



CSS Positioning

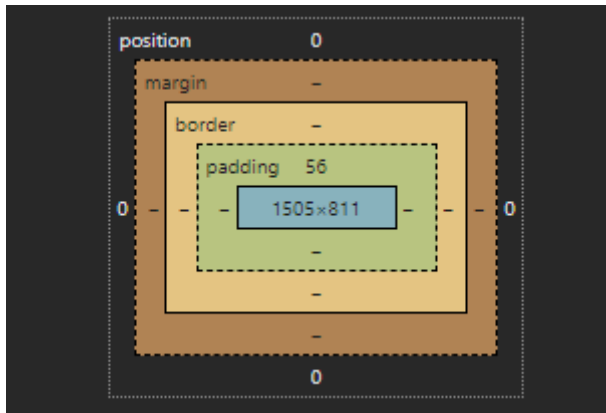


LESSON OBJECTIVES

- 
- Understand the Box model
 - Implement positioning with grid and flexbox efficiently
 - Understand and implement media queries
- 

Box model

Refers to a box that is wrapped around every element and it contains: the content, padding, border, margin.

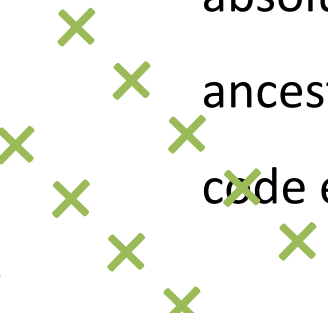
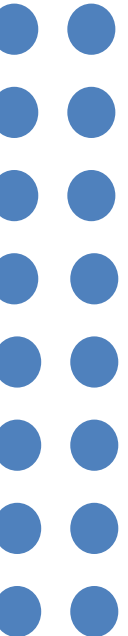


Positioning

You are able to change the position of an element. This means you can take the element out of the normal document flow, and change where it is displayed.

- static - default position of an element.
- relative - position will be based on it's original position(Have to set top, left, bottom and right properties).
- sticky - An element with position as sticky is positioned based on the user's scroll position.
- fixed - The element stays in the exact same place no matter how much the user scrolls.
- absolute - An element with position as absolute is positioned relative to the nearest positioned ancestor.

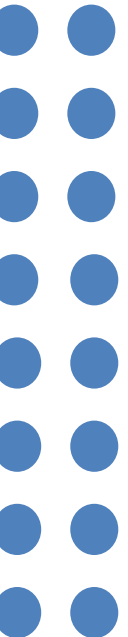
code example to follow...



Flexbox

The Flexbox Layout aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex"). The flex layout is to give the container the ability to alter its items' width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes).

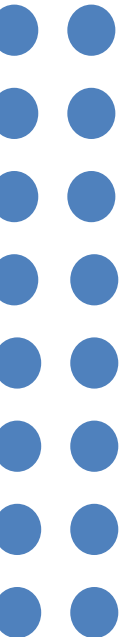
```
Parent-element{  
  display: flex;  
}
```



Grid

CSS Grid is a 2-dimensional layout system, meaning that it can control both columns and rows. You can work with the grid layout system by applying CSS rules to both the parent element (Grid Container), and direct children elements (Grid Items).

```
Parent-element{  
  display: grid;  
  grid-template-columns: repeat(4, 1fr) ;  
}
```

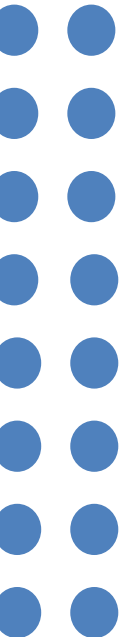


Media Queries

A media query limits the scope of CSS styles to only apply when certain media conditions are met.

The values for the min-width and the max-width are called break points.

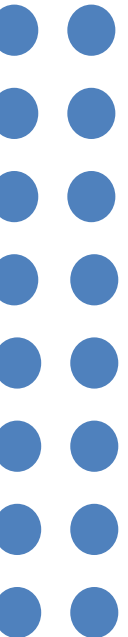
```
@media (min-width: 12345) {  
    /* CSS properties and values */  
}  
@media (max-width: 6789px) {  
    /* CSS properties and values */  
}
```



Animations & @keyframes

The animation property in CSS can be used to animate many other CSS properties such as color, background-color, height, or width. Each animation needs to be defined with the @keyframes at-rule which is then called with the animation property.(See shorthand syntax below)

```
.element {  
  animation: pulse 5s infinite;  
}  
@keyframes pulse {  
  0% {  
    background-color: #001F3F;  
  }  
  100% {  
    background-color: #FF4136;  
  }  
}
```



Animations (depth)

- animation-name: declares the name of the @keyframes at-rule to manipulate.
- animation-duration: the length of time it takes for an animation to complete one cycle.
- animation-timing-function: establishes preset acceleration curves such as ease or linear.
- animation-delay: the time between the element being loaded and the start of the animation sequence
- animation-direction: sets the direction of the animation after the cycle. Its default resets on each cycle.
- animation-iteration-count: the number of times the animation should be performed.
- animation-fill-mode: sets which values are applied before/after the animation.
- animation-play-state: pause/play the animation.

