

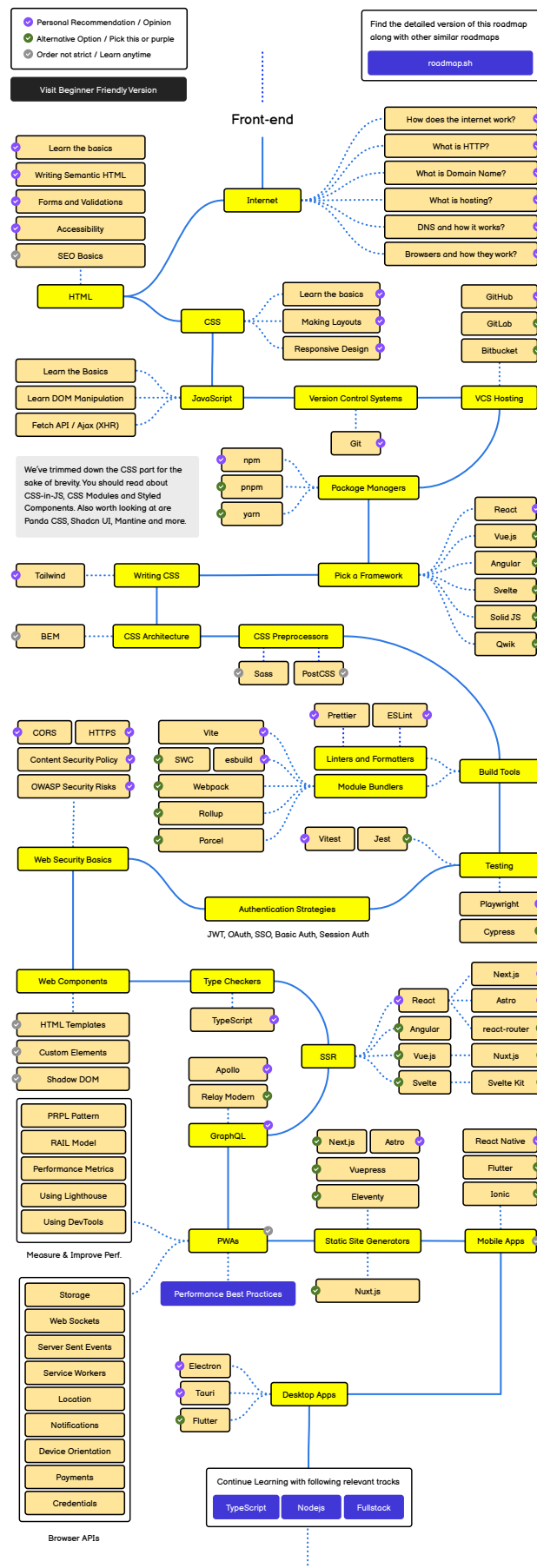


WEB DEVELOPMENT

Lesson 2

Web development roadmap

<https://roadmap.sh/>



<http://info.cern.ch/>

World Wide Web — Tim Berners-Lee (1989)

<**H**ypertext **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

```
<!-- 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>

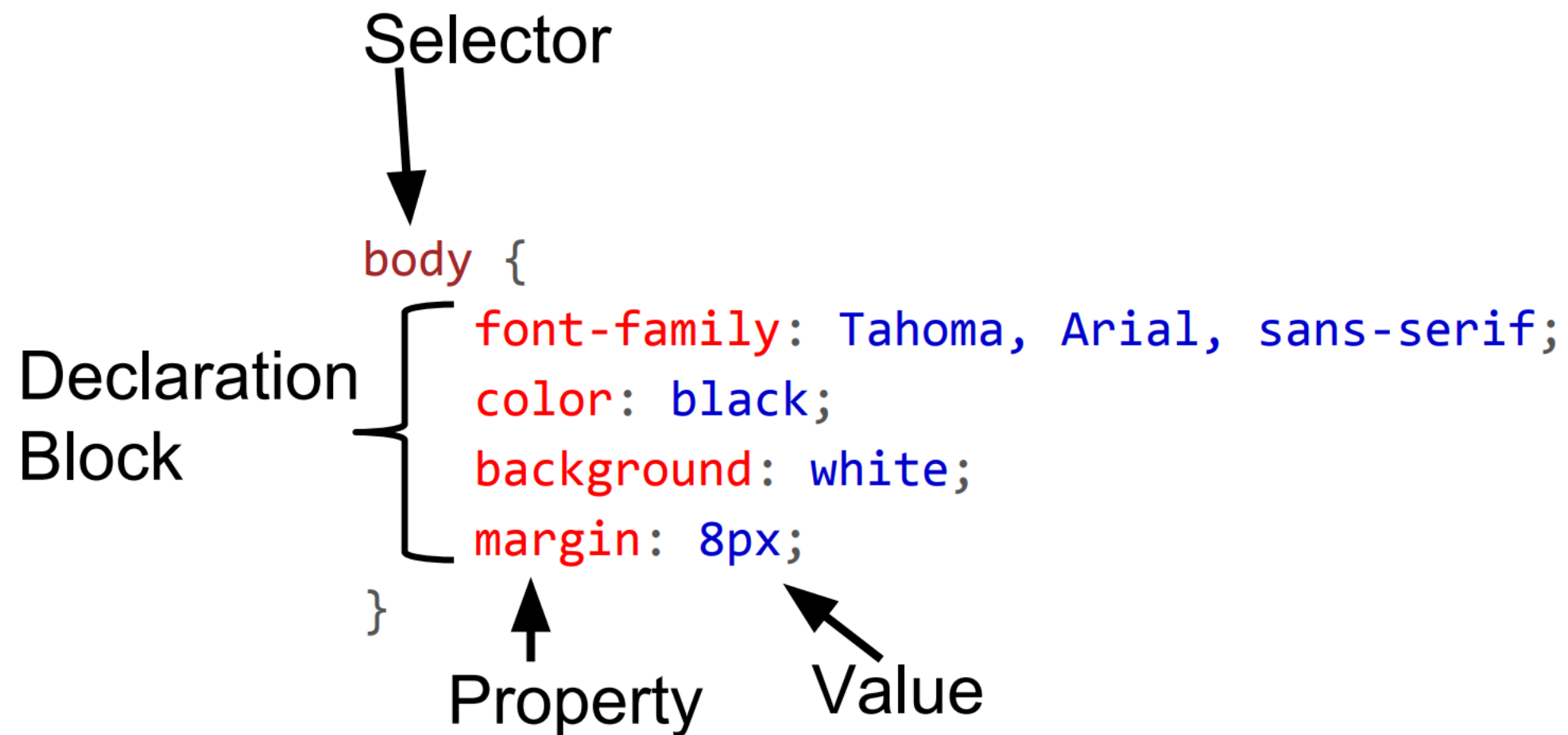
Common CSS Properties

- 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 Selector	CSS	HTML
Tag name	<pre>h1 { color: red; }</pre>	<pre><h1>Today's Specials</h1></pre>
Class attribute	<pre>.large { font-size: 16pt; }</pre>	<pre><p class="large">...</pre>
Tag and Class	<pre>p.large {...}</pre>	<pre><p class="large">...</pre>
Element id	<pre>#p20 { font-weight: bold; }</pre>	<pre><p id="p20">...</pre>

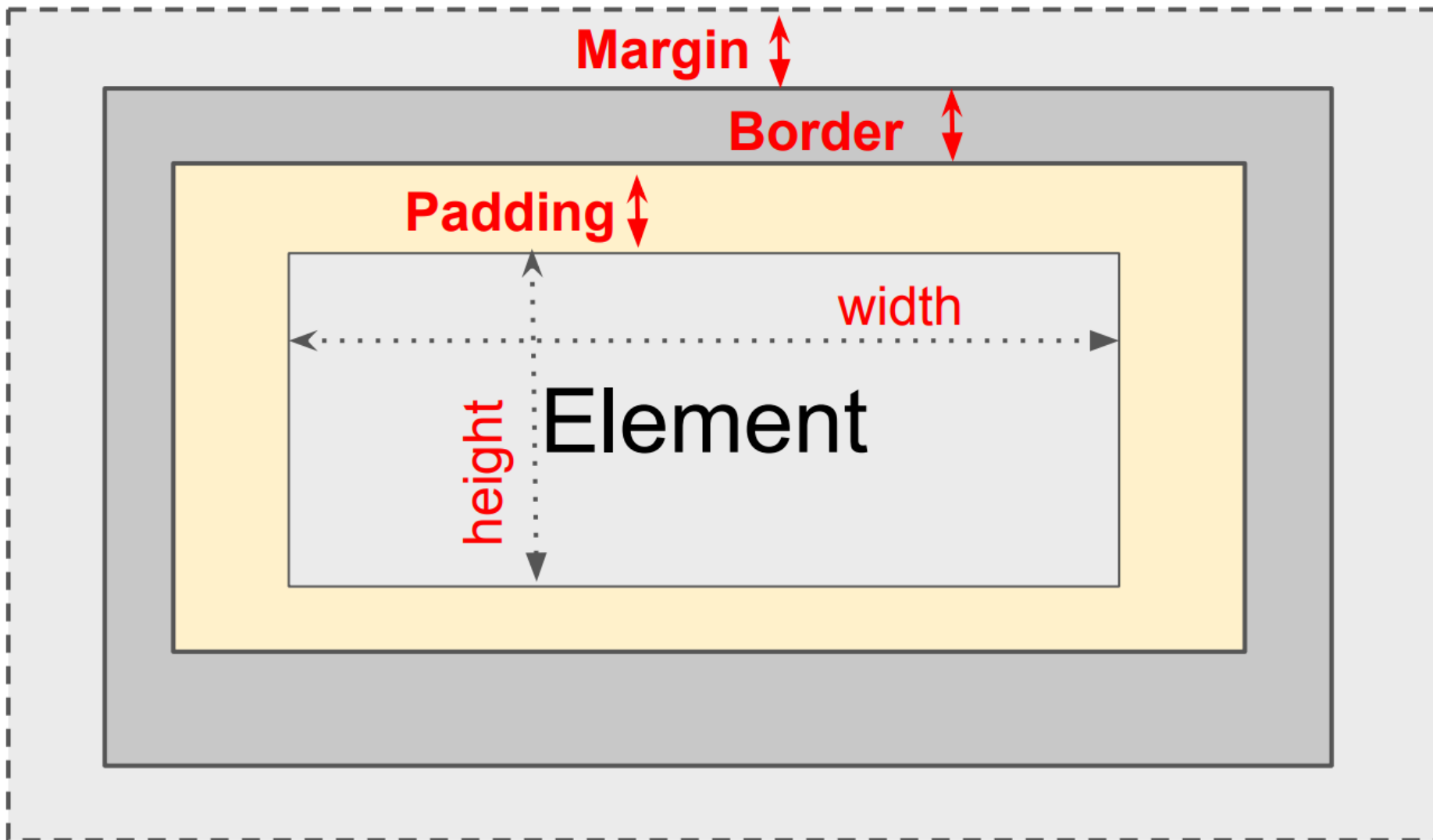
CSS Pseudo Selectors

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

```
a:link {  
    color: blue;  
}
```

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


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

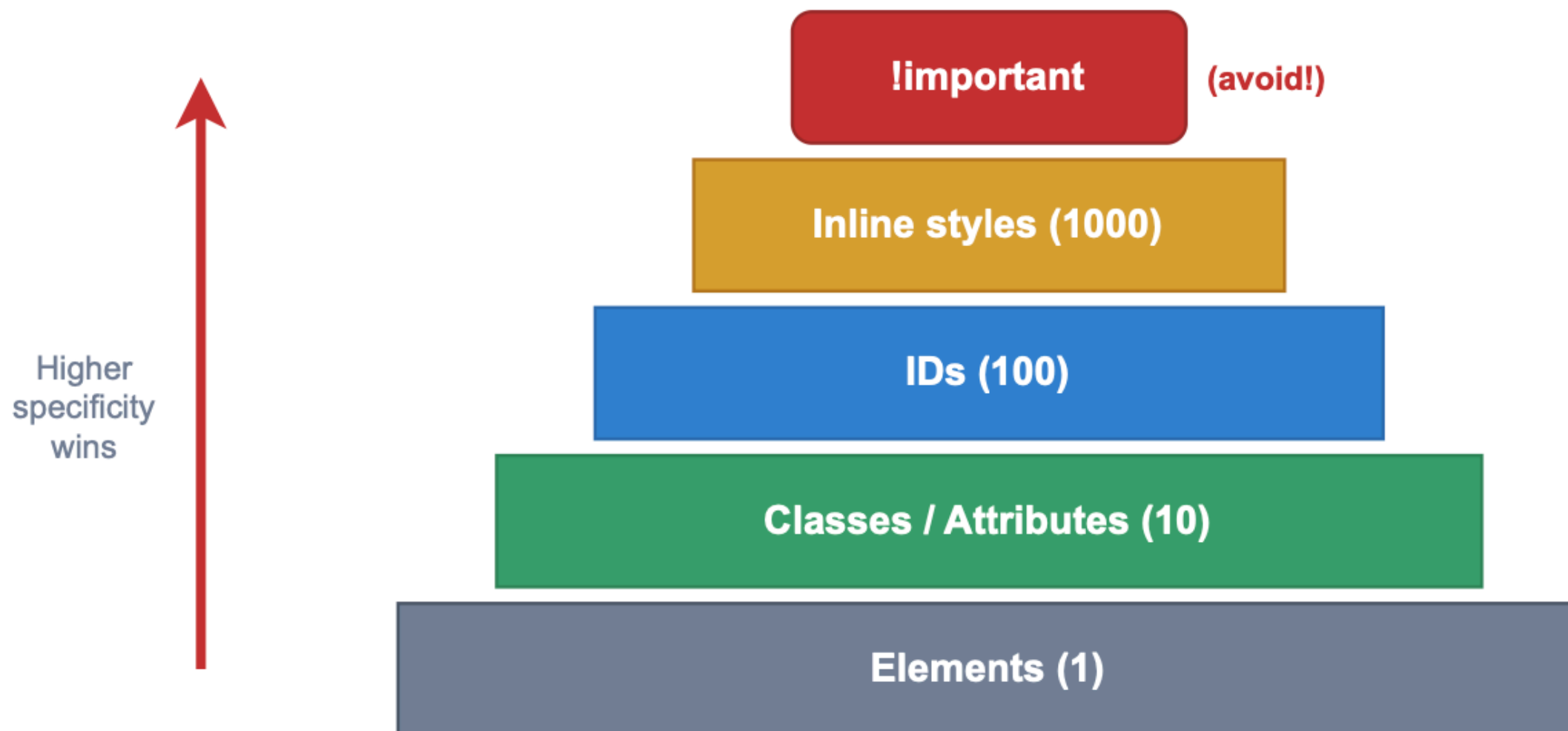
CSS Specificity

How does the browser decide which style wins?

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

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
<code>display</code>	<code>flex</code>	Enables flexbox
<code>flex-direction</code>	<code>row</code> , <code>row-reverse</code> , <code>column</code> , <code>column-reverse</code>	Main axis direction
<code>justify-content</code>	<code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>space-between</code> , <code>space-around</code> , <code>space-evenly</code>	Align on main axis
<code>align-items</code>	<code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>stretch</code> , <code>baseline</code>	Align on cross axis
<code>flex-wrap</code>	<code>nowrap</code> , <code>wrap</code> , <code>wrap-reverse</code>	Allow wrapping
<code>gap</code>	<code>10px</code> , <code>1rem</code>	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
<code>flex-grow</code>	1	How much item grows
<code>flex-shrink</code>	0	How much item shrinks
<code>flex-basis</code>	200px	Initial size
<code>flex</code>	1 0 auto	Shorthand (grow shrink basis)
<code>align-self</code>	center	Override container alignment

CSS Grid

Two-dimensional layout system for rows AND columns simultaneously

Container Properties (parent)

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

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

Questions?