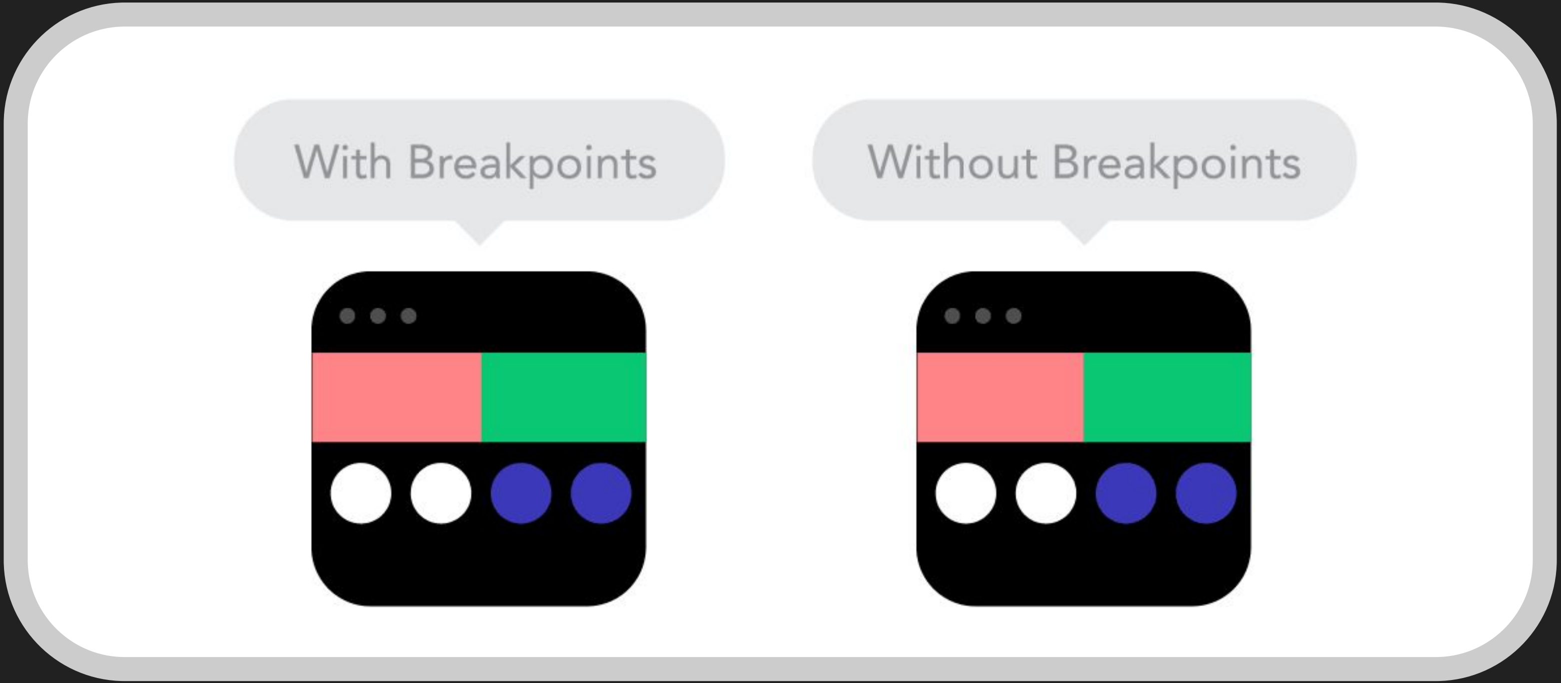
Principles of Responsive Design



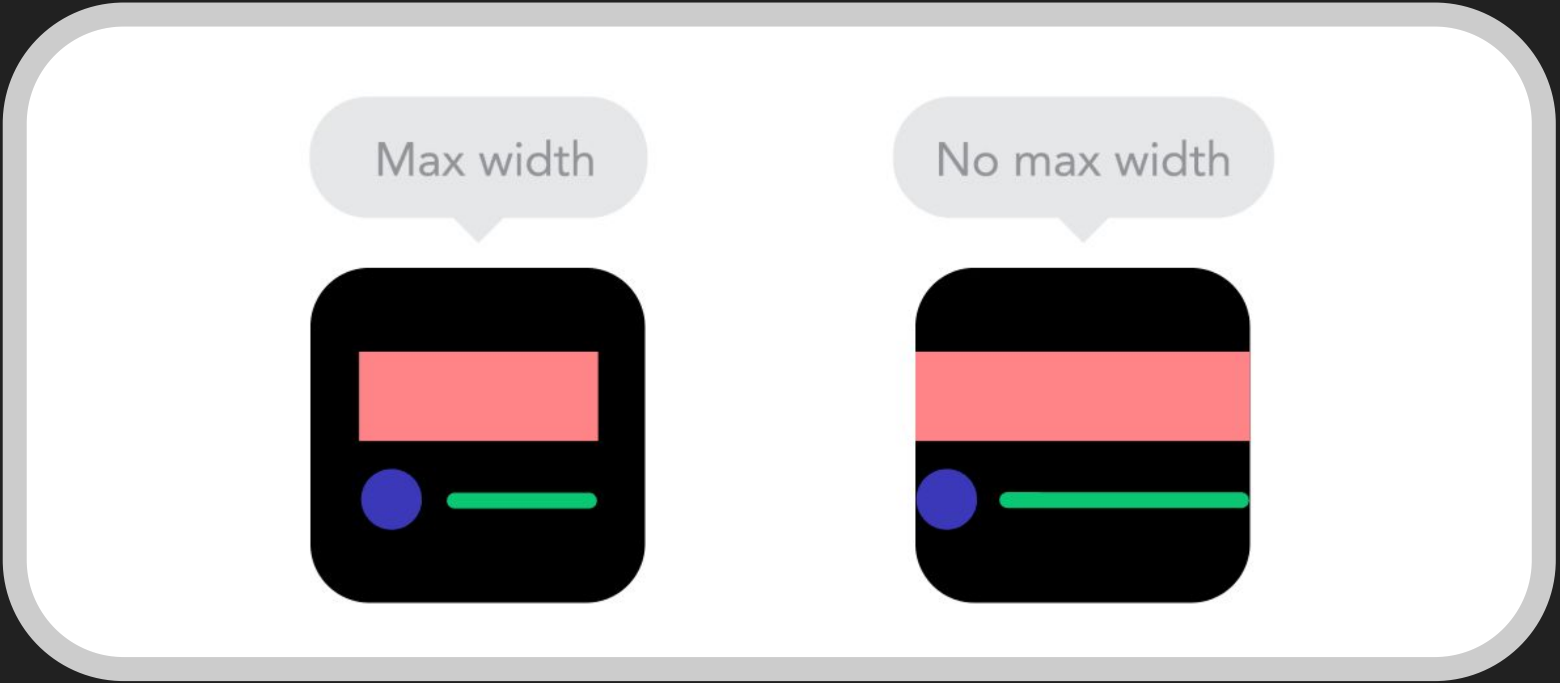
Principles of Responsive Design



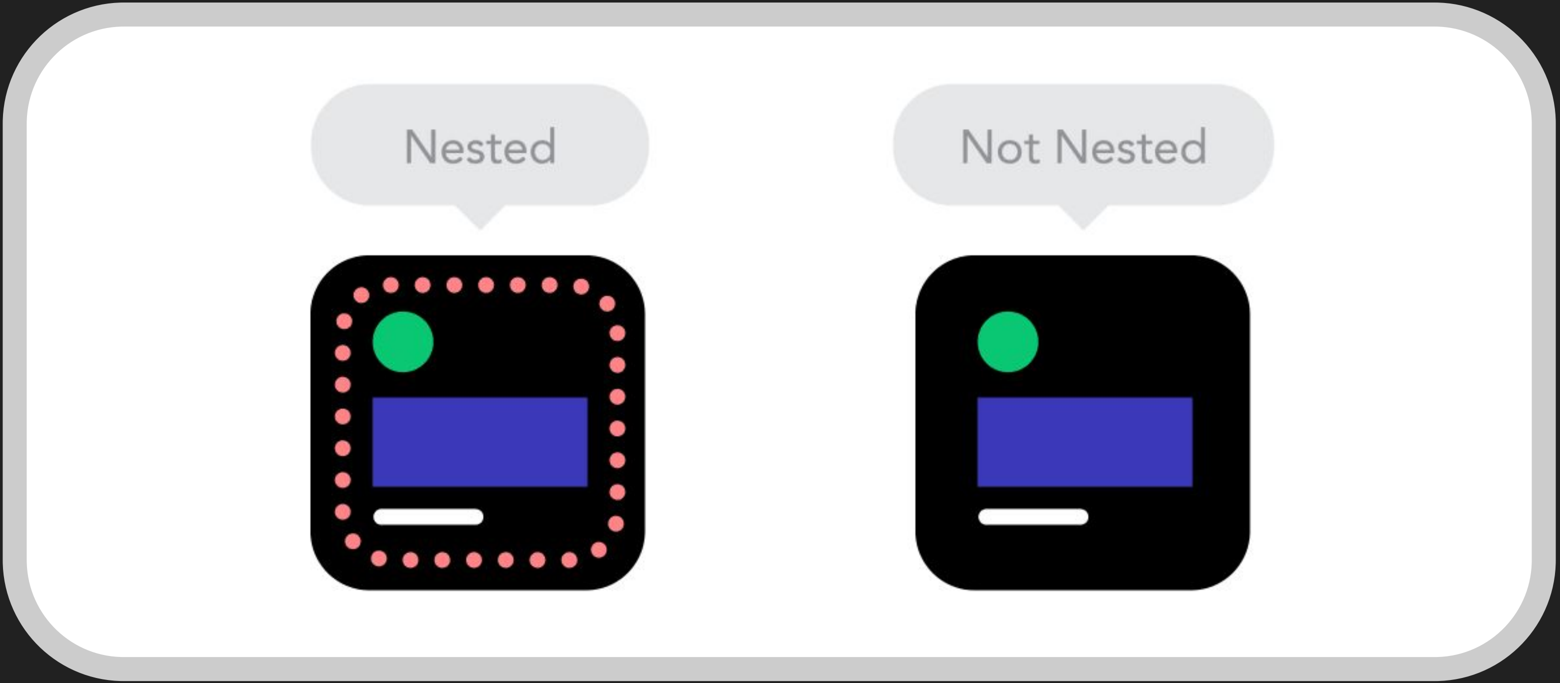
Principles of Responsive Design



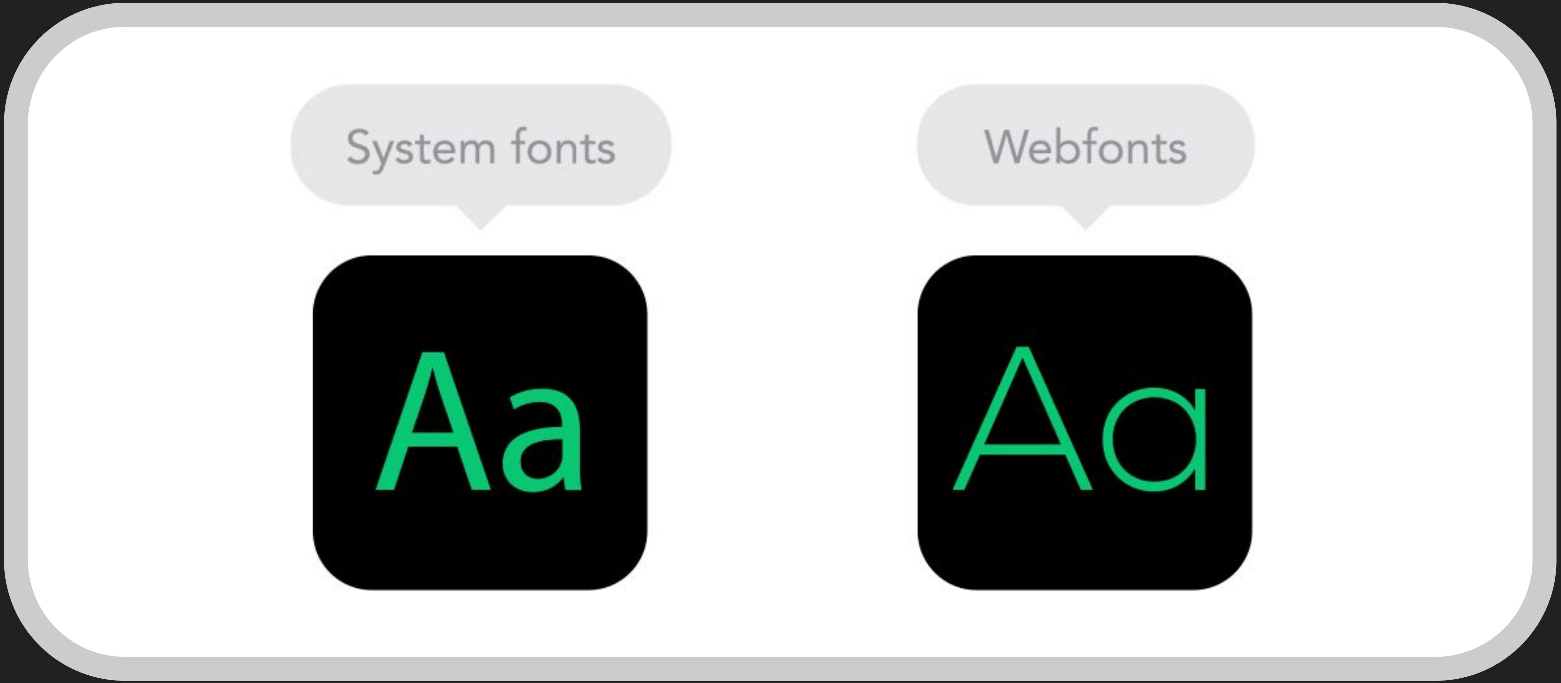
Principles of Responsive Design



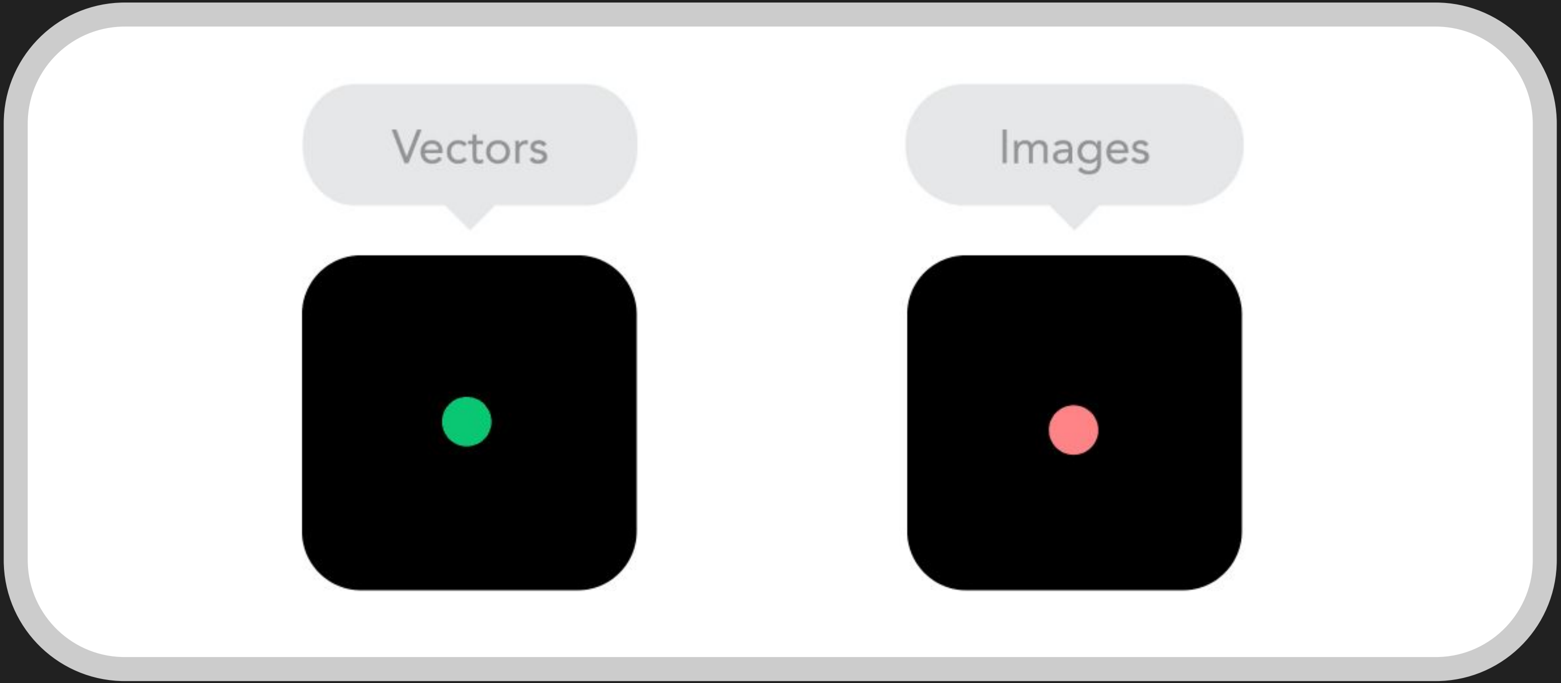
Principles of Responsive Design



Principles of Responsive Design

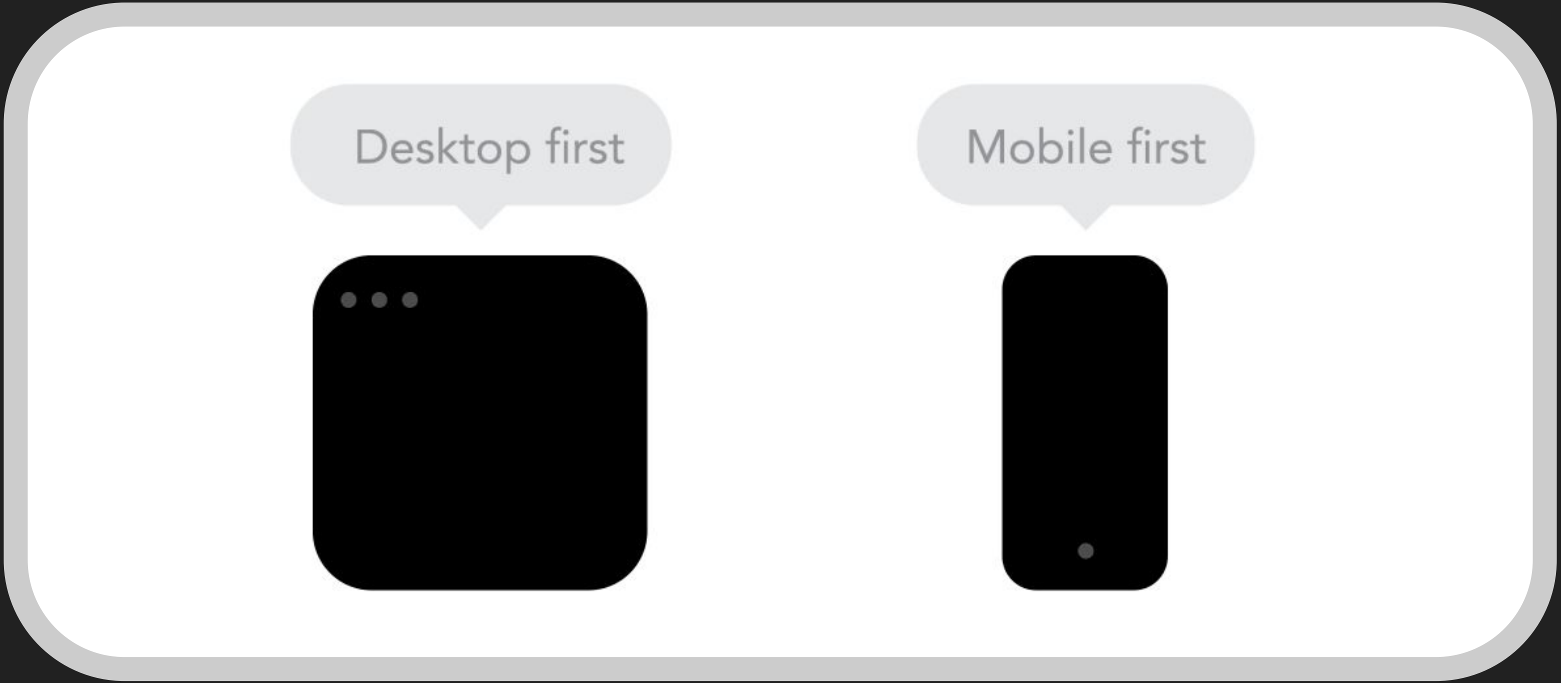


Principles of Responsive Design



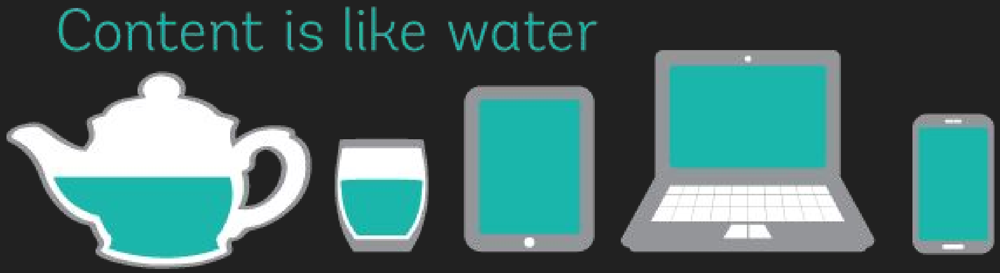
IRON HACK

Principles of Responsive Design



What is Responsive Design?

Responsive web design is broken down into three main components, including flexible layouts, media queries, and flexible media.



"We can't design a new experience on every platform from scratch every time. Content is like water; it takes many forms and flows into all these different containers."

Josh Clark, designer & developer

Viewport meta tag

Viewport meta tag allows you to control the viewport's size and scale. Many other mobile browsers now support this tag, although it is not part of any web standard.

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

The `width` property controls the size of the viewport.
The `initial-scale` property controls the zoom level when the page is first loaded.

The Grid

Flexible layouts is the practice of building the layout of a website with a flexible grid, capable of dynamically resizing to any width.

Why did the web developer leave the restaurant?

Because of the table layout

Let's code!

What we'll do here:

- Start with a very basic grid.
- Create a little CSS3 Media Query Reporter.
- Make our grid fluid instead of fixed.
- Add a media query to help the grid adjust to small screen widths.

IRON HACK

Let's code!

- To organize your code properly, it would be a good idea to create these files:
    - responsive.html
    - responsive-style.css
    - responsive-mediaquery-reporter.css
  -
- In your html file body, you should have something like this:

```
<body>
     <div class="container clearfix">
          <h1>Web Development</h1>
          <h2>iOS Development</h2>
          <!-- Some columns and elements -->
</body>
```

Media Queries

CSS Media queries are a feature of CSS3. They allow you to target CSS rules based on - for instance - screen size, device-orientation or display-density. The basic syntax looks like this:

```
#header-image {
     background-repeat: no-repeat;
     background-image:url('image.gif');
}
/*show a larger image when you're on a big screen*/
@media screen and (min-width: 1200px) {
     #header-image {
          background-image:url('larger-image.gif');
     }
}
```

Media Queries - *Let's code!*

- This is where the fun begins! In our css file let's create a rule for screens bigger than 1024px.

We use `only` *to hide phone style sheets from older browsers*

```
@media only screen and ( min-width: 1024px ) {
    body:after {
        content: "1024 and up"
    }
}
```

IRON HACK

Media Queries - *Exercise*

- Now create by yourself rules for screens bigger than 320px, 480px and 768px. Set different content and backgrounds to see changes.

Let's code!

We can use the same basic concepts to build a responsive grid system. Copy these lines in your style.css file:

```css
.container {
    width: 940px;
    margin: 0 auto;
}
.row {
     margin-left: -20px;
}
[class*="span"] {
     display: inline;
     float: left;
     margin-left: 20px;
}
.span-one-third {
     width: 300px;
}
```

Select elements with a class that contains 'span'

Keep in mind! This code is a very basic version of Bootstrap's grid. We are going to easily convert a fixed-width grid to a percentage-based fluid grid.

IRON HACK

The grid

Let's code

Add some columns and elements to your html to test the new style you just created

A fluid grid
Let’s code

Change your style.css:

```
.container {
        margin: 0 40px;
    }
    .row {
        margin-left: -3%;
    }
[class*="span"] {
     display: inline;
     float: left;
     margin-left: 3%;
    }
    .span-one-third {
        width: 30%;
    }
```

Keep in mind! This code is a very basic version of Bootstrap’s grid. We are going to easily convert a fixed-width grid to a percentage-based fluid grid.

IRON HACK

Media queries - Single transition
Let’s code

At widths smaller than 768px wide, our design will have a single column, so that the content can easily flow vertically down a narrow screen. But once we have a screen size that is at least 768px wide, the three-column layout will kick in, allowing us to make efficient use of the available screen space.

Can you change the code to fix this?

Solution: You just have to wrap `.span-one-third` *with a rule for screens under 768px*