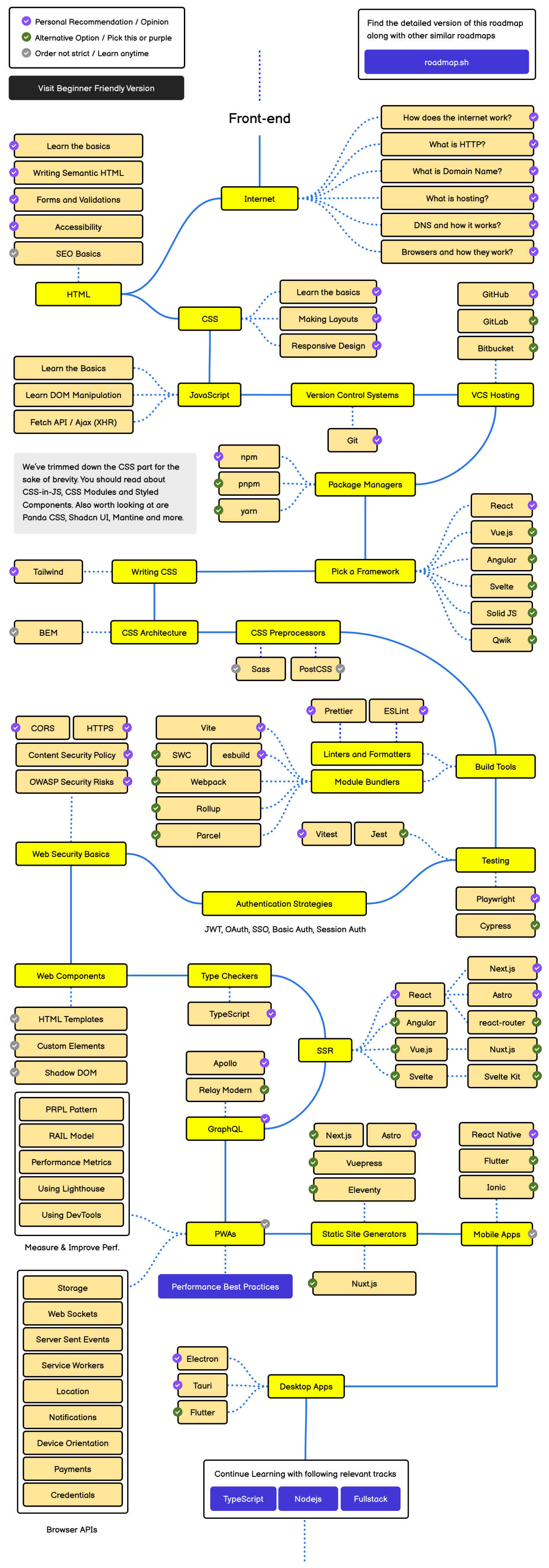
KAZAKH BRITISH TECHNICAL UNIVERSITY

WEB DEVELOPMENT

Lesson 2

Web development roadmap

https://roadmap.sh/



We've trimmed down the CSS part for the sake of brevity. You should read about CSS-in-JS, CSS Modules and Styled Components. Also worth looking at are Panda CSS, Shadcn UI, Mantine and more.

http://info.cern.ch/

World Wide Web — Tim Berners-Lee (1989)

<**H**yper**t**ext **M**arkup **L**anguage>

https://www.w3schools.com/html/

HTML Elements

- Headings
- Paragraphs
- Links
- Images
- Buttons
- Lists
- Tables
- …

https://www.w3schools.com/html/

Element attributes

- id
- class
- style
- data-
- title

https://www.w3schools.com/html/

HTML Comments

```html
<!-- This is a comment -->

<p>This is a paragraph.</p>

<!-- Remember to add more information here -->
```

https://www.w3schools.com/html/

HTML Forms

- Text input
- Radio button
- Checkbox
- Submit
- Select option
- Textarea
- Button
- Label

HTML Form Input Types

- Text
- Password
- Submit
- Reset
- Radio
- Checkbox
- Button

HTML Form Input Types

- Text
- Password
- Submit
- Reset
- Radio
- Checkbox
- Button
- File

HTML5 Input Types

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

HTML Form Attributes

- value
- readonly
- disabled
- size
- autocomplete
- autofocus
- min and max
- multiple
- pattern (regexp)
- placeholder
- required

HTML5

- Semantic HTML — header, footer, article, section, nav
- Form attributes — date, number, range
- Graphical elements — svg, canvas
- Multimedia elements — audio, video

https://html5book.ru/html-html5/

Style guide: https://www.w3schools.com/html/html5_syntax.asp

Cascading **S**tyle **S**heets

https://html5book.ru/css-css3

CSS can be added:

1. Inline
2. Internal
3. External

https://html5book.ru/css-css3

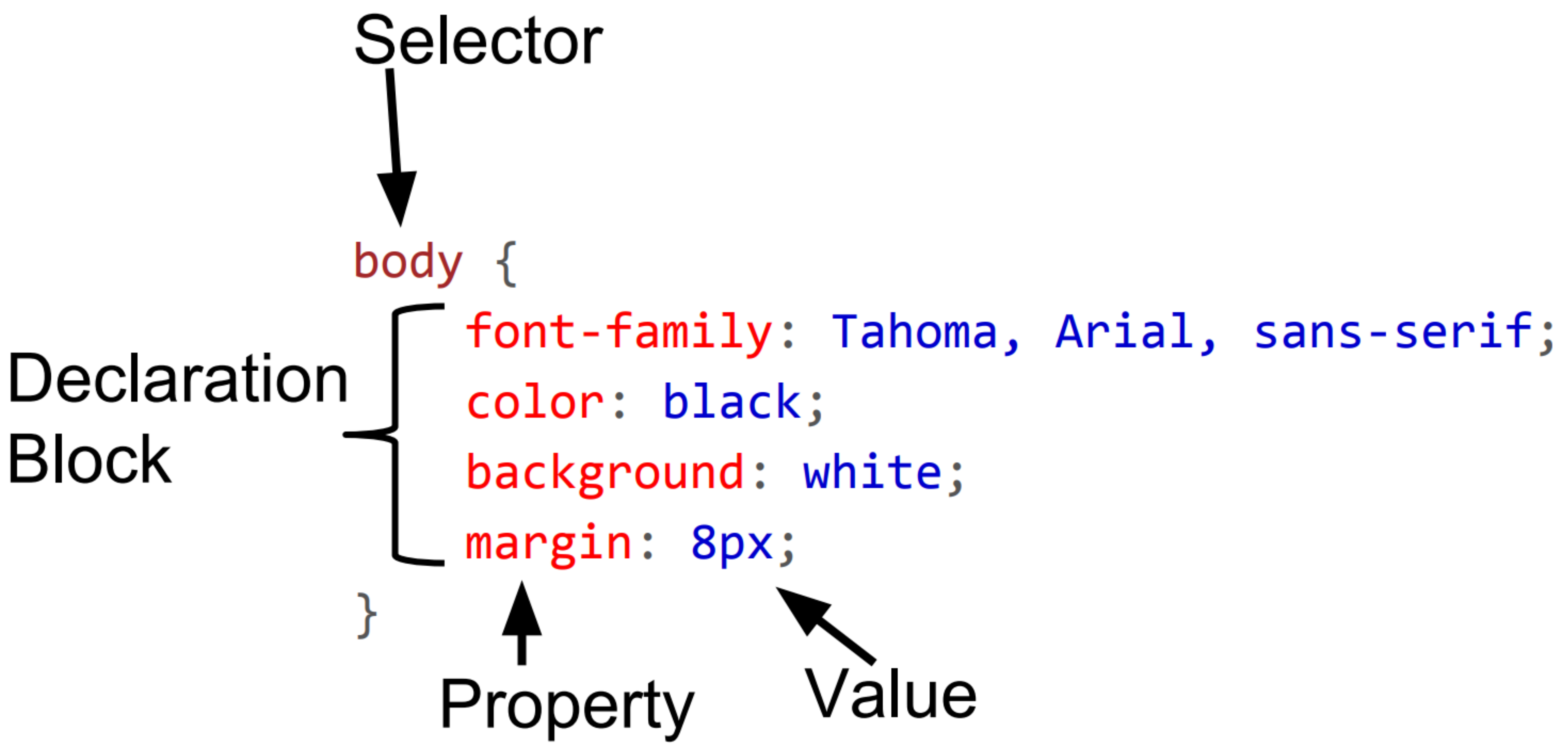
Common CSS Properties

- font
- border
- padding
- margin
- with *id / class*

https://html5book.ru/css-css3

DRY principle: Don't Repeat Yourself

Style sheet contain one or more CSS Rules



```css
body {
  font-family: Tahoma, Arial, sans-serif;
  color: black;
  background: white;
  margin: 8px;
}
```

CSS Selector	CSS	HTML

Tag name	`h1 {`
	`  color: red;`
	`}`
		`<h1>Today’s Specials</h1>`

Class attribute	`.large {`
	`  font-size: 16pt;`
	`}`
		`<p class="large">...`

Tag and Class	`p.large {...}`	`<p class="large">...`

Element id	`#p20 {`
	`  font-weight: bold;`
	`}`
		`<p id="p20">...`

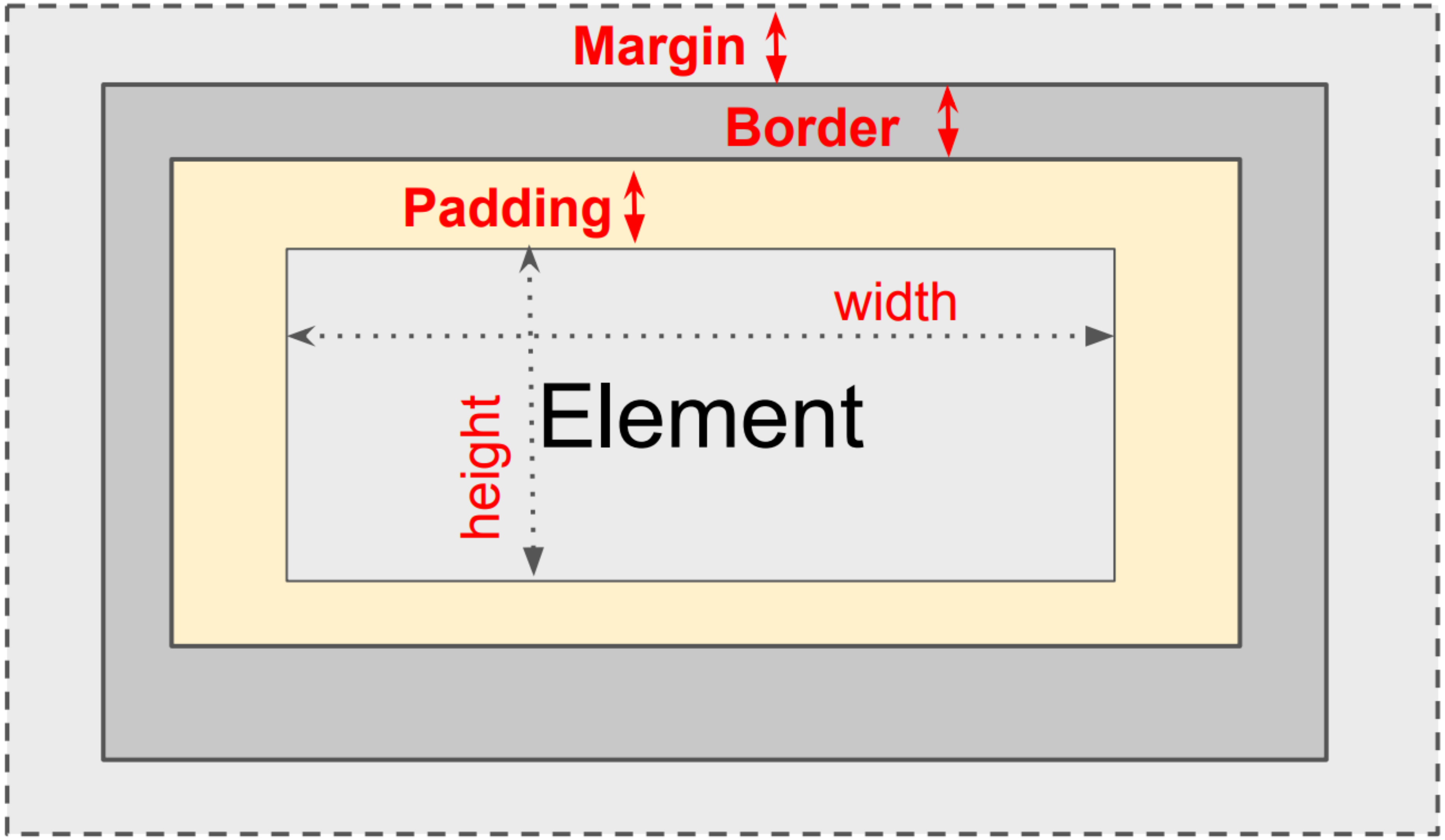
CSS Pseudo Selectors

```
p:hover, a:hover {
  background-color: yellow;
}

a:link {
  color: blue;
}

a:visited {
  color: green;
}
```

Colors

- Predefined names: `red, blue, green, white, etc.`
- 8-bit hexadecimal numbers for red, green, blue: $\texttt{\#}\underbrace{\texttt{ff}}_{R}\underbrace{\texttt{00}}_{G}\underbrace{\texttt{00}}_{B} \longrightarrow$ 🟥
- 0-255 decimal intensities: $\texttt{rgb(}\underbrace{\texttt{255}}_{R}\texttt{,}\underbrace{\texttt{255}}_{G}\texttt{,}\underbrace{\texttt{0}}_{B}\texttt{)} \longrightarrow$ 🟨
- Percentage intensities: $\texttt{rgb(}\underbrace{\texttt{80\%}}_{R}\texttt{,}\underbrace{\texttt{80\%}}_{G}\texttt{,}\underbrace{\texttt{100\%}}_{B}\texttt{)} \longrightarrow$ 🟪

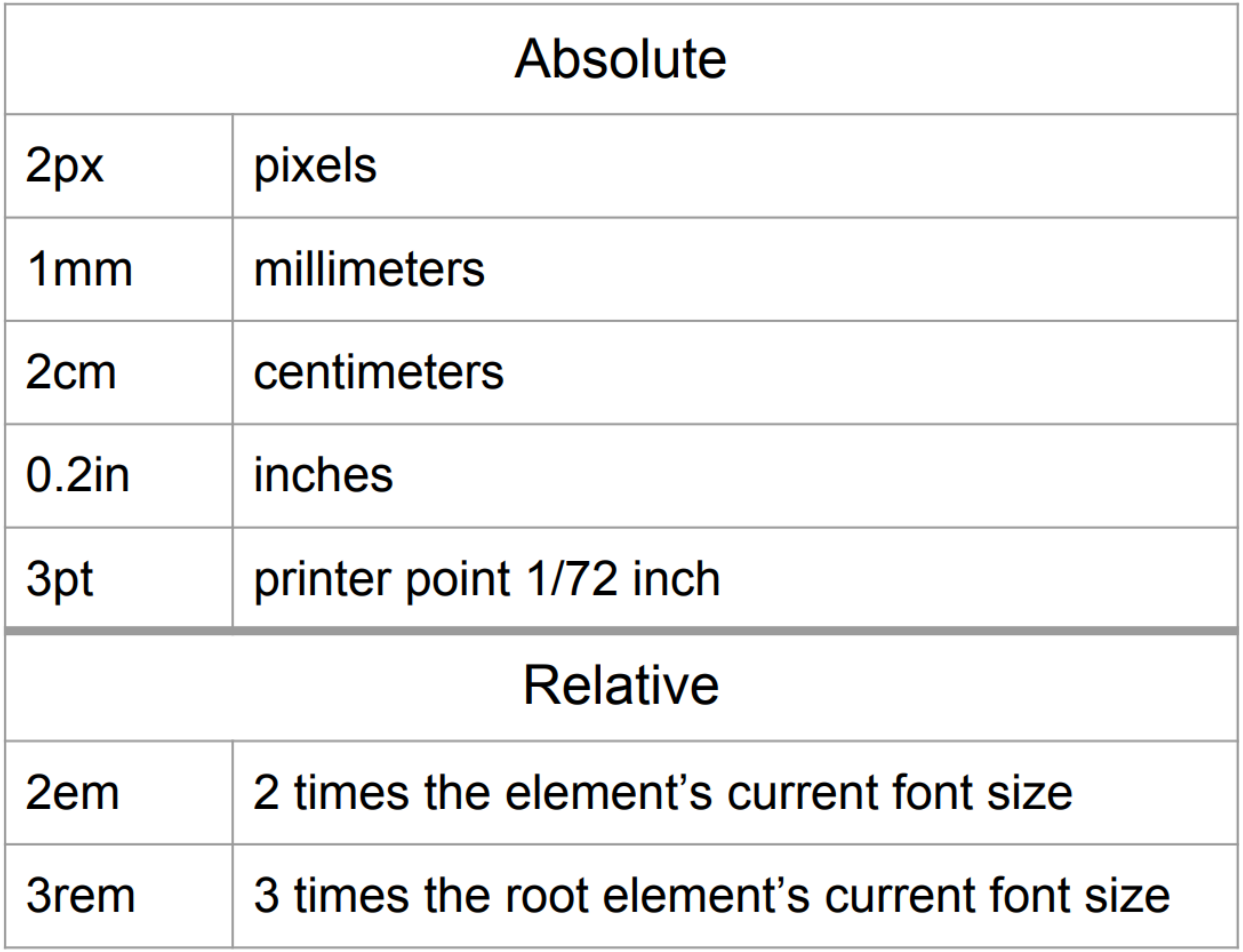
CSS Box Model



Total element width =
width +
left padding +
right padding +
left border +
right border +
left margin +
right margin

Margin & Padding
Transparent

CSS distance units



Absolute	
2px	pixels
1mm	millimeters
2cm	centimeters
0.2in	inches
3pt	printer point 1/72 inch
Relative	
2em	2 times the element’s current font size
3rem	3 times the root element’s current font size

Size Properties - Element, pad, margin, border

width - Override element defaults
height

padding-top
padding-right
padding-bottom
padding-left

margin-top
margin-right
margin-bottom
margin-left

```
border-bottom-color
border-bottom-style
border-bottom-width
border-left-color
border-left-style
border-left-width
border-right-color
border-right-style
border-right-width
```

etc.

```
p {
  border: 5px solid red;
}
```

`position` property

`position: static;`	(default) - Position in document flow
`position: relative;`	Position relative to default position via `top`, `right`, `bottom`, and `left` properties
`position: fixed;`	Position to a fixed location on the screen via `top`, `right`, `bottom`, and `left` properties
`position: absolute;`	Position relative to ancestor `absolute` element via `top`, `right`, `bottom`, and `left` properties

Fixed position (0,0) is top left corner

Element visibility control properties

`display: none;` - Element is not displayed and takes no space in layout
`display: inline;` - Element is treated as an inline element.
`display: block;` - Element is treated as an block element.
`display: flex;` - Element is treated as an flex container.
`display: grid;` - Element is treated as an grid container.

`visibility: hidden;` - Element is hidden but space still allocated.
`visibility: visible;` - Element is normally displayed

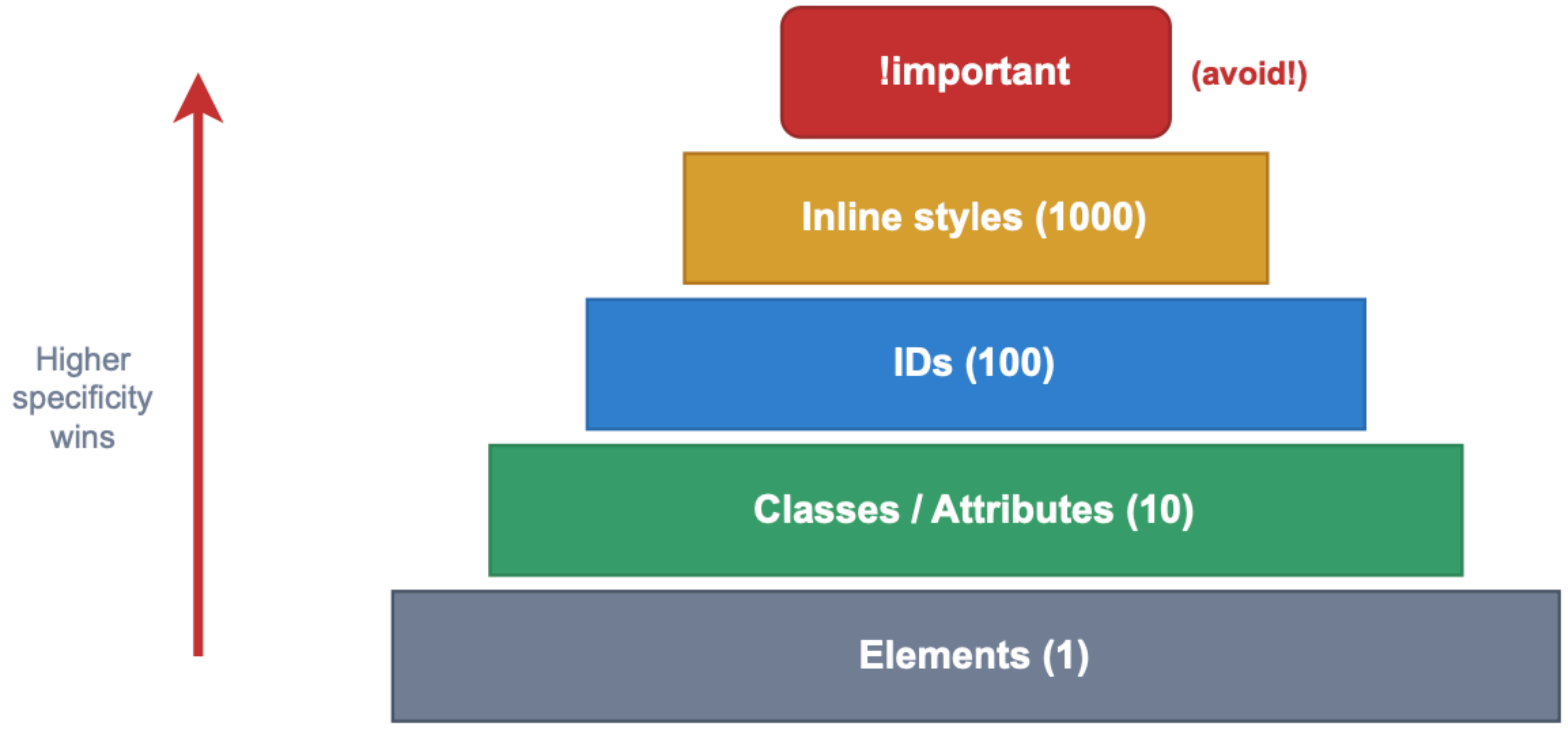
CSS Specificity

How does the browser decide which style wins?

Selector Type	Example	Score
Inline style	`style="color: red"`	1000
ID	`#header { }`	100
Class, attribute, pseudo-class	`.nav { }`, `[type="text"]`, `:hover`	10
Element, pseudo-element	`p { }`, `::before`	1
Combined	`div.card { }`	1 + 10 = 11

CSS Specificity

How does the browser decide which style wins?



CSS Flexbox

One-dimensional layout system for arranging items in rows or columns

Container Properties (parent)

Property	Values	Description
`display`	`flex`	Enables flexbox
`flex-direction`	`row`, `row-reverse`, `column`, `column-reverse`	Main axis direction
`justify-content`	`flex-start`, `flex-end`, `center`, `space-between`, `space-around`, `space-evenly`	Align on main axis
`align-items`	`flex-start`, `flex-end`, `center`, `stretch`, `baseline`	Align on cross axis
`flex-wrap`	`nowrap`, `wrap`, `wrap-reverse`	Allow wrapping
`gap`	`10px`, `1rem`	Space between items

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

CSS Flexbox

One-dimensional layout system for arranging items in rows or columns

Item Properties (children)

Property	**Example**	**Description**
`flex-grow`	`1`	How much item grows
`flex-shrink`	`0`	How much item shrinks
`flex-basis`	`200px`	Initial size
`flex`	`1 0 auto`	Shorthand (grow shrink basis)
`align-self`	`center`	Override container alignment

CSS Grid

Two-dimensional layout system for rows AND columns simultaneously

Container Properties (parent)

Property	**Values**	**Description**
`display`	`grid`	Enables grid
`grid-template-columns`	`repeat(3, 1fr)`	Define columns
`grid-template-rows`	`100px auto 100px`	Define rows
`gap`	`10px 20px`	Row and column gaps
`justify-items`	`start`, `end`, `center`, `stretch`	Align items horizontally
`align-items`	`start`, `end`, `center`, `stretch`	Align items vertically

https://css-tricks.com/css-grid-layout-guide/

Questions?