



Modern CSS

Lebleu Corentin - Serdar Luka -
Juri Joseph - Marinkovic Sanja



History of CSS

1996

CSS 1

First official CSS specification

1998

CSS 2

Expanded capabilities including positioning and media types

Browser Wars

Struggles with inconsistent browser implementations

2004

CSS 2.1

Improved stability and browser compatibility

2005

CSS 3

Introduction of modules including selectors, box model, and backgrounds

Pre-Modern CSS

2009

Flexbox

The flex layout allows responsive elements within a container to be automatically arranged depending on viewport size

2011

Grid Layout

Complex responsive web design grid layouts

Transitions and Animations

Animation of HTML document elements

2012

Custom Variables

Values that can be reused throughout the document

Modern CSS (features in groups)

- A. Units
- B. Color and Effect
- C. Pseudo-Elements
- D. Pseudo-Classes
- E. Fonts
- F. Functions
- G. Custom Properties
- H. Layout Properties
 - I. Layers and Scope
 - J. Container Queries

A. Units

1. inline/block size
2. vi
3. vb
4. vh
5. svh
6. lvh
7. dvh

C. Pseudo-Elements

17. ::slotted
18. ::part

B. Color and Effect

8. rgb(), rgba(), hsl(), hsla(), lch(), oklab(), oklch(), hwb(), lab()
9. conic-gradient()
10. backdrop-filter
11. ::backdrop
12. :modal, show() and showModal()
13. :picture-in-picture
14. :placeholder-shown
15. relative color vs. color-mix()
16. accent-color

D. Pseudo-Classes

19. :has(), :where() and :is()

E. Fonts

- 20. `override-colors()`
- 21. `font-tech()`, `font-format()`
- 22. `initial-letter`
- 23. `font-variations-setting`

G. Custom Properties

- 26. `@property`
- 27. `border`
- 28. `aspect-ratio`
- 29. `overscroll-behavior`
- 30. individual transform properties

F. Functions

- 24. `min()`, `max()`, `clamp()`
- 25. `var()`

H. Layout Properties

- 31. `subgrids-` `div`, `dl`, `dt`, `dd`
- 32. `margin-inline`, `margin-inline-end` and `margin-block`
- 33. `inset`, `inset-inline`, `inset-block`
- 34. `gap`, `grid-template-columns`, `grid-template-rows`

I. Layers and Scope

35. @layers

36. @scope

J. Container Queries

37. @container

38. container size : @container (min-width, orientation, aspect-ratio, ...)

39. container style : @container style()

40. naming containers

41. container units : cqw, cqh, cqi, cqb, cqmin, cqmax

Units: inline vs block

*taking into account the writing mode(horizontal and vertical)

Width and height vs. inline-size and block-size

- width: actual width, regardless of the writing mode
- height: actual height, regardless of the writing mode
- inline-size: logical width, respecting writing mode
- block-size: logical height, respecting writing mode

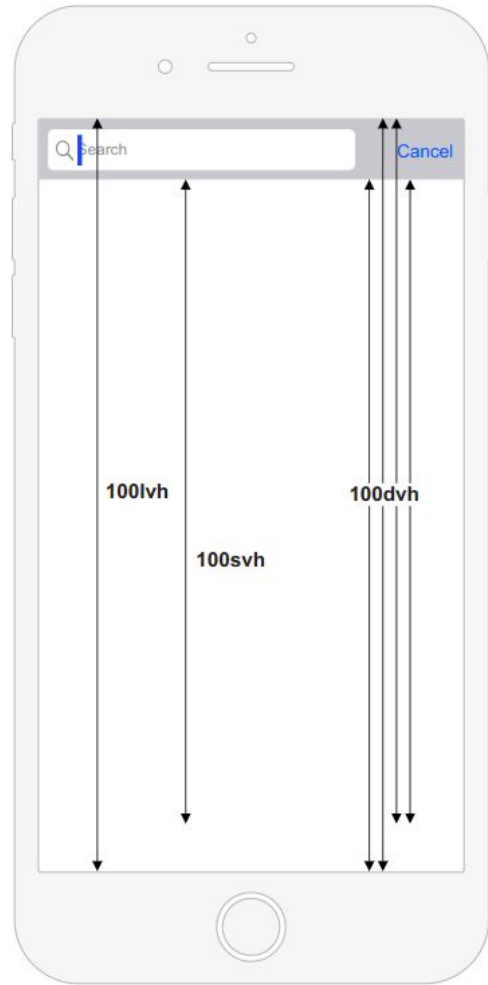
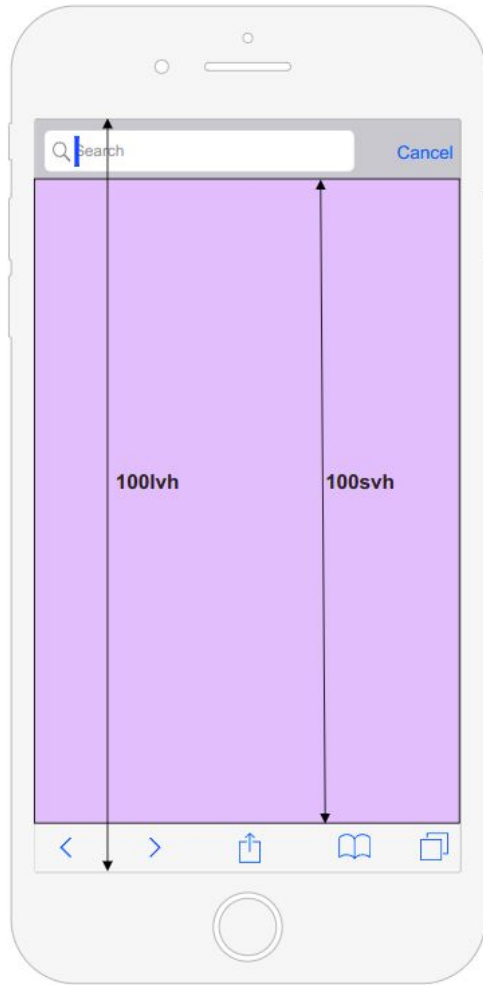
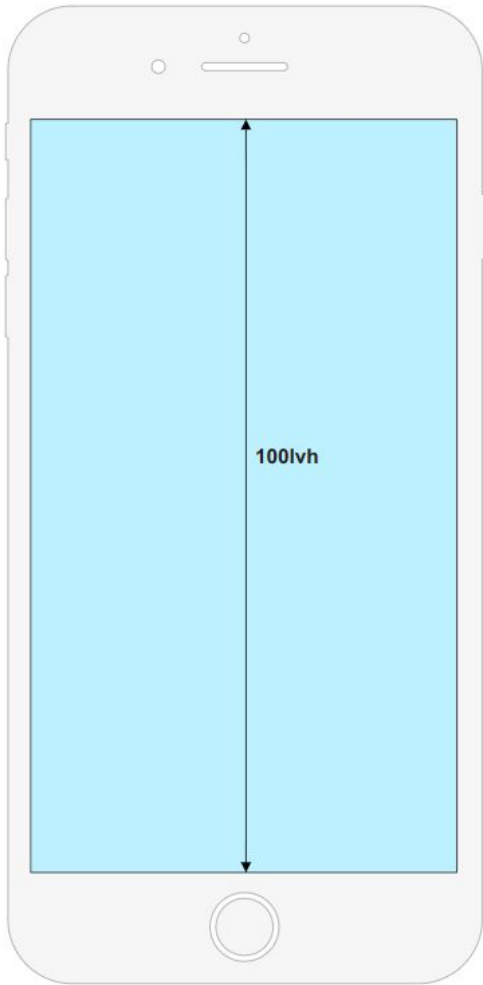
vw&vh vs. vi&vb

- vi: inline width (width in horizontal, height in vertical)
- vb: inline height (height in horizontal, width in vertical)

Showcase video: <https://youtu.be/3tzb4B5uScM?si=RO4rgSAan7MLZyyD>

Units: viewport

- **vh** does not respect if user interface elements are expanded or not
- **lvh** for large viewport
- **svh** for small viewport
- **dvh** dynamically matches either svh or lvh



Color spaces

- hwb (background-color: hwb(113 1% 61%))
 - Hue, Whiteness, Blackness
- lab (background-color: lab(75% 75 75))
 - lightness, red/green, yellow/blue
- oklab (background-color: oklab(59.69% 0.1007 0.1191))
 - red/green and yellow/blue does not exceed 0.5
- lch (background-color: lch(50% 132 130))
 - lightness, chroma, hue
- oklch (background-color: oklch(40.1% 0.123 0.11))
 - chroma and hue does not exceed 0.5

Color functions

- Relative colors
- Color-Mix
- Accent-color

Showcase videos:

1. Relative colors: https://youtu.be/eu6SYt-t6l0?si=B42z87GsrxnSdq_L
2. Color-Mix: <https://youtu.be/Q0yV1I5wvao?si=GbxEH1x9S1oalAdR>
3. Accent-color: https://youtu.be/HJOBf1IfMlo?si=DWt4_AUzgrwVJLnD

:has() / :is()

- Previously done in JavaScript
- Changes styles accordingly
- :has()
 - Parent selector
- :is()
 - Evaluates selector

Showcase videos:

1. has(): https://youtu.be/UeGKxMW5Ltk?si=yFWa_aLoiouXWJ_3
2. is(): <https://youtu.be/OqwVGkUSWe0?si=LWtosWhJwOvSKceQ>

The CSS cascade

- Cascading

- relevance (@media, ...)
- origin and importance (origin: browser/author, !important)
- specificity
- order of appearance (last is applied)

- Specificity

- ID (1-0-0): id selectors **#id**
- Class (0-1-0) : class selectors **.class** class attribute selectors **[att=""]**
pseudo-class **:hover**
- Type (0-0-1) : type selector **p, h1, etc** pseudo-elements **::before**

```
#content (1-0-0)
.wrapper h1 (0-1-1)
div p h1 (0-0-3)
div.wrapper p h1.title (0-2-3)
```

Layers

- Managing specificity in CSS
- A way to apply multiple styles to an element
- The order of the styles is determined by the cascading order
- !important can override the cascading order
 - but not recommended

```
@layer defaults, components, utilities;

@layer defaults {
  a:is(:link, :visited) {
    /* low-priority layers are easy to override by default */
    color: mediumvioletred;

    /* but take priority with importance */
    text-decoration: underline !important;
  }
}

@layer utilities {
  /* high-priority layers win by default */
  .maroon {
    color: maroon;
  }
}
```

Showcase video: https://youtu.be/x4DtDLJ0fk?si=tMg_cobSaix3b2k0

Container Queries

Container Size Queries :

- based on size of the parent container
- @container <container-condition>
- max/min-width/height, width/height, orientation, aspect-ratio, inline/block-size
- needs “container-type” on the container (inline-size, size, normal)
- can have a name : container-name
- new container units : cqw, cqh, cqi, cqb, cqmin, cqmax

Container Style Queries :

- based on computed style of the parent container
- @container style(<style>)
- any styles
- every element is a style container by default

Showcase video: <https://youtu.be/rZRdqw7-Tno?si=clnt3ZD4J-DK3fCe>

Further reading

- CSS Days
 - <https://cssday.nl/2023>
- News and Updates
 - <https://moderncss.dev/>

References and Sources

- Wikipedia (<https://en.wikipedia.org/wiki/CSS>)
- W3C (<https://www.w3.org/Style/CSS/Overview.en.html>)
- 100 Days Css Challenge (<https://www.matuzo.at/>)
- Mozilla Dev (<https://developer.mozilla.org/en-US/>)
- Chrome Dev (<https://developer.chrome.com/docs/devtools/>)
- CSS Days (<https://cssday.nl/2023>)
- Moderncss (<https://moderncss.dev/>)