Responsive Design in 2022

Who am I?

- I'm Emily!
- A Director for the Cincinnati chapter of Women Who Code and I'm also the UX Designer for LCvista.
- Designing and developing professionally for 11 years.
- I love photography, PC gaming, and hiking.

Let's break it down

We will go over the past, present and future of responsive design and how we can implement it.

- The Past and Benefits of Responsive Design
- What is Responsive Design?
- Fundamentals and Future Features
- Implementation

IN THE PAST

the blue abyss

bio
diary
poetry
prose
photos
friends
links

survey
quizzes
guestlog

i had a dream about getting trapped inside a black hole
but instead of black, everything was a beautiful, intense blue.

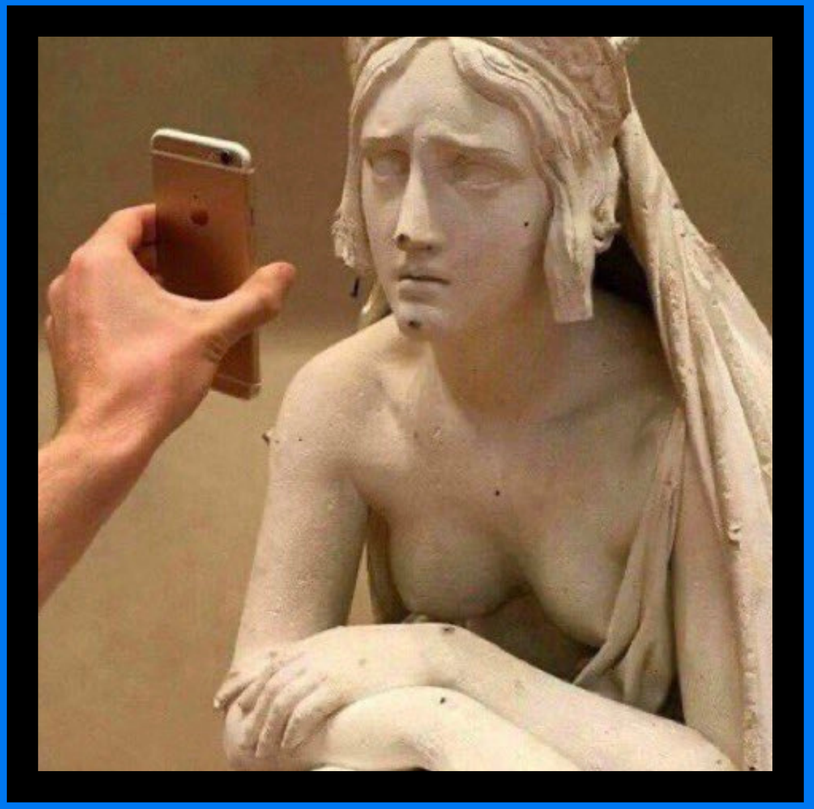
what was your last dream?

(answer)

!

(email)

(designed on a mac for ie 5+ and 800x600.)



Desktop

This is an Example of a Non-Responsive Design

This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space.

This is a subheading level 2

I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of

Accessibility Topics

- Types of disabilities
  - Blindness
  - Deafblindness
  - Colorblindness
  - Low vision
  - Deafness
  - Dexterity/Motor
  - Cognitive
  - Seizure
- Assistive technologies
  - Screen readers
  - Screen magnifiers
  - Alternative pointing devices
- Laws and regulations
  - Americans with Disabilities Act (ADA)
  - Section 508
  - Air Carrier Access Act (ACAA)
  - (CVAA)
- Guidelines
  - Web Content Accessibility Guidelines (WCAG)
- Certification
  - IAAP Certified Associate in Accessibility
  - IAAP Certified Professional in Web Accessibility

Tablet

This is an Example of a Non-Responsive Design

This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space.

This is a subheading level 2

I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of text just to fill the space. I will repeat it over and over again because that will fill up the space with more words, which is what I want to do. This is the content for this page. This is a lot of

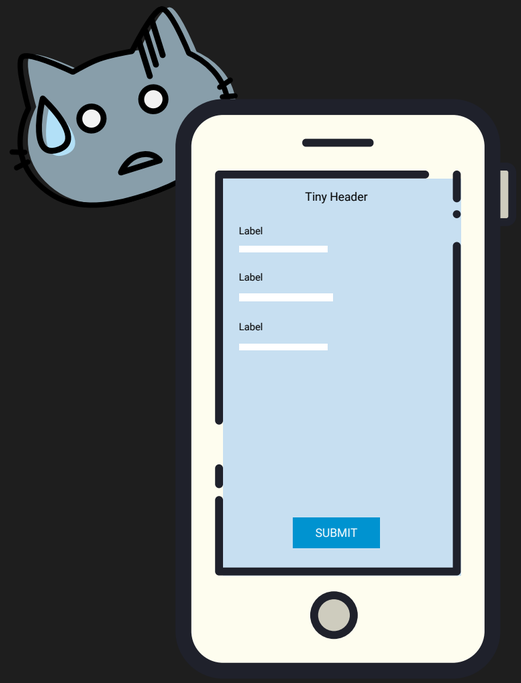
Why Learn Responsive Design?

Senior Front End Developer

Requirements / Qualification

- 7+ years of Front-End UI Architecture Design & Development Experience
- 5+ years of Consulting experience, designing and developing Consumer facing enterprise level responsive web applications
- 3+ years of experience working as a technical/team/module lead for team size of 3-4
- Demonstrated ability building websites using HTML5/CSS/JavaScript/Bootstrap/SASS/Compass etc.
- Extensive expertise using script frameworks: ReactJS / Redux lifecycles, AngularJS, Node.js
- Ability to build responsive, cross-browser, highly-performant, and accessibility compliant applications (i.e. WCAG)
- Proficient in using built-in browser tools like Chrome Developer Tools
- Experience in NPM, node, grunt and other similar build tools.
- Experience with browser-based debugging and performance testing software
- Experience evaluating component libraries and choosing the best toolkit for the job
- Experience of automated testing of UI in a micro-service environment.
- Strong working experience with version control (Git/Bitbucket) and branching strategies is a plus
- Strong experience with publishing and consuming REST APIs
- Knowledge of SEO and user analytics tools and techniques
- Have a deep understanding of modern web development techniques and how to apply new technologies to improve the architecture and boost page speed
- Experience with Agile Software Development

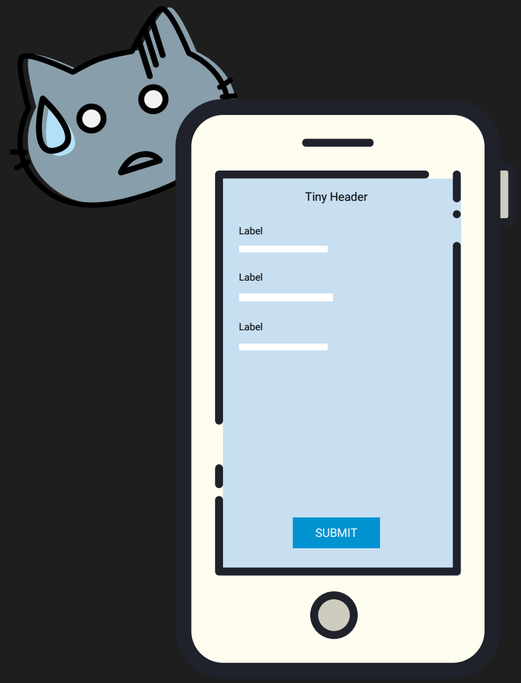
Accessibility.



Applications that aren't rendered correctly on mobile devices will have massive accessibility problems. Some of the most common issues are:

- Text that is too small
- Buttons, links, and form fields that can't be pressed
- Incorrect color contrast on text

Accessibility.



Responsive design is an additional tool that you can use to solve accessibility issues.

It's important that everyone--including people with disabilities--have access to the same resources.

So, what is Responsive Design?

Responsive applications and websites respond to their environment. Responsive design is an approach to web design that makes your web content adapt to the different sizes of a variety of devices.

What is Adaptive Design?

Certain companies still use something called **adaptive design**. It takes some concepts of responsive design, but does not account for all current and future resolutions. With this, design teams provide absolute pixel requirements for layouts and elements that do not respond accordingly.

What is Adaptive Design?

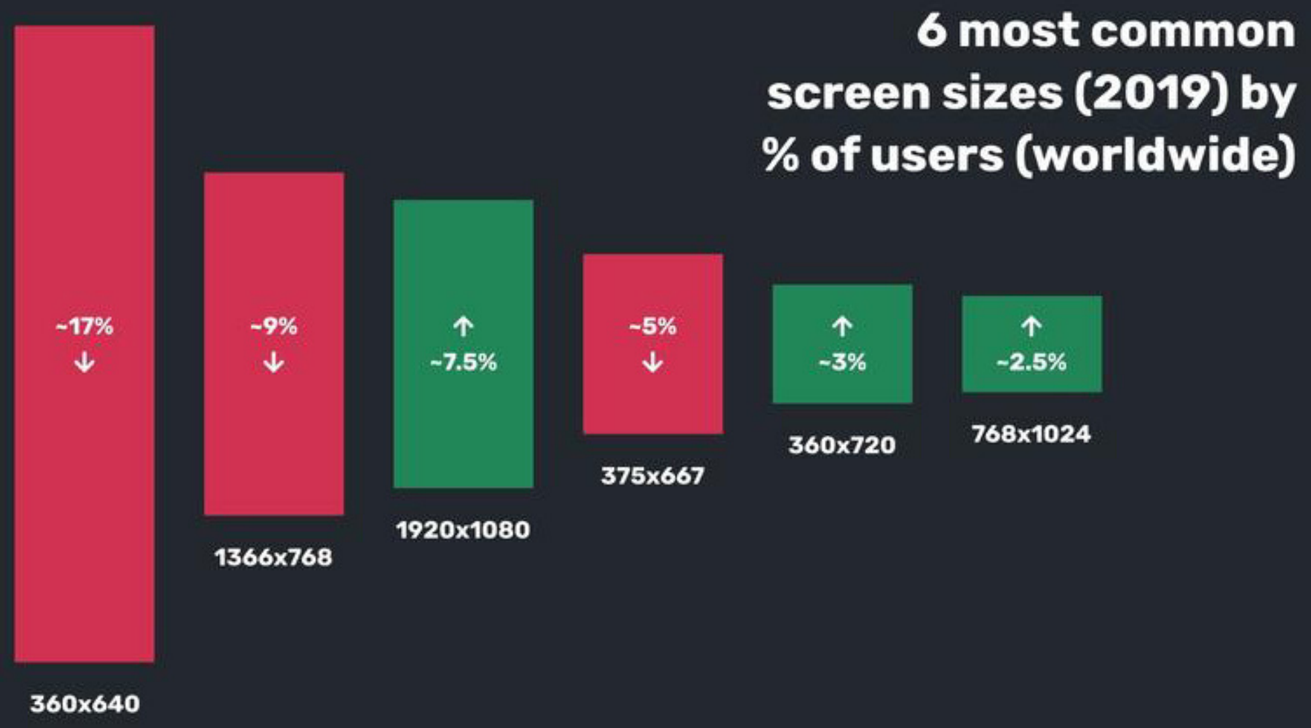
The difference between responsive design and adaptive design is that responsive design adapts the rendering of a single page version. In contrast, adaptive design delivers multiple versions of the same page.

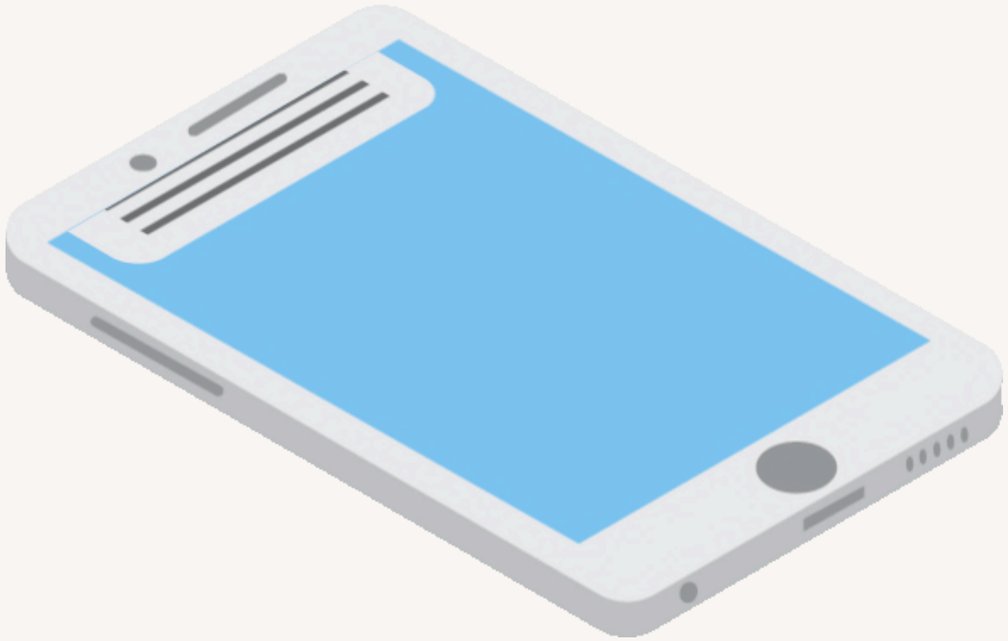
But what about common resolutions?

Having static designs was common 8-10 years ago, and it was a carry over from print. There is no "standard website or app width" in 2022 due to the wide variety of devices out today.

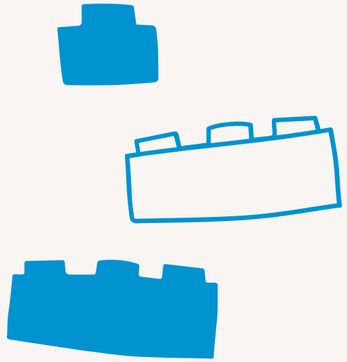
But what about common resolutions?

We have to consider that mobile users might use landscape orientation, and desktop users might not have their window maximized, so really, it could be anything.





It's about designing and developing to fit all resolutions. Screen sizes are always changing, so it's important that your web app can adapt to **any** screen size, today or in the future.



Fundamentals of Responsive Design

Fundamentals of Responsive Design

Fundamentals of Responsive Design

Relative Units

A relative unit gets sizing from either a parent, root element, or the size of the viewport.

Fundamentals of Responsive Design

Relative Units

A relative unit gets sizing from either a parent, root element, or the size of the viewport.

Fluid Grids

The grid is an element with areas in columns and rows. A fluid grid can be set to auto-fit or be set manually to adjust across breakpoints.

Fundamentals of Responsive Design

Relative Units	A relative unit gets sizing from either a parent, root element, or the size of the viewport.
Fluid Grids	The grid is an element with areas in columns and rows. A fluid grid can be set to auto-fit or be set manually to adjust across breakpoints.
Flexible Images	Images that can be served to the browser in different sizes depending on the image size and the resolution.

Fundamentals of Responsive Design

Relative Units	A relative unit gets sizing from either a parent, root element, or the size of the viewport.
Fluid Grids	The grid is an element with areas in columns and rows. A fluid grid can be set to auto-fit or be set manually to adjust across breakpoints.
Flexible Images	Images that can be served to the browser in different sizes depending on the image size and the resolution.
Breakpoints	Points in screen size (informed by media queries) where a website or app adjusts its layout.

Fundamentals of Responsive Design

Relative Units	A relative unit gets sizing from either a parent, root element, or the size of the viewport.
Fluid Grids	The grid is an element with areas in columns and rows. A fluid grid can be set to auto-fit or be set manually to adjust across breakpoints.
Flexible Images	Images that can be served to the browser in different sizes depending on the image size and the resolution.
Breakpoints	Points in screen size (informed by media queries) where a website or app adjusts its layout.
Media Queries	A CSS function that receives information about size from the viewer’s device to trigger the breakpoints in the design.

Fundamentals of Responsive Design

Relative Units	A relative unit gets sizing from either a parent, root element, or the size of the viewport.
Fluid Grids	The grid is an element with areas in columns and rows. A fluid grid can be set to auto-fit or be set manually to adjust across breakpoints.
Flexible Images	Images that can be served to the browser in different sizes depending on the image size and the resolution.
Breakpoints	Points in screen size (informed by media queries) where a website or app adjusts its layout.
Media Queries	A CSS function that receives information about size from the viewer’s device to trigger the breakpoints in the design.
Mobile-First	The design strategy of designing the mobile design first before designing for larger devices or resolutions. This is to push designers to prioritize smaller, more accessible mobile designs.

Helpful CSS

Box Model

Box Sizing

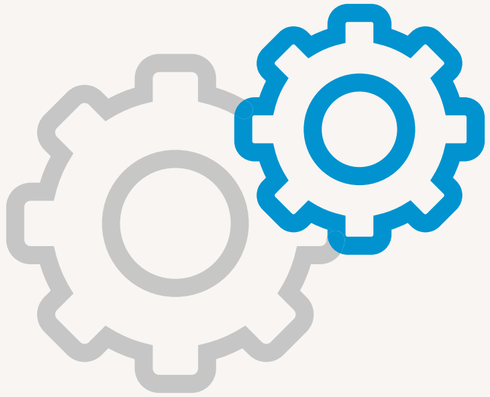
In general terms, this is essentially a box that wraps around every HTML element. It contains margins, padding, borders, and the content itself.

```
*, :after, :before {
  box-sizing: border-box;
}
```

By setting it to border-box you're telling the browser to account for any padding and border to the element's width and height.

width = border + padding + content width

height = border + padding + content height



Modern Implementation

Relative Units

em units

Beyond pixels, the em unit is an older unit of measurement that came from print. A unit that allows setting the font-size of an element relative to the font-size of its parent. For other properties it will be relative to the font-size of the current element.

```
.parent {
  font-size: 16px;
}
.child {
  font-size: 1.5em;
}
```

Relative Units

em units

Beyond pixels, the em unit is an older unit of measurement that came from print. A unit that allows setting the font-size of an element relative to the font-size of its parent. For other properties it will be relative to the font-size of the current element.

```
.parent {
  font-size: 16px;
}
.child {
  font-size: 1.5em;
}
```

This would make the child font-size 24px. (1.5 * 16px = 24px)

```
<div class="parent">
  I'm 16px
  <div class="child">
    I'm 24px
  </div>
</div>
```

Relative Units

rem units

AKA root em unit which is based on the font-size of the html element. This means that the parent elements are ignored and only the root value is considered.

```
html {
 font-size: 16px;
}
.parent {
  font-size: 14px;
}
.child {
  font-size: 2rem;
}
```

This would make the child font-size 32px. (2 * 16px = 32px)

```
<div class="parent">
  I'm 14px
  <div class="child">
    I'm 32px
  </div>
</div>
```

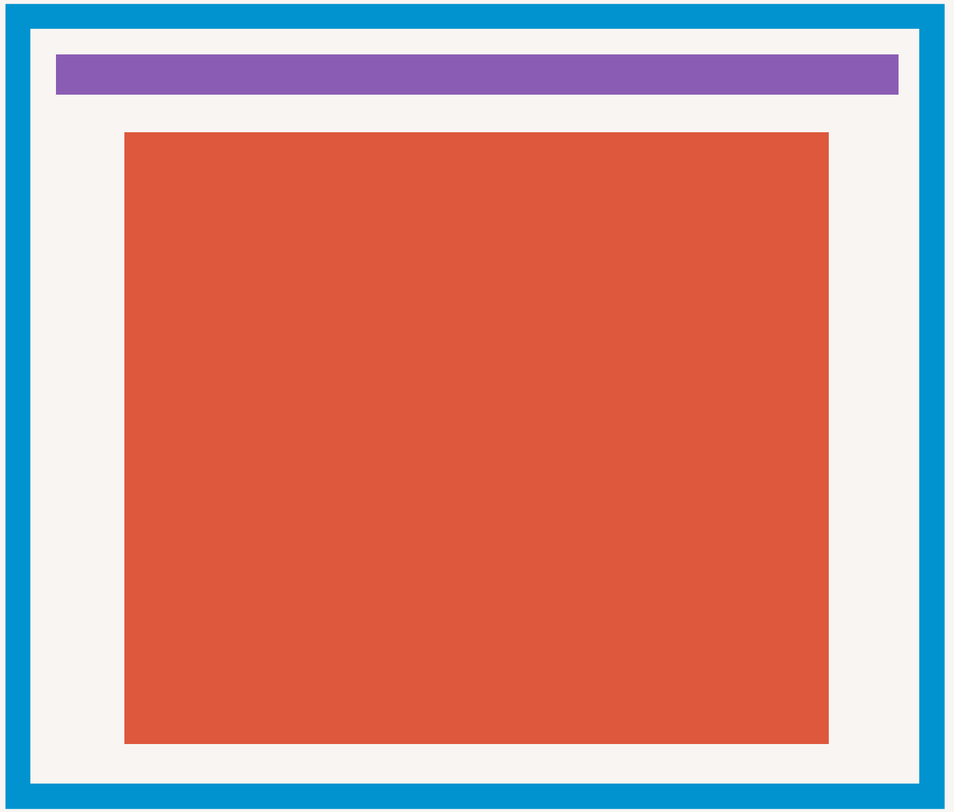
Relative Units

vw and vh

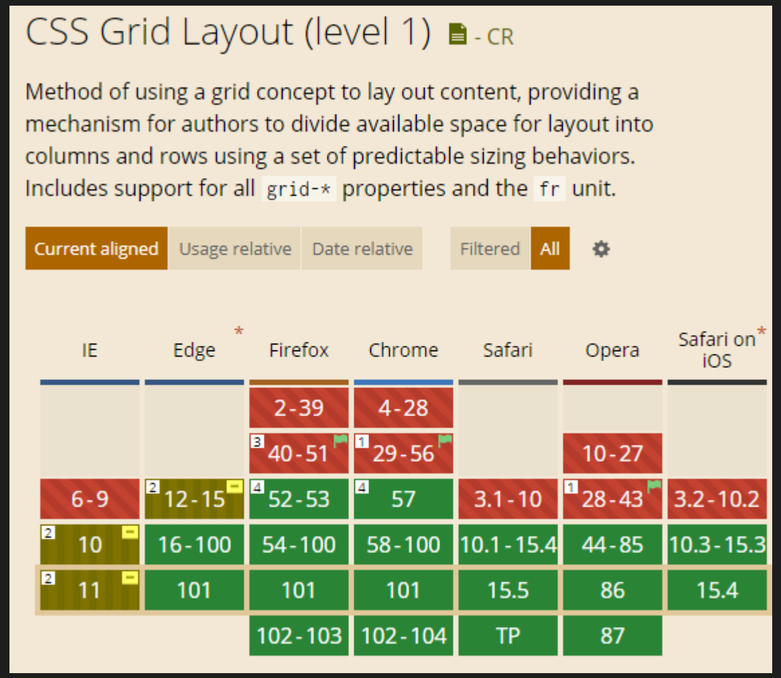
These are units relative to the viewport width and height. 1vw is equal to 1% of the viewport's width, and 1vh is 1% of the viewports height.

```
nav {
  height: 5vh;
}

main {
  width: 75vw;
}
```



Fluid Grids.



CSS Grid Layout (level 1) - CR

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all `grid-*` properties and the `fr` unit.

Current aligned | Usage relative | Date relative | Filtered | All

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *
		2-39	4-28			
		3 40-51	1 29-56		10-27	
6-9	2 12-15	4 52-53	4 57	3.1-10	1 28-43	3.2-10.2
2 10	16-100	54-100	58-100	10.1-15.4	44-85	10.3-15.3
2 11	101	101	101	15.5	86	15.4
		102-103	102-104	TP	87	

For years the only tools available for creating CSS layouts were custom grids that used floats and positioning.

Now modern layout methods are responsive by default, such as Flexbox and CSS Grid.

Fluid Grids.

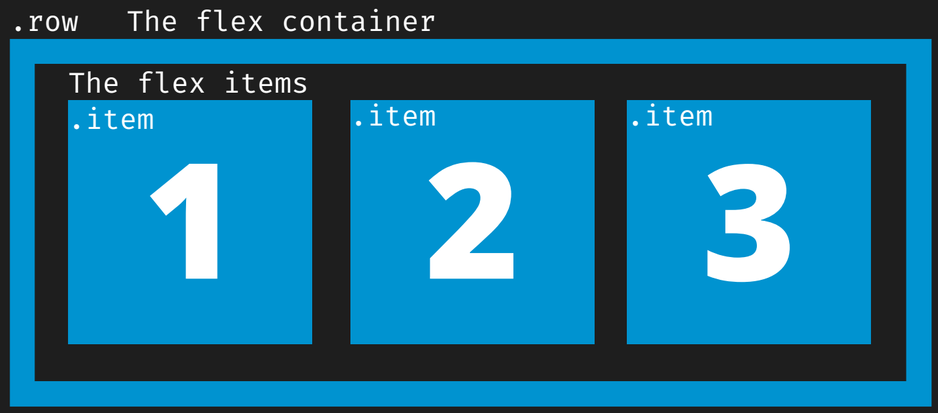
Flexbox

A set module of a number of properties that define parent and child items. Flex items will shrink or expand to distribute space.

Fluid Grids.

Flexbox

A set module of a number of properties that define parent and child items. Flex items will shrink or expand to distribute space.




```css
.row {
  display: flex;
  flex-flow: row wrap; /* Define the flow direction */
}
.item {
  flex: 1; /* Space in the container will be distributed equally */
}
```

Fluid Grids.

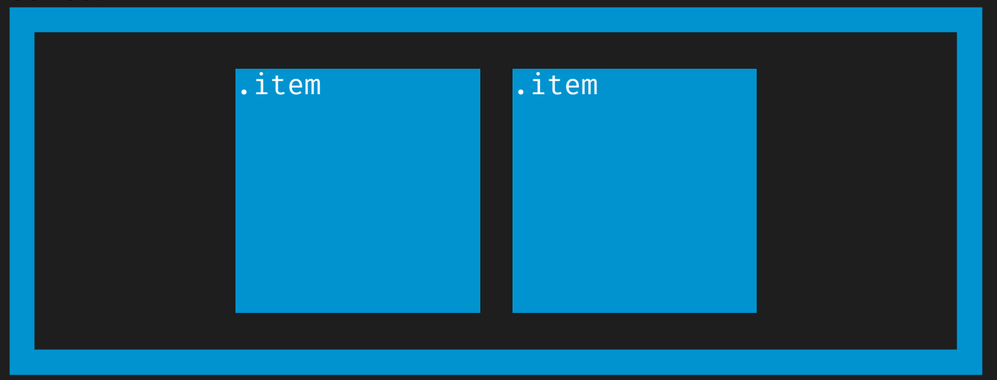
Flexbox / justify-content

```
.row {
  display: flex;
  justify-content: center;
}
```

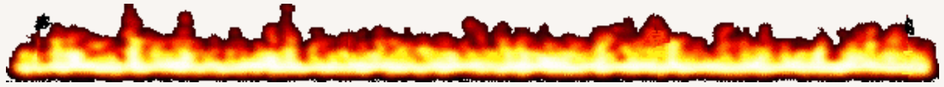
x-axis

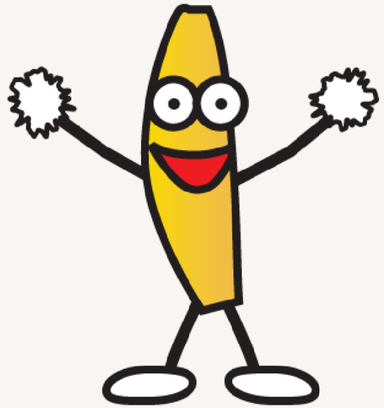
flex-start

center



You just Learned How to Center a Div!





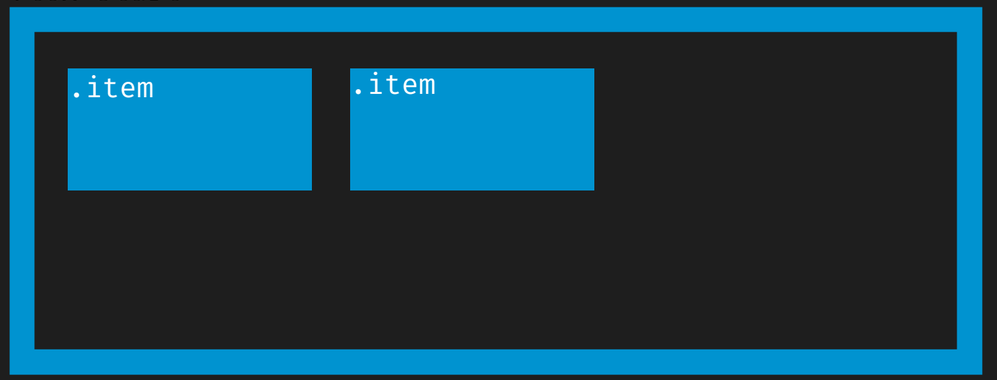
Fluid Grids.

Flexbox / align-content

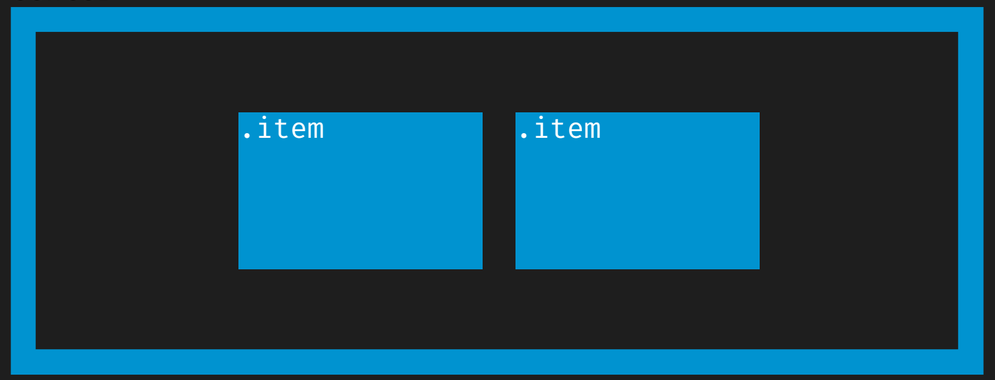
```
.row {
  display: flex;
  align-content: center;
}
```

y-axis

flex-start



center



Fluid Grids.

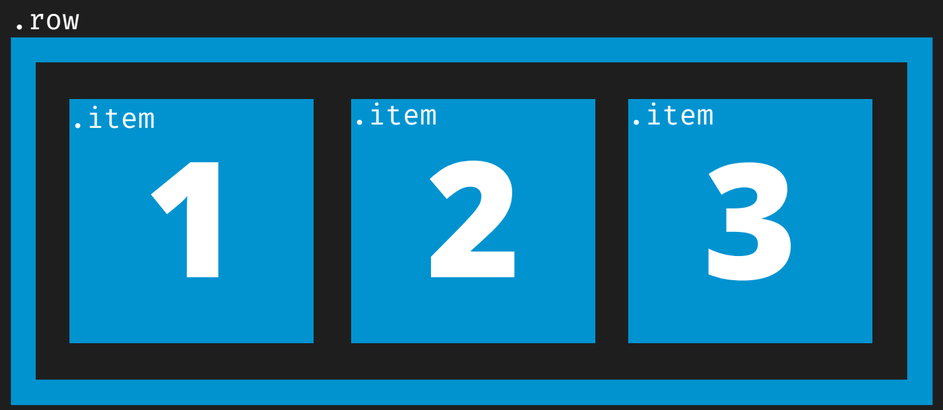
CSS Grid

CSS Grid was adopted by most browsers in 2017 and it was designed to work with other parts of CSS. Grid was designed for two-dimensional layout instead of one like Flexbox.

Fluid Grids.

CSS Grid

It uses containers(or rows) and columns which can be in any order.

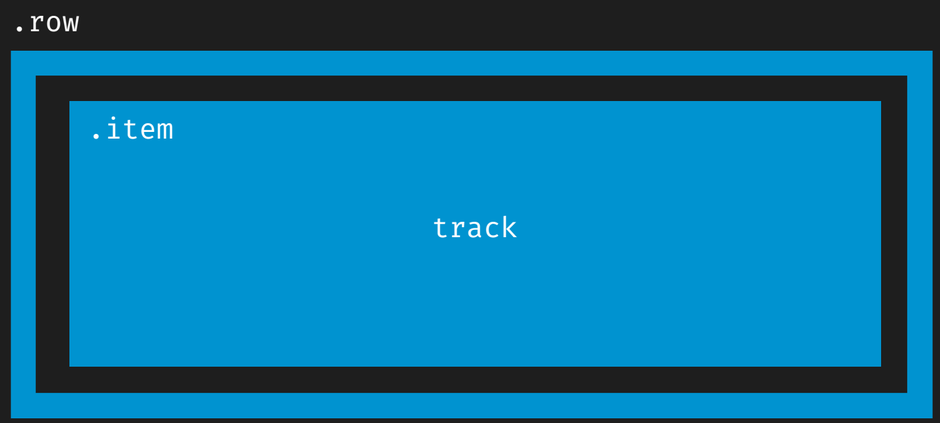



```
<div class="row">
  <div class="item">
  </div>
  <div class="item">
  </div>
  <div class="item">
  </div>
</div>
```

Fluid Grids.

CSS Grid / Tracks

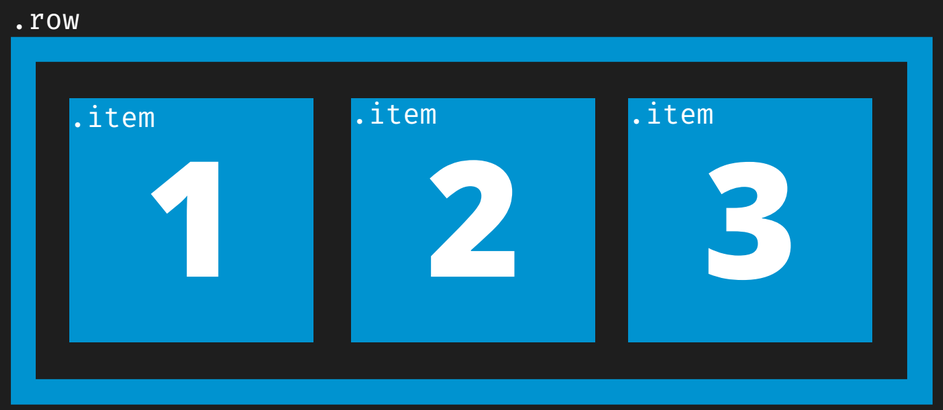
A track is the space between any two lines on the grid. The box below is an example of a track.




```
<div class="row">
  <div class="item">
  </div>
</div>
```

Fluid Grids.

CSS Grid / fr Unit

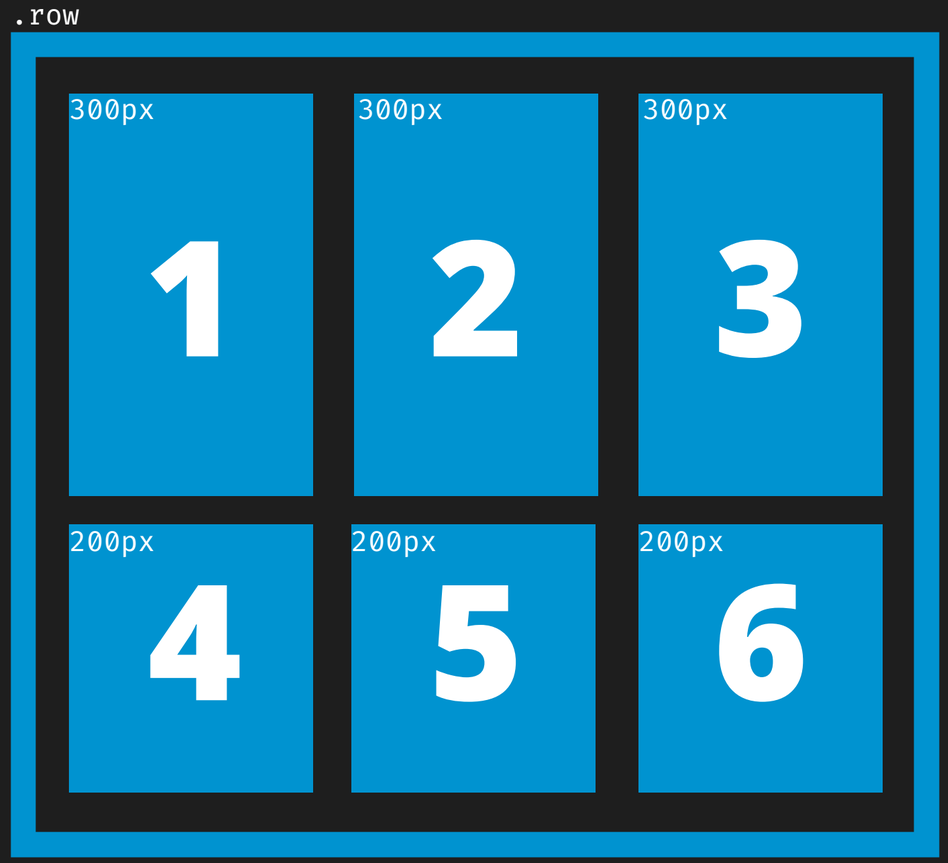



```
.row {
 display: grid;
 gap: 20px;
 grid-template-columns: 1fr 1fr 1fr;
}
```

A block level grid

Fluid Grids.

CSS Grid / minmax




```
.row {
 display: grid;
 grid-template-columns: repeat(3, 1fr);
 grid-auto-rows: minmax(200px, auto);
}
```

Flexible Images.

1. Images should adapt according to viewport size and screen resolution.
2. Use smaller images for smaller resolutions.

Image Element

```
<img
  alt="A tiny grey kitten"
  srcset="
    cat-sm.jpg  200w,
    cat-med.jpg  600w,
    cat-lg.jpg  1200w,
    cat-xl.jpg 2000w
  "
  sizes="50vmin"
>
```

```
sizes="(max-width: 599px) 200px,
       (max-width: 1199px) 600px,
       (max-width: 1999px) 1200px,
       2000px">
```

Flexible Images.

1. Images should adapt according to resolution.
2. Use smaller images for smaller resolutions.

Image Element

```
<img
  alt="A tiny grey kitten"
  srcset="
    cat-sm.jpg  200w,
    cat-med.jpg  600w,
    cat-lg.jpg  1200w,
    cat-xl.jpg 2000w
  "
  sizes="50vmin"
>
```

Picture Element

```
<picture>
  <source
    srcset="cat-zoomed-out.jpg"
    media="(min-width: 1200px)"
  />
  <source
    srcset="cat.jpg"
    media="(min-width: 600px)"
  />
  <img
    src="cat-zoomed-in.jpg"
    alt="Cute cat photo"
  />
</picture>
```

Media Queries & Break points

Media Query

Break point

```
@media (min-width: 300px) {
  h1 { font-size: 1.2rem; }
}

@media (min-width: 768px) {
  h1 { font-size: 1.5rem; }
}

@media (min-width: 1024px) {
  h1 { font-size: 2rem; }
}
```

PLEASE

Make that font size bigger.

What should we do instead?

For Desktop (in rems)

- Body text size of 1rem which equals 16px.
- H1s of about 2.5rem, or 40px.
- Other headers will be smaller. 2rem, 1.5rem, 1.2rem(your choice)
- Functional text can be reduced to 14px.

For Mobile

- Keep the body size the same. 16px is good for mobile.
- Decrease your header sizes.
- You will have to play around with what looks good for your app.
- Don't use tiny sizes!

Typography

Responsive

Font sizes that break according to specific breakpoints that developers set.

```
@media (min-width: 300px) {
  h1 {
  font-size: 1.3rem;
 }
}

@media (min-width: 768px) {
  h1 {
  font-size: 2rem;
 }
}
```

Typography

Responsive

Font sizes that break according to specific breakpoints that developers set.

```
@media (min-width: 300px) {
  h1 {
  font-size: 1.3rem;
 }
}

@media (min-width: 768px) {
  h1 {
  font-size: 2rem;
 }
}
```

Fluid Method

Elements that have a minimum value, preferred value, and a maximum value.

```
clamp([value-min], [value-preferred], [value-max]);


      h1 {
       font-size: clamp(1.5rem, 2.5vw, 2.4rem);
      }
```

Typography

Fluid Method

There are limitations to using fluid values like viewport width. It's not a catch-all for every problem. It doesn't address every accessibility issue you might come across. There's also a chance that users might not be able to scale text beyond 200%.

```
clamp([value-min], [value-preferred], [value-max]);

    h1 {
     font-size: clamp(1.5rem, 2.5vw, 2.4rem);
    }
```

Future Implementation

New and The Future / Container Queries

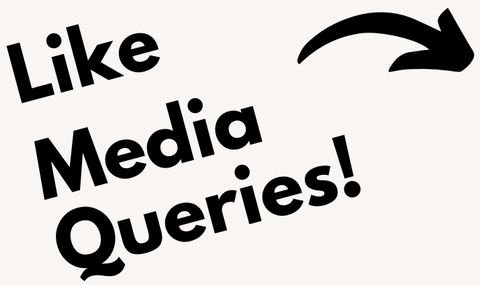
Container Queries are similar to media queries, but only pertain to the size of the container. This will allow you to change the layout or styling based on the size of an item, and not the viewport.

Container Type	Defines an item as a query container. Descendants can query aspects of its sizing, layout, style and state.
Container Name	Specifies a list of query container names for @container rules to use to filter which query containers are targeted
Query Units	Introduces new units of measure to cover a container's width, height, and different sizes. These are cqw, cqh, cqi, cqb, cqmin, cqmax

New and The Future / Container Queries

```
.container {
 container-type: inline-size;
 container-name: left-card;
}
```

```
.main-content {
 display: flex;
 flex: row;
 gap: 20px;
}
```




```
@container left-card (width > 300px) {
  .main-content {
    flex-row: reverse;
  }
}
```

Responsive Frogs

https://github.com/eegans/responsive-frogs

placekitten

A quick and simple service for getting pictures of kittens for use as placeholders in your designs or code. Just put your image size (width & height) after our URL and you'll get a placeholder.

Like this: `http://placekitten.com/200/300`
or: `http://placekitten.com/g/200/300`

Delivered by Mark James & Photographers (Image attribution) — Inspired by placehold.it

CUPCAKE IPSUM

SUGAR-COATED LOREM IPSUM GENERATOR

TIRED OF HOW BORING LOREM IPSUM GOT ?

How about using auto-generated text that will actually make people love your project even more?

Pretty sweet, right?

5 PARAGRAPHS

LONG MEDIUM SHORT

START WITH "CUPCAKE IPSUM DOLOR SIT AMET"

GIVE YOUR TEXT SOME "LOVE"

GENERATE

Supported by Basia Madej

Let's take a look at some code!

Responsive Frogs

https://github.com/eegans/responsive-frogs


THANK
YOU