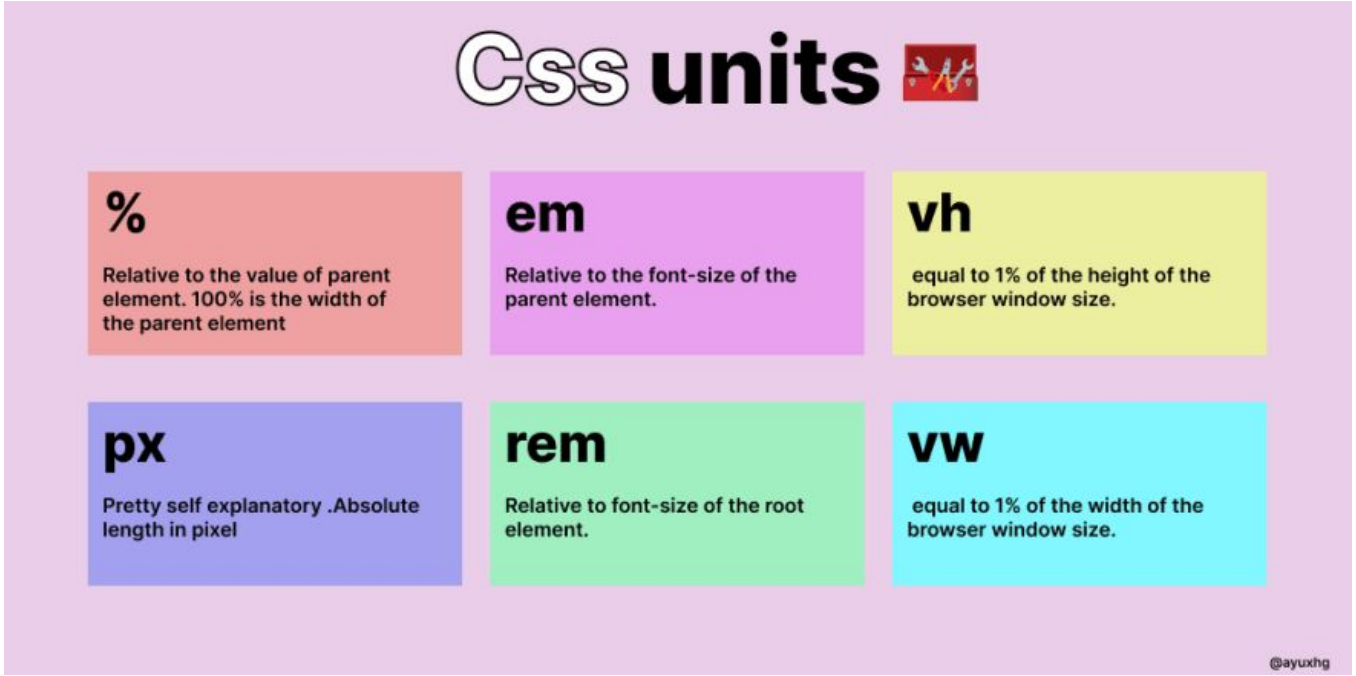
Web development basics

Css properties, Responsive Design and Animations

CSS Measurements



CSS Position

Static: This is the **default** value of the position property, and it means that the element is positioned according to the **normal flow** of the document. Elements with position: static **are not affected** by the **top**, **right**, **bottom**, and **left** properties.

Relative: This value positions the element **relative to its normal position** in the document flow. Setting top, right, bottom, or left properties will move the element in the **respective direction from its original position**.

Absolute: This value positions the element relative to its nearest positioned ancestor (ancestor elements with a position value **other than static**). If there is no positioned ancestor, then the element is positioned relative to the **initial containing block** (usually the <html> element).

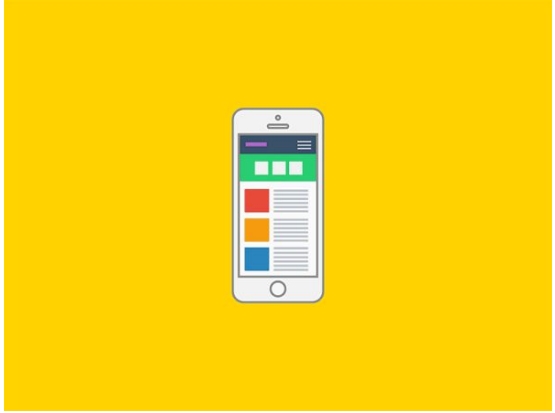
Fixed: This value positions the element **relative to the viewport** (the visible area of the browser window). Fixed elements **are always visible**, even when the user **scrolls the page**.

Sticky: This value is a **hybrid of relative and fixed positioning**. A sticky positioned element is positioned **relative to its nearest positioned ancestor or the viewport**. However, it behaves like a fixed element when it **reaches a certain threshold** (specified with top, right, bottom, or left), and then it sticks to that position.

CSS Pseudo-Classes

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

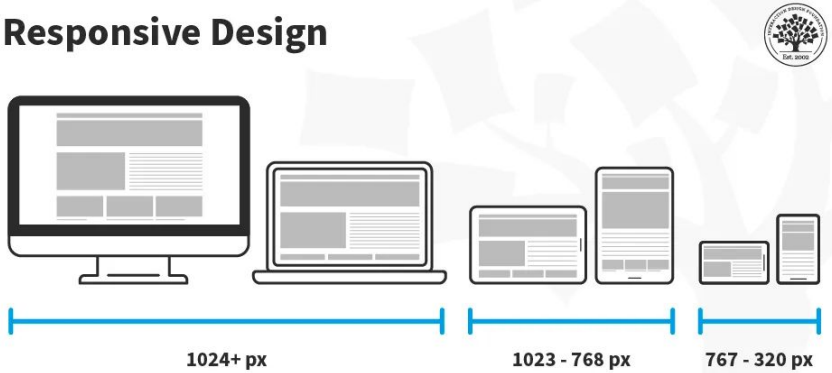
Responsive Design and Animation



Responsive design ensures that a website or application looks and functions correctly on a variety of devices, such as laptops, tablets, and smartphones. This is important because more and more people are accessing the internet on mobile devices, and a website that is not optimized for mobile may be difficult to navigate or use.

Animation can help to guide the user through the interface and make the experience more engaging. For example, animation can be used to highlight important information, provide feedback on user interactions, and create a sense of hierarchy on the page. Additionally, animation can make a website or application feel more dynamic and alive, which can lead to a more positive user experience.

What is Responsive Design?

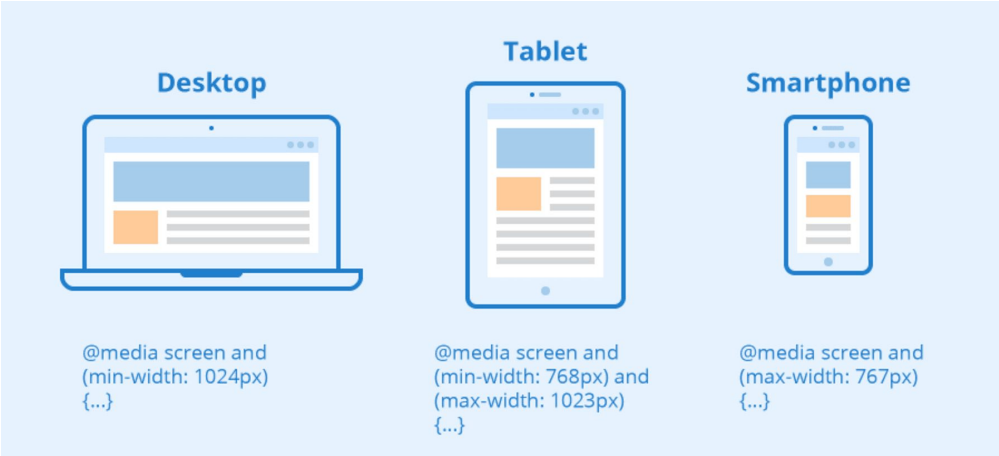


Responsive design is a method of designing and developing websites and applications that adapt to the size of the user's screen. This allows the website or application to look and function correctly on a variety of devices, such as laptops, tablets, and smartphones.

One key aspect of responsive design is the use of media queries, which are a way to apply CSS styles based on the characteristics of the device. Media queries allow designers to create different layouts for different screen sizes, so that the content can be rearranged and resized to fit the screen.

CSS media queries

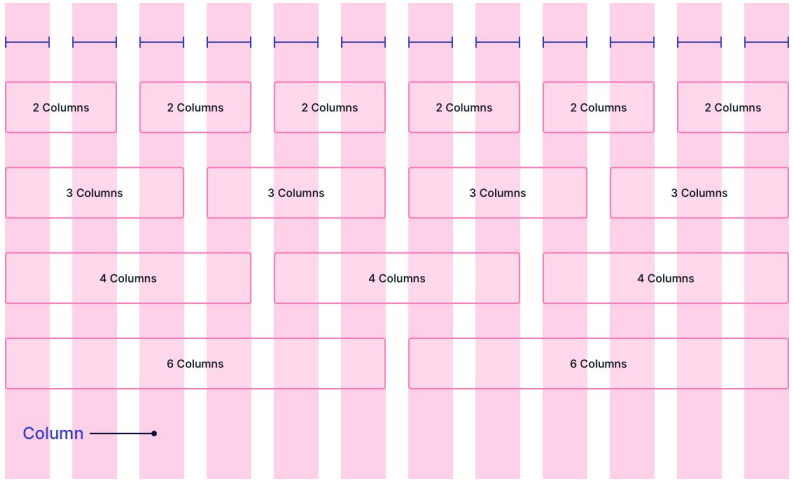
By using media queries, designers can create websites and applications that adapt to the size of the user's screen, ensuring that the content is always easy to read and navigate, regardless of the device being used.




```
@media (condition) {
  /* CSS styles that will be applied when the condition is true */
}
```

```
@media (max-width: 600px) {
  /* CSS styles that will be applied when the viewport is less than 600px
*/
}
```

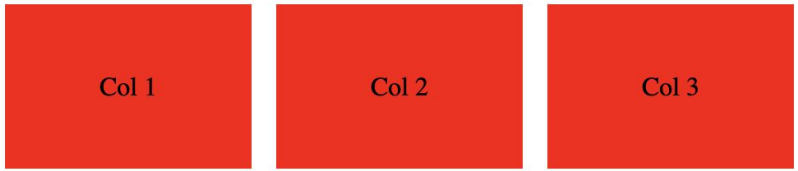
Fluid grids

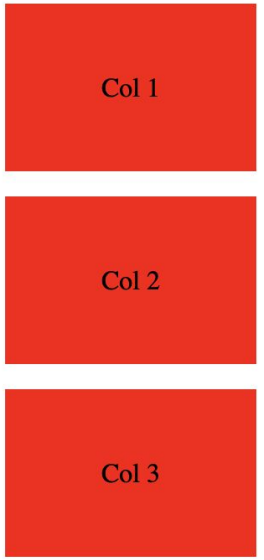


Another important aspect of responsive design is the use of fluid grids, which are a way to create layouts that automatically adjust to the size of the screen. This is done by defining the layout in terms of relative units, such as percentages, rather than absolute units, such as pixels.

This allows the layout to automatically adjust to the size of the screen, so that the content can be rearranged and resized to fit the screen. Together, media queries and fluid grids allow designers to create websites and applications that adapt to the size of the user's screen, ensuring that the content is always easy to read and navigate, regardless of the device being used.

Using Media Queries to create Responsive Design






```
<div class="wrapper">
  <div>Col 1</div>
  <div>Col 2</div>
  <div>Col 3</div>
</div>
```

Creating Smooth Transitions with CSS

CSS transitions allow you to smoothly change the values of CSS properties over a given duration. They are used to add visual effects and make changes in the layout of a webpage look more natural.

https://www.w3schools.com/css/css3_transitions.asp

```
<button class="my-button">Hover over me</button>
```

```
.my-button {
  background-color: blue;
  transition: background-color 0.5s;
}

.my-button:hover {
  background-color: red;
}
```

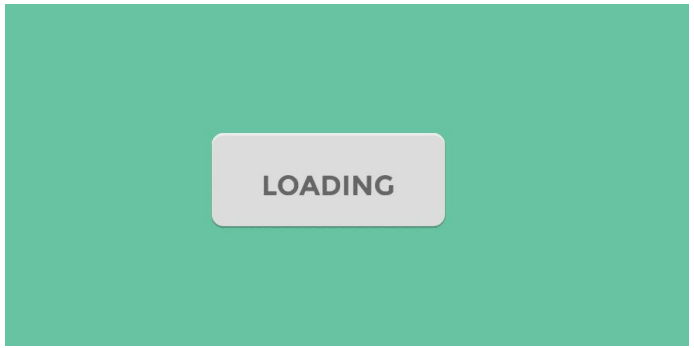
Introducing Animation to your website

Animation is the process of creating the illusion of motion by rapidly displaying a sequence of static images, called frames. When these frames are played back in quick succession, the human eye perceives them as a continuous motion. Animation can be used to create engaging and interactive user interfaces by providing visual feedback, guiding the user's attention, and creating a sense of motion and dynamism.

Animation use case examples

Animation can be used in a variety of ways to enhance the user experience. For example, an animation can be used to:

- Provide visual feedback when a button is clicked or a form is submitted
- Guide the user's attention by highlighting important elements on the page
- Create a sense of motion and dynamism, making the interface feel more alive and engaging
- ...



Creating Animations with CSS

```css
/* Define the keyframes for the animation */
@keyframes fadeIn {
  0% {
    opacity: 0; /* Start with the element invisible */
  }
  100% {
    opacity: 1; /* End with the element fully visible */
  }
}

/* Apply the animation to the element */
.my-element {
  animation: fadeIn 2s ease-in;
}
```

```css
/* Define the keyframes for the animation */
@keyframes rotate {
  0% {
    transform: rotate(0deg); /* Start with the element in its original position */
  }
  100% {
    transform: rotate(360deg); /* End with the element rotated 360 degrees */
  }
}

/* Apply the animation to the element */
.my-element {
  animation: rotate 2s linear infinite;
}
```

Animations task

Task: Create a simple animation that makes a square move across the screen horizontally.
Steps:

1) Create the HTML structure for the square. In the body of your HTML file, add a div with the class "square"
2) In your CSS file, give the square a size and a background color
3) Define the keyframes for the animation. In your CSS file, create a new @keyframes rule called "move" that defines the starting and ending positions of the square:
4) Apply the animation to the square. Using the animation property, specify the animation name, duration, and timing function:
5) Test your animation by running the HTML file in a browser. The square should move horizontally across the screen from left to right over a duration of 2 seconds, using a linear timing function.

Animations task explanation

```html
<div class="square"></div>
```

```css
.square {
  width: 50px;
  height: 50px;
  background-color: blue;
  animation: move 2s linear;
}
```

```css
@keyframes move {
  0% {
    transform: translateX(0);
  }
  100% {
    transform: translateX(300px);
  }
}
```

Tailwind CSS

Tailwind CSS is a utility-first CSS framework that makes it easy to build responsive, consistent, and highly customizable user interfaces. It provides a set of pre-defined CSS classes that can be used to quickly add styling to your HTML elements. One of the main advantages of Tailwind CSS is that it uses a set of utility classes that are designed to be composable. This means that you can easily build complex layouts and designs by combining multiple classes together.

https://tailwindcss.com/

Tailwind CSS 2

For example, if you want to create a button with a blue background and white text, you could use the classes "bg-blue-500" and "text-white" to achieve this:

```
<button class="bg-blue-500 text-white px-4 py-2">Click me</button>
```

In this example, "bg-blue-500" sets the background color to blue and "text-white" sets the text color to white. The "px-4" and "py-2" classes set the padding on the x and y axis respectively.

Tailwind CSS - Responsiveness

Tailwind CSS also provides a set of responsive classes that allow you to easily create responsive designs. These classes allow you to specify different styles for different screen sizes, so you can ensure that your design looks good on all devices. For example, you can use the class "sm:text-lg" to make text larger on small screens:

```
<h1 class="text-lg sm:text-xl">Welcome to my website</h1>
```

Tailwind CSS practice

Task: Create a simple webpage that displays a card with an image, title, and description.

Steps:

1) Create the HTML structure for the card. In the body of your HTML file, add a div with the class "card"
2) In your CSS file, include the Tailwind CSS styles. You can either include the pre-built CSS file that you can download from the Tailwind website or you can use the npm package
3) Use Tailwind CSS classes to style the card. To style the card, you can use the classes "w-full" and "bg-white" to set the width and background color of the card, respectively

Tailwind CSS practice 2

4) Style the image, title, and description. To style the image, you can use the class "w-full" to make the image take up the full width of the card. To style the title and description, you can use the classes "text-lg" and "text-gray-700" to set the font size and color

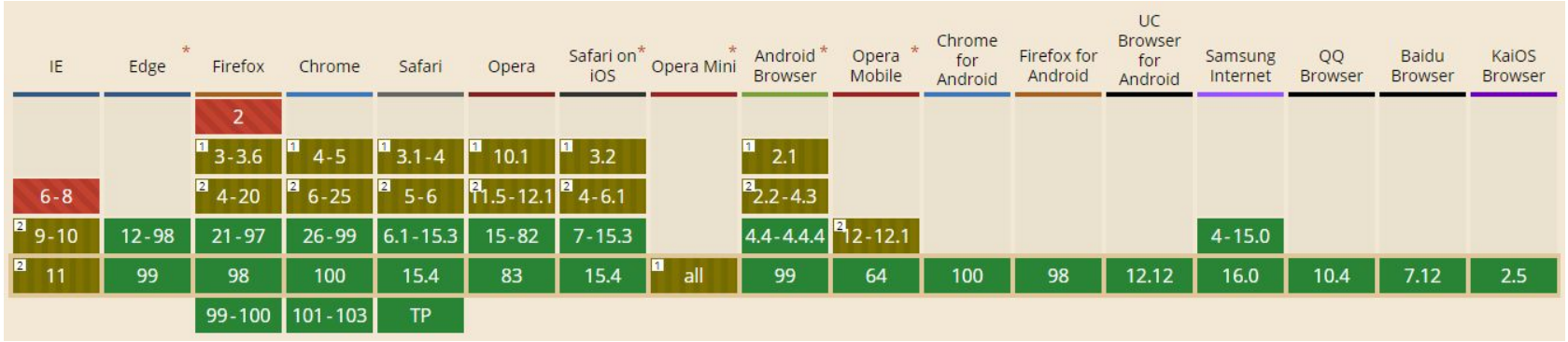
5) Add spacing to the card. To add padding and margin to the card, you can use the classes "py-4" and "px-6" to add padding on the y and x axis respectively and "my-4" and "mx-6" to add margin on the y and x axis respectively

6) Test your design by running the HTML file in a browser. The card should have a white background, the image should take up the full width of the card, and the title and description should be correctly styled

Tailwind CSS practice explanation

```
<div class="card w-full bg-white py-4 px-6 my-4 mx-6">
  <img src="image.jpg" alt="" class="w-full">
  <h2 class="text-lg font-medium">Card Title</h2>
  <p class="text-gray-700">Card Description</p>
</div>
```

Can I use it?



IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
		2														
		1 3-3.6	1 4-5	1 3.1-4	1 10.1	1 3.2		1 2.1								
6-8		2 4-20	2 6-25	2 5-6	2 11.5-12.1	2 4-6.1		2 2.2-4.3								
2 9-10	12-98	21-97	26-99	6.1-15.3	15-82	7-15.3		4.4-4.4.4	2 12-12.1				4-15.0			
2 11	99	98	100	15.4	83	15.4	1 all	99	64	100	98	12.12	16.0	10.4	7.12	2.5
		99-100	101-103	TP												

Homework

1) Add responsiveness and transitions to created portfolio.
    a) Play around with animations
    b) Refactor application to use tailwindcss instead

Portfolio ideas - https://bashooka.com/html/free-html-css-portfolio-web-design-templates/