Section 8 advanced CSS

<https://www.udemy.com/course/the-complete-web-development-bootcamp/learn/lecture/37368052#overview>

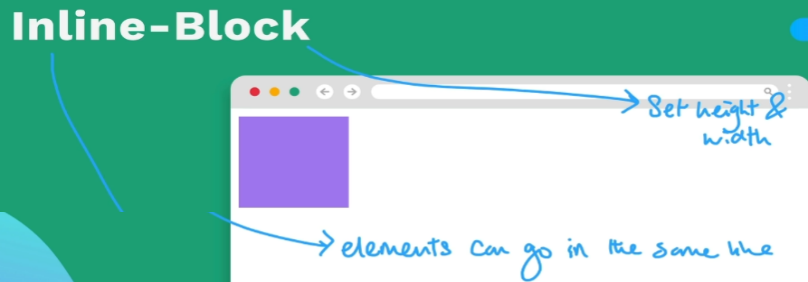
48 CSS Display property inline, block , inline-block

DIV vs Span

Div – by default its display is block

Span – by default its display is inline – note that height stays the same and just gets longer

Inline-block allows for same line and height and width



Display none makes the element disappear from the website

Demo web site

<https://appbrewery.github.io/css-display/>

<https://developer.mozilla.org/en-US/docs/Web/CSS/display>

The **display** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) property sets whether an element is treated as a [block or inline box](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flow_layout) and the layout used for its children, such as [flow layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flow_layout), [grid](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout) or [flex](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flexible_box_layout).

Formally, the **display** property sets an element's inner and outer *display types*. The outer type sets an element's participation in [flow layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flow_layout); the inner type sets the layout of children. Some values of display are fully defined in their own individual specifications; for example the detail of what happens when display: flex is declared is defined in the CSS Flexible Box Model specification.

Syntax – lots of possible values

/\* precomposed values \*/

display: block;

display: inline;

display: inline-block;

display: flex;

display: inline-flex;

display: grid;

display: inline-grid;

display: flow-root;

/\* box generation \*/

display: none;

display: contents;

/\* multi-keyword syntax \*/

display: block flex;

display: block flow;

display: block flow-root;

display: block grid;

display: inline flex;

display: inline flow;

display: inline flow-root;

display: inline grid;

/\* other values \*/

display: table;

display: table-row; /\* all table elements have an equivalent CSS display value \*/

display: list-item;

/\* Global values \*/

display: inherit;

display: initial;

display: revert;

display: revert-layer;

display: unset;

**49 CSS Float** –

Wrapping text around an element – not commonly used any more this may be the only use case in modern web design

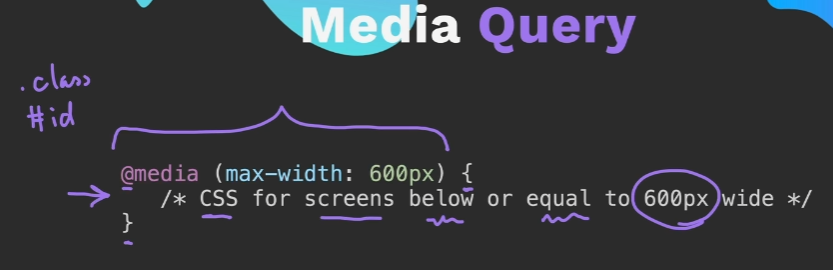
**50 Creating Responsive Websites**- can use Media Queries, CSS Grid, CSS Flexbox or an external framework like Bootstrap



**Media query**

The @symbol then the CSS selector (what is inside the { } in the css when adding styles ect)

Instead of a . or # using the @ symbol then in the parans define the break point or below or equal to 600 px wide in this case



CSS Grid



Define layout of columns and rows

Here 1fr is dividing the screen into 2 eq columns .

3 rows here defining with px



Gap is space between the rows and columns



Define the span of the columns –

The first div will span 2 columns 

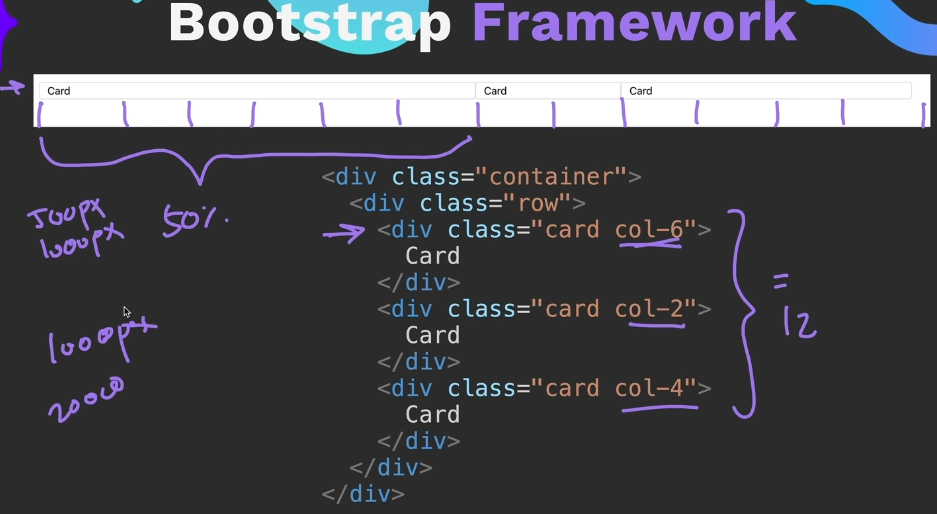
CSS Flexbox -responsive



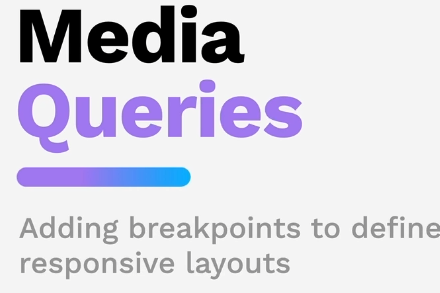
Flex of 2 has twice the width of the .card, 0.5 has half

Could also flex the height

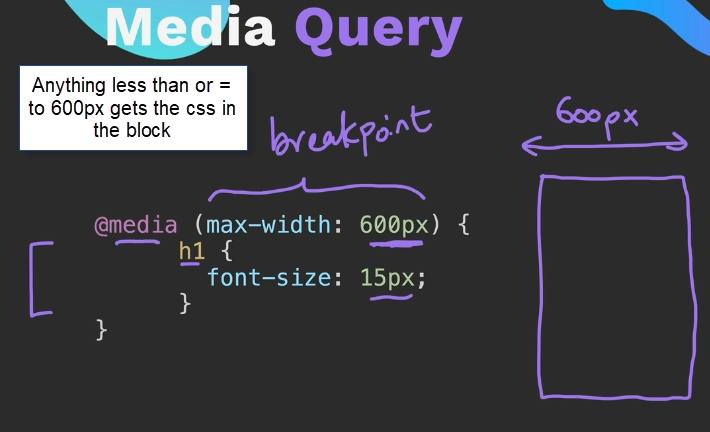
**Bootstrap** - a framework – responsive

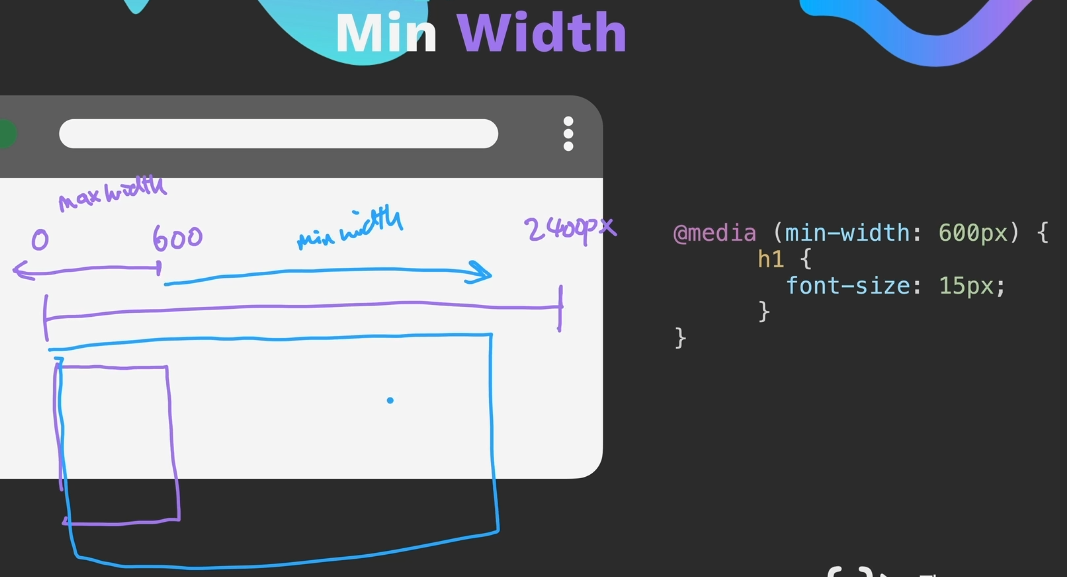


**51 Media Queries**

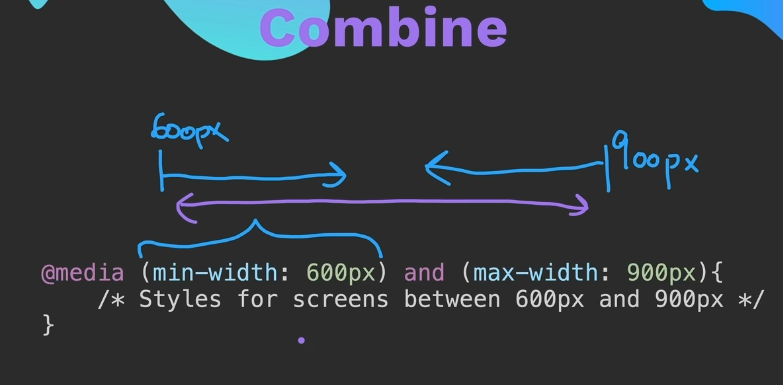


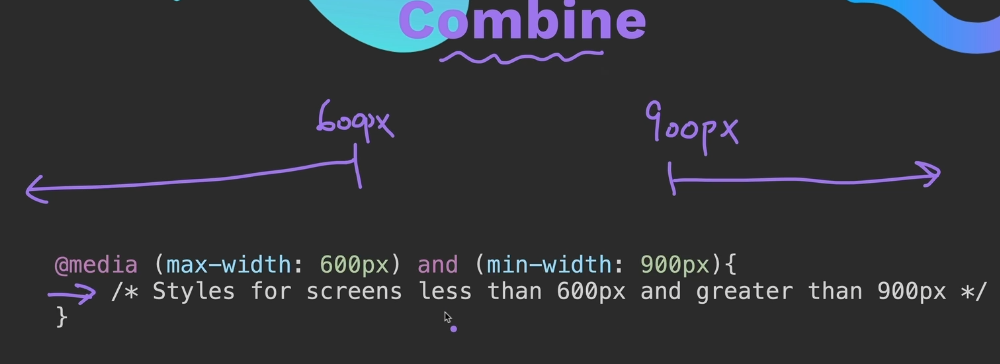
Up to a max width of 600 change font size



If the media query was at min-width then it would be 600 px and up apply the style 

Can combine break points

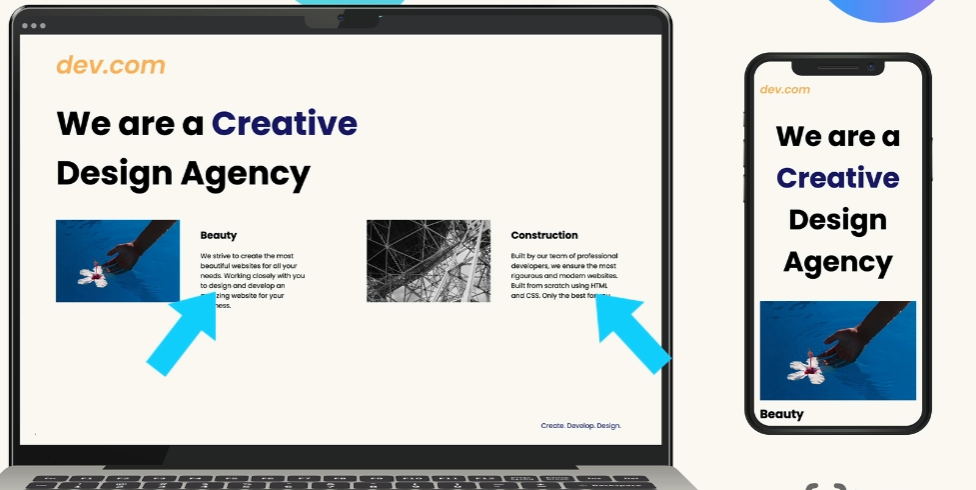




<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries>

52 Project Web Design Agency Website

Make this – responsive , stack cards on top of one another on devices

-

Restart here

<https://www.udemy.com/course/the-complete-web-development-bootcamp/learn/lecture/37368196#announcements>