CSS

inline-block vs inline vs block: Compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element. Also, with display: inline-block, the top and bottom margins/paddings are respected, but with display: inline they are not. Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

Reset vs Normalize: reset removes all default styles while normalize keeps some necessary styling.

For modals - <dialog> element with .showModal() (for modals), or, .show() for non-modals(allows interaction with rest of page) - places dialog element on top layer of stacking context.

transform: translateZ(-1px); - (will put element behind).

Fixed position will be affected by parent that has a transform. Can movie move it under another parent.

Sticky containing block is parent, fixed is viewport.

Absolute positioning uses padding box not content box as its containing box.

Flexbox flex-basis keep default of auto and use width unless changing direction - Will also use height.

Use position: relative or translate instead of absolute when possible.

Use height: 100svh or 100dvh for mobile devices (they have address bar at top)

Height: 100vh;

Height: 100svh; (in case not supported)

Use percentages for font-size: 30%; for responsiveness

width: clamp(20rem, 40%, 50rem);

Responsive Design: Start simple (mobile) then add complexity (desktop)

Use mobile first and min-width for media queries

GRID

For a structured layout (even columns)

* grid-template-columns: repeat(auto-fit, minmax(min(22rem, 100%), 1fr)):

For a fluid intrinsic layout:

To keep width of elements:

* display: flex;
* flex-wrap: wrap;
* gap: 1rem;

To allow elements to grow add:

* flex-parent > \* {
* flex: 1;}

Can make a (Carousel) row of grid columns that are scrollable: See youtube videos ‘Useful & responsive layouts, no media queries’ 5:00 min – also – ‘Can I create Netflixs's video carousel with CSS only’

display: grid;

place-items: center: (for vertical and horizontal)

grid-auto-columns: 1fr; Will create equal columns based on smallest grid- layouts-areas

grid-auto-rows will do the same.

border-inline, border-block, padding & margin

margin-inline: auto; instead of margin: 0 auto; (won't affect top or bottom).

For text, titles, headings, etc. spacing will be related to font size.

flex-parent > \* {

margin-block-start: 1em; }

gap: min(2rem, 5vw);

font-size: clamp(1rem, 5vw + 0.5 rem, 1.5rem);

all: unset; - Will act as inherit or initial to reset back to original state.

Button { outline-color: transparent } instead of outline: none for users with high contrast mode (will still see if hover, focus).

inset – shorthand for positioning

inset: o auto auto 10px;

To center an element

inset: 0;

margin: auto;

###

Flex and Grid can use z-index without positioning. Z-index will work even if the position is set to static (default).

HTML Entities – Easier to use than svgs or icon fonts.

BEM: Can have blocks nested within blocks and elements nested within elements. Elements have no stand alone meaning. Modifier – is a different version(s) of a block or element.

Isolation:isolate – creates new stacking context.

box-shadow: can be used to create borders, that can even be

aspect-ratio: lets you create boxes that maintain proportional dimensions where the height and width of a box are calculated automatically as a ratio.

Using opacity and background-blend-mode for effects on background image.

clamp() min() max()

Grid - place-content:center

attr() function

Grid line names work well with media queries that rearrange the Grid layout.

Positioning elements: margin, translate, absolute.

Excellent Grid formatting techniques – Building the Header Part 1 & Part 2.

Between 1100 (1140) and 1200 pixels for standard fixed width.

Cannot use css variables in media queries but can use sass variables.

Ex. flex: 0 0 70% - when a percentage for basis is given grow and shrink are usually 0.

Very useful! - Flexbox items – Using margin auto to create a margin space between items.

Z-index – Only works if position is defined . Can be position:relative.

& > \* - To select direct children instead of selecting each one separately. & > \*:hover

Flexbox is perfect for aligning and centering elements.

A svg sprite file contains all of the svgs so only one http request is needed.

Svg should be used instead of icon fonts. Icon fonts are really a hack which are like images using a font. If it fails it then just displays something not useful. Screen readers cannot read icon fonts.

Root has higher specificity than html.

Css custom properties need to be defined in a scope- usually the root scope.

Css variables vs sass variables: can use in calc function, can access in Javascript, can edit in devtools, cascade and can be inherited.

Overflow: hidden will fix a lot of overflow problems such as an image overflowing a shape.

If elements have position:absolute and want to float them, such as in a media query for a smaller screen, then need to change them back to position: relative before floating.

Inside of media queries ems and rems are not affected by root font size but instead by font size of browser. Media queries use font size of browser. Best to use ems instead of rems in media queries.

Icon fonts are vectors

Icon fonts are treated as text

Control D – Multiple selects

Browser animations optimized for only 2 properties - opacity and transform. But transform can handle most animations.

backface-visibility:hidden – Removes shake on animation.

display: inline-block – Treated as text – can center with text-align:center.

::after Pseudo element treated as a child of actual element or initial state.

animation-fill-mode: backwards – starts at 0%.

BEM – Prevents nesting thus keeps specificity low

7 / 1 CSS Architecture

4 Basic Responsive Design Principles

1. Fluid Layouts
   1. Use %, vw, vh
   2. Use max-width instead of width
2. Responsive Units
   1. Use rem instead of px for lengths
3. Flexible Images
   1. Always use %
   2. Use max-width instead of width
4. Media Queries / Breakpoints

1140 Pixel width grid is standard?

Using max-width instead of width allows width of element to adapt to viewport width

Can simulate table with css with display:table and display:table-cell instead of using html table element.

**Natours**

6 bth1 17:20 clip path8 ccca 11:20 animation options

8 coca 11:50 removing animation shake

10 bca 11:30 animation-fill-mode:backwards

34 bcgwf 16 :not(:last-child)

35 btas1 12 background-clip

color:transparent

37 btas3 13 outline-offset

38 btfs 16:00 background-clip with icon fonts (text)

39 btts1 8:20 perspective (rotating card) backface-visibiliy:none

39 btts2 9:30 background-blend-mode:

39 btts2 11:40 overflow:hidden – hides child content that overflows parent boundaries

39 btts2 12:50 clip-path: polygon

40 btts2 23:00 box-decoration-break:clone

41 btts3 15:15 overflow:hidden not working because of clip-path

Start 41 btts3 4:20

42 btss1 6:15 & 14:45 figure element

42 btss1 9:40 shape-outside

42 btss1 17:20 transform:skewX

42 btss1 18:20 handling multiple transforms

43 btss2 :55 figure element

43 btss2 7:10 backface-visibiliy:none fixing another type of problem (try using if having a problem with translations or animations)

43 btss2 9:45 filter: blur() brightness ()

44 btss3 background video

44 btss3 9:45 object-fit:cover

44 btss3 12:20 overflow:hidden

45 btbs 9:20 linear-gradient with angle in degrees and image below.

45 btbs 10:30 linear-gradient with percentages, multiple settings and solid gradients and transparent color.

46 btbs2 16:40 opacity:0, visibility:hidden.

49 btn1 27:20 – using inline-block to fix formatting problems.

50 btn2 1:00 – setting width to zero to remove from page. Opacity: 0 – items stillon page.

50 btn2 8:30 – Cubic Bezier for transition

51 btn3 – Hamburger Menu Icon

51 btn3 12:30 – transform-origin

52 bapcp 13:45 – display: table, display: table-cell, vertical-align: center

52 bapcp 18:30 - column-count:#, column-gap:#

53 bapcp2 2:00 – difference between display: none, opacity:0, visibility: hidden. Can also use width:0 or position element out of window.

53 bapcp2 4:45 – :target pseudo class

57 wmq 2:15 – media queries modification order

Start 54 or 57?

**Trillo**

75 bth1 3:40 icomoon svg icons

77 bth3 all review user nav section

77 bth3 9:45 using flex on span elements to vertically and horizontally align text in them

79 bth3 – before active hover menu animation

80 btho 14:00 - margin-right:auto – for an item in flex container (can also use margin-left: auto for adjacent item)

81 btho2 2:30 – fixing svg element acts like inline element with small space below

81 btho2 5:40 – Use of current color

81 btho2 5:40 – How to make generic global button

81 btho2 9:00 – Pulsating animation

81 btho2 12:15 – align- self: stretch

81 btho2 13:20 – flex alignment tricks

82 btds1 3:00 – flex basis

Start btho 80 11:20 - coding

**Nexter**

109 btfs2 3:25 extents

109 btfs2 16:40 auto-fit minmax

112 bths1 8:30 html sup command

114 btg1 5:45 vw used for gallery row height

114 btg1 11:40 object-fit: cover – have to specify width and height

Start 115 btg2 - coding

**General Notes**

margin: 0 auto; - center a block element inside a block element. Or, margin-left: auto and margin-right:auto.

Set image display: block to get rid of space below.

text-align: sets the horizontal alignment of the content inside a block element or table-cell box. It works like vertical-align but in the horizontal direction.

Browsers are optimized for 2 animation properties – opacity and transform.

Removes animation shake - background-visibility: hidden.

Cannot use -1 for end of implicit, must be explicit.

Practice linear gradients.

object-fit: cover – same as – background-img: cover. Specify on image block and have a parent element. Need to specify height and width on image.

Positioning schemes: 1) Normal flow 2) Floats 3) Absolute positioning

CSS Custom Properties advantages over SASS Variables

1. Can use in calc
2. Are cascaded and are inherited
3. Javascript can reference them
4. Are scoped to element declared
5. Usually defined in root for inheritance purposes

calc() function can mix unit types.

Vendor prefix comes before actual property.

Position absolute affects parent height same as float but there is no clearfix for it. Instead set height of child same as parent.

shape-outside: requires float and height and width.

Floated element – best way to move is with transform: translate

object-fit:cover – like background-size:cover but for html elements

overflow:hidden – very useful on parent element.

Background-size: 100% same as cover?

opacity:0, visibility:hidden – opacity:0 (item is hidden but still on page). Visibility:hidden (item is hidden and not on page – but cannot animate visibility.) 53 bapcp2 2:10 Q. How does display: none fit in? Also, transform: scale(0). Also, can set width:0 to remove element from window. Also, color:transparent to make invisible. Or, move element off of page. Or move out of parent and set overflow: hidden.

Position:absolute with parent position: absolute ok too

z-index – only works if position is defined on element.

Specifying width on elements fix many strange behaviors

svg elements act like inline elements with small space below.

Can remove small space under inline element by setting line-height and font-size to 0?, or by specifying flex on parent.

Html entities - &ndash; &times; https://alligator.io/html/html-entities-you-need/

Create columns with same height - display: table, display: table-cell, vertical-align: center

vertical-align: baseline | 10px | sub | super | top | text-top | middle | bottom | text-bottom | initial

Media Query Order: 1) Base + Typography 2) General Layout & Grid 3) Page Layout 4) Components