**Basics** (https://youtu.be/wRNinF7YQqQ)

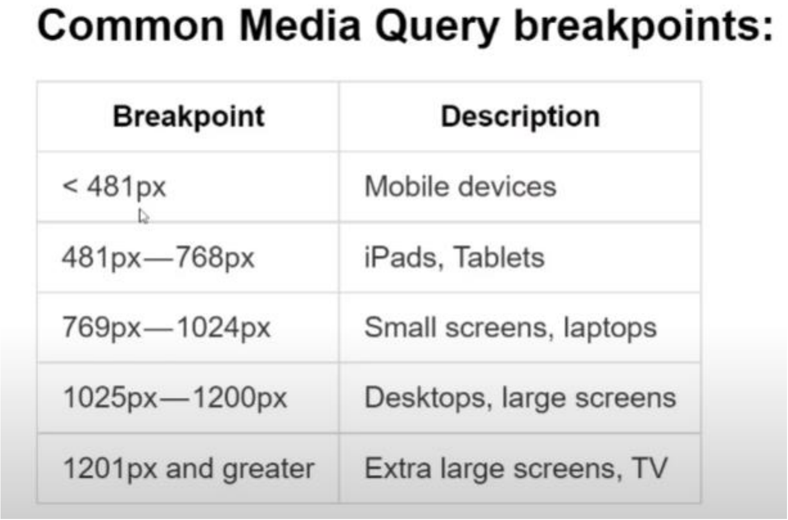
- Fonts : Fonts.google for *font.family* (put multiples for backup)

Change font style, weight and *text-decoration* use rem for text for adaptative text size

- Border : -style, -radius, -width (can be specified with -top etc)
- Background : linear-gradient, -repeat + -attachement
- Margin : space around/outside the thing vs Padding = space around the content/inside
- Float : position element in regard to the rest of the content (it wraps around). *Clear* “clears” the float relatives to floating elements
- Position : static (is pinned in the page), relative (to the page ; it covers other elements if collision, have a reserved space), absolute (is a ghost over the page and is positioned from the last static element, a parent /!\), fixed (relative to the viewport), sticky (normal then fixed until scrolled passed, its then position in the viewport depend on top etc props, stay within its container)
- Pseudo classes : it’s a state (like it’s a link, it was visited etc). *nth-child*(**nb**) or **3n** for each 3n element & :focus. (::marker)
- Shadows : text-shadow (can have multiples shadows separated with ,)
- Icons : Font awesome
- Transform : translate, rotate, scale, skew (& matrix that combine all)
- Animation : *@keyframes* Name ( *from{ … ;} to{ … ;}* or *100%{ }*, *50%* etc) with animation : Name + -iteration-count, -delay, - duration, -timing-function (can be in a single prop) => combine with pseudo classes

**Medium** (https://youtu.be/OXGznpKZ_sA)

- <u>Sizes :</u> rem are for relative to root font size, very useful. We can use width : 40ch to limit a paragraph to 40characters length. vw & vh = viewport width&height -> use % instead
- <u>Box :</u> box-sizing : content-box/border-box. Enclose in a box all text.
- <u>Typography :</u> text-transform (uppercase), text-align (justify), text-indent : 2em (indent first line !), line-height (1.5), letter/word-spacing
- <u>Display :</u> block = block, inline, flow, flex, grid, none ; inline-flex
- <u>Column :</u> existe only if enough space (else put as many columns as possible), -rule (the |), -gap, break-inside : avoid, break-before : column (shrink page when shrinking to much)



**Common Media Query breakpoints:**

| Breakpoint | Description |
|---|---|
| < 481px | Mobile devices |
| 481px—768px | iPads, Tablets |
| 769px—1024px | Small screens, laptops |
| 1025px—1200px | Desktops, large screens |
| 1201px and greater | Extra large screens, TV |

**Notes**

- calc( ) : calculate sum/mult etc
- aspect-ratio
- *{ } : for all content (with padding,margin : 0 reset everything)
- clamp(min,preferred,max)    (for font e.g.)
- flexbox : display-> flex then flex-direction row or column to flex-basis & flex-wrap ; flex-flow combine direction and wrap. -basis (item size), -grow/-shrink (repartition the space between items)
- justify-content : align items in the direction set by flex-direction. See also space-between, space-around, space-evenly.
- gap : between items
- filter : apply a filter to an image (blur, grayscale, contrast etc) (see also backdrop-filter)
- grid (columns & rows)
- content : replaces an element with another value (ex : 🏆at the start of a <p>) can be url, image-set, counter, open-quote etc
- attribute selector : a[title] { } for all elements with a title or a[class~= ‘logo’] for elements with a class that contains ‘logo’. **~= : contains word, *= : contains string, $= ; ends with.**
- attr( ) :  retrieve the value of an attribute of the element (ex : data-count, cite, src url, tittle, data-count number, 0 ; here the 0 is a fallback option)
- image-set : provide a set of images for alternative image option, sizing or format (ex : image-set(url(“…“) 1x, url(“….“) 2x) )
- tooltip : extra information shown when hovering over button (uses the visibility : hidden/visible prop)
- :focus-within : pseudo-class if focus of an element inside it
- :focus : triggered with click, tap or tab
- ::before : pseudo-class of the first child of the element (used to add content : ). Similar for ::after
- min-height
- counter( ) : -> string of the value of the counter named. Used with -increment & -reset & -set
- white-space : normal, nowrap, pre-line, break-space etc
- detecting dar/light theme : @media (prefers-color-scheme: light){ :root { --bg-color : ….}}
- center div : display : grid & place-content : center
- clip-path : polygon(stuff); transition : clip-path 1s;

Sites :

https://specificity.keegan.st/

https://fontawesome.com/

https://cubic-bezier.com/#.17,.67,.83,.67

https://www.uplabs.com/

https://developer.mozilla.org/en-US/docs/Web/HTML/Element

https://live.browserstack.com/dashboard#os=iOS&os_version=13.0&device_browser=safari&zoom_to_fit=true&full_screen=true&url=https%3A%2F%2Fsupport.google.com%2Faccounts%2Fanswer%2F61416&speed=1

https://gtmetrix.com/reports/m-waharte.great-site.net/nX9L9cnz/

https://gtmetrix.com/blog/how-to-eliminate-render-blocking-resources/

Examples :

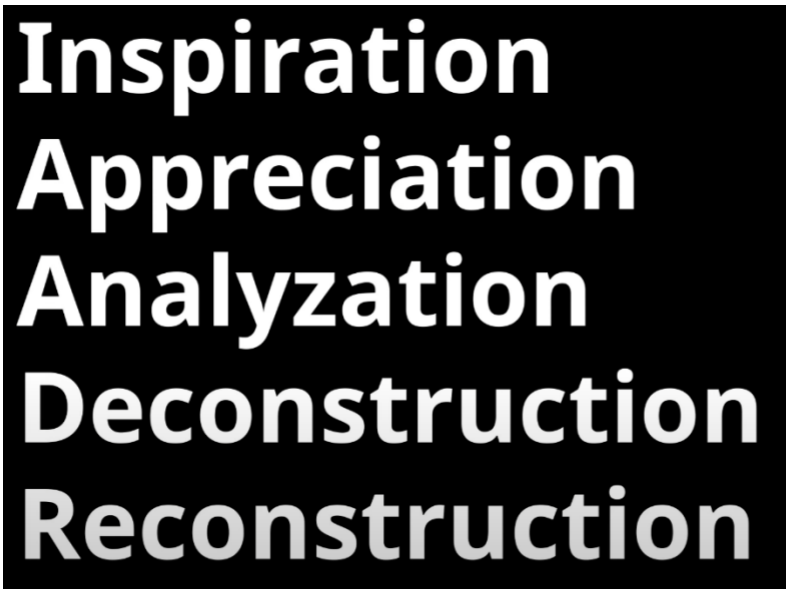
https://youtu.be/6HFpw5fcaD8

https://fireship.io/snippets/basic-css-dropdown-menu/

**Design guidelines**

60-30-10 rule : 60 = bg color, 30 = main color, 10 = active/interactive color

HOW TO START:



Inspiration

Appreciation

Analyzation

Deconstruction

Reconstruction