CSS 3

Cascading Style Sheet

1.What is CSS?

CSS allows to create rules that specify how HTML elements should appear.

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

1.1. Applications of CSS

CSS saves time - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

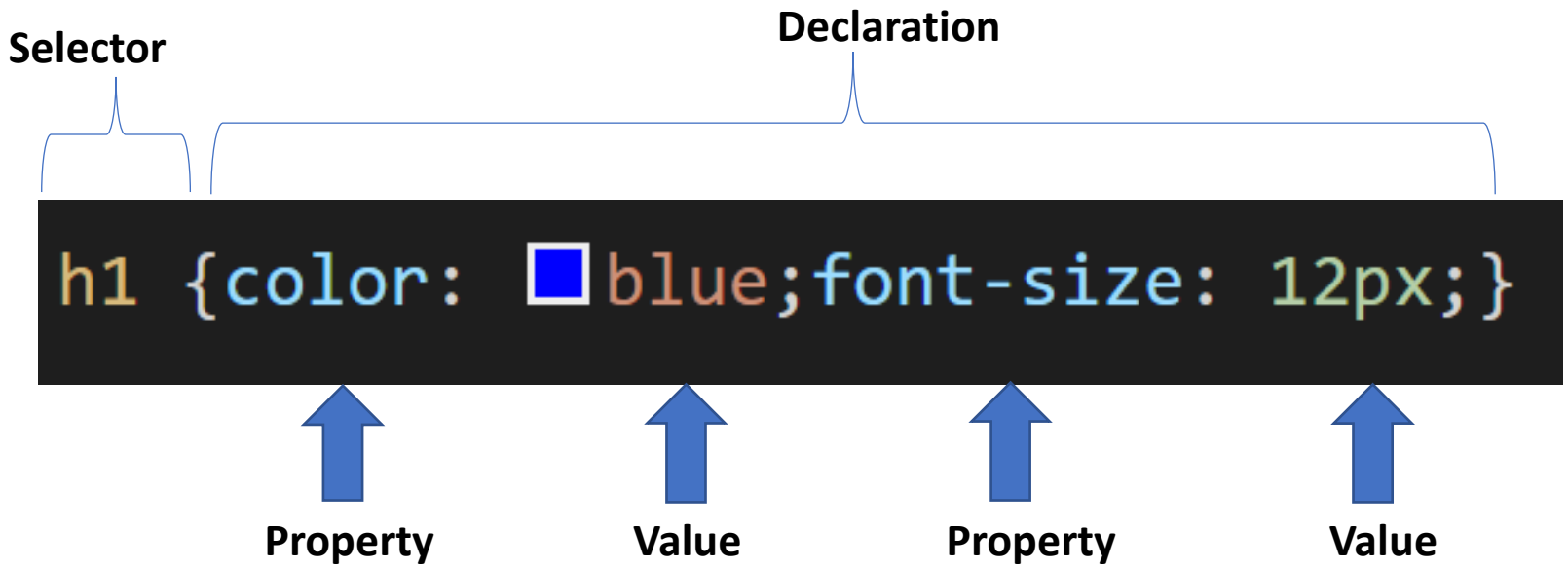
Pages load faster - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.

Easy maintenance - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

Superior styles to HTML - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as laptops/mobile/tablets.

1.2. CSS Syntax



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

1.3. Three Styling methods

1. Inline Styling method
   - An inline style may be used to apply a unique style for a single element.
   - To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

2. Internal Styling method
   - An internal style sheet may be used if one single HTML page has a unique style.
   - The internal style is defined inside the <style> element, inside the head section.

3. External Styling method
   - An external style sheet may be used if you want to use common styling across multiple web pages
   - Having an external style sheet, improves the code reusability and maintenance.
   - Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.
   - An external style must be saved with a .css extension.

1.3.1 Cascading Order Styling methods

Cascading Order:

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

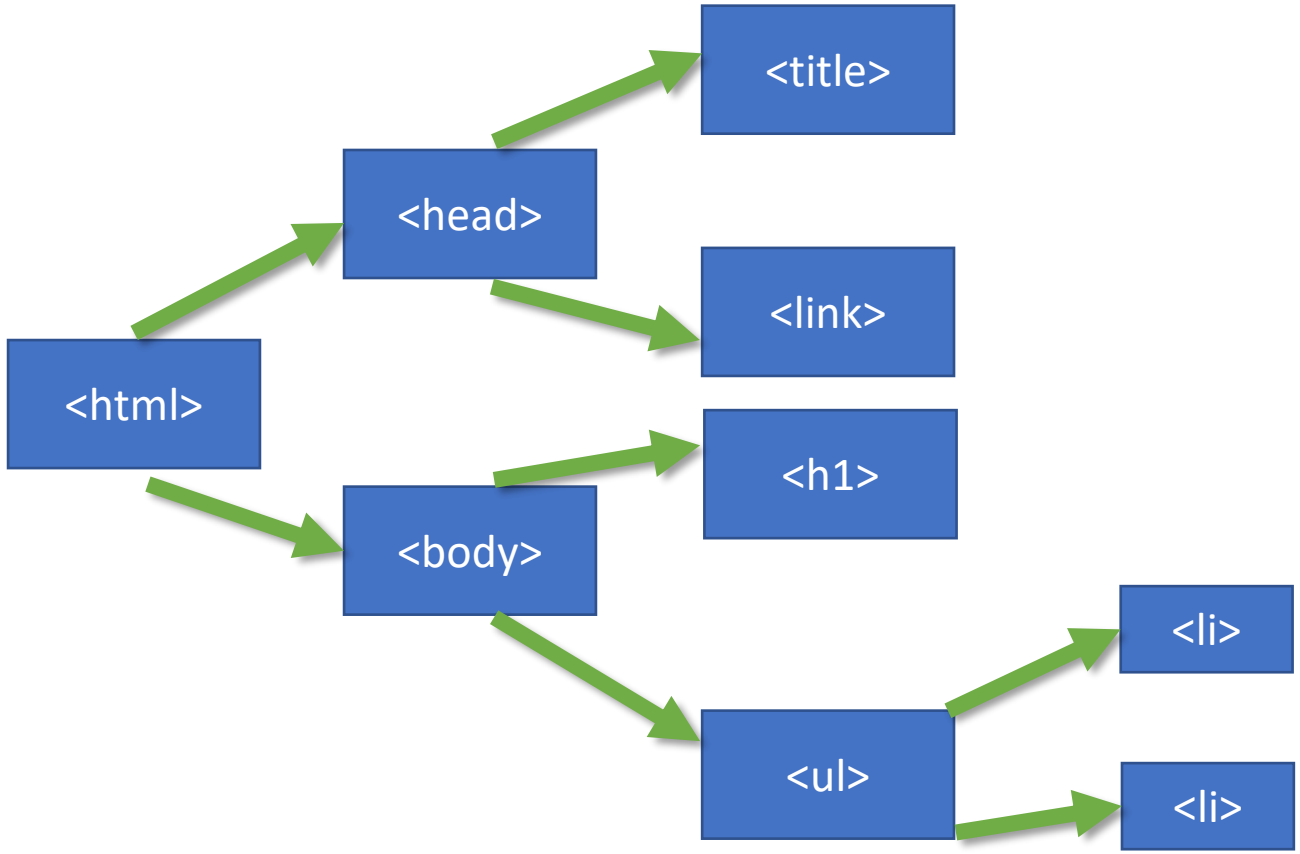
So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

NOTE:

A CSS comment is placed inside the <style> element, and starts with /* and ends with */

1.4. HTML Element Tree

HTML Element Tree



1.4.1 HTML Element Tree

Each HTML document can be referred to as a document tree. We describe the elements in the tree like we would describe a family tree.
There are ancestors, descendants, parents, children and siblings.
It is important to understand the document tree because CSS selectors use the document tree.

Ancestor

An ancestor refers to any element that is connected but further up the document tree - no matter how many levels higher.

Descendant

A descendant refers to any element that is connected but lower down the document tree - no matter how many levels lower.

Parent and Child

A parent is an element that is directly above and connected to an element in the document tree.
A child is an element that is directly below and connected to an element in the document tree.

Sibling

A sibling is an element that shares the same parent with another element.

1.4.1 HTML Element Tree

Each HTML document can be referred to as a document tree. We describe the elements in the tree like we would describe a family tree.
There are ancestors, descendants, parents, children and siblings.
It is important to understand the document tree because CSS selectors use the document tree.

Ancestor

An ancestor refers to any element that is connected but further up the document tree - no matter how many levels higher.

Descendant

A descendant refers to any element that is connected but lower down the document tree - no matter how many levels lower.

Parent and Child

A parent is an element that is directly above and connected to an element in the document tree.
A child is an element that is directly below and connected to an element in the document tree.

Sibling

A sibling is an element that shares the same parent with another element.

1.5 CSS Selectors

CSS selectors are used to select the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on element name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

1.5.1 Simple Selectors

The CSS element Selector

- The element selector selects HTML elements based on the element name.

```
h2{ background-color: tan; }
```

The CSS id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```
#emailId{border: 2px solid red;}
```

The CSS class Selector

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

```
.color-red{ background-color: red; }
```

1.5.2 Combinator Selectors

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Descendant Selector (space)
The descendant selector matches **all elements** that are descendants of a specified element.

Child Selector (>)
The child selector selects **all elements** that are the immediate children of a specified element.

1.5.2 Combinator Selectors

Adjacent sibling selector (+)

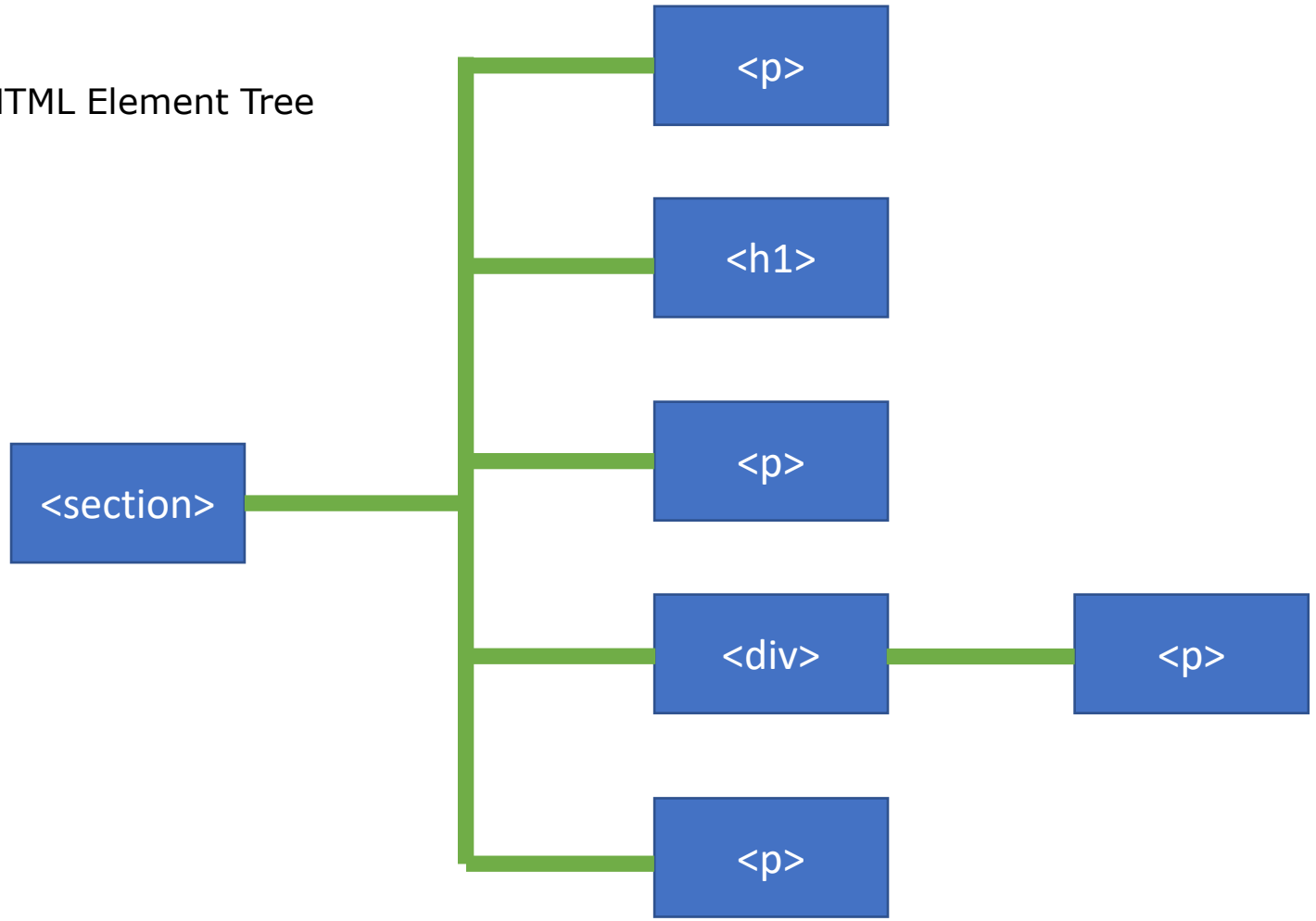
- Selects immediately **following** sibling of the specified element
- Sibling elements must have the same parent element, and "adjacent" means "immediately following".

General sibling selector (~)

- Selects all the sibling **after** the specified element

1.5.2 Element Tree

HTML Element Tree



1.5.3 Pseudo-class selectors

A pseudo-class is used to define a special state of an element.

Syntax:

```
selector:pseudo-class {
    property: value;
}
```

1.5.3.1 Anchor pseudo elements

Link Selector:
The :link selector is used to select unvisited links.

Visited Selector:
The :visited selector to style links to visited pages.

Hover Selector:
The :hover selector to style links when you mouse over them.

Active Selector:
The :active selector to style links when you click on them.

Focus Selector:
The :focus selector to style links when you focus on them.

```
/*  Unvisited link */
a:link{
    color: yellow;
}
/* visited link */
a:visited {
color: green;
}

/* mouse over link */
a:hover {
color: hotpink;
}
/* Focus link */
a:focus {
color: red;
}

/* selected link */
a:active {
color: blue;
}
```

1.5.3.2 Other pseudo elements

Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every element that has no children (including text nodes)
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every element that is the first element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every element that is the last element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a particular element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects <input> elements with no "required" attribute
:out-of-range	input:out-of-range	Selects <input> elements with a value outside a specified range
:read-only	input:read-only	Selects <input> elements with a "readonly" attribute specified
:read-write	input:read-write	Selects <input> elements with no "readonly" attribute
:required	input:required	Selects <input> elements with a "required" attribute specified
:root	root	Selects the document's root element
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all <input> elements with a valid value
:visited	a:visited	Selects all visited links

1.5.4 Pseudo-Element selector

- A CSS pseudo-element is used to style specified parts of an element.

```
selector::pseudo-element{
    property: value;
}
```

Selector	Example	Example description
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
::first-letter	p::first-letter	Selects the first letter of each <p> element
::first-line	p::first-line	Selects the first line of each <p> element
::marker	::marker	Selects the markers of list items
::selection	p::selection	Selects the portion of an element that is selected by a user

1.5.5 Attribute Selectors

- The [attribute] selector is used to select elements with a specified attribute.

```
element[attribute]{
    property:value;
}
```

Selector	Example	Example description
[*attribute*]	[target]	Selects all elements with a target attribute
[*attribute=value*]	[target=_blank]	Selects all elements with target="_blank"
[*attribute~=value*]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[*attribute|=value*]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[*attribute^=value*]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[*attribute$=value*]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[*attribute*=value*]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"

1.6 CSS Colors

Colors are specified using

- predefined color names,
- RGB,
- RGBA,
- HEX,
- HSL,
- HSLA.

1.6.1 Color Names:

In CSS, a color can be specified by using a predefined color name.

```
h2{ background-color: tan; }
```

1.6.2 RGB Value

An RGB color value represents RED, GREEN, and BLUE light values.
Range of values between 0 – 255(0 being lowest, 255 being highest)

Syntax:
rgb(*red, green, blue*)

```
h2{background-color: rgb(255, 0, 0);}
```

rgb(red, green, blue);

rgb(255, 0, 0); - red;
rgb(0, 255, 0); - green;
rgb(0, 0, 255); - blue;

1.6.3 RGBA Value

An RGBA color value represents RED, GREEN,BLUE,ALPHA values.
Range of RGB values between 0 – 255(0 being lowest, 255 being highest)
Range of Alpha values between 0 – 1 (0 being full transparent, 1 being zero transparent)

Syntax:
rgba(*red, green, blue, alpha*)

```
h2{background-color: rgba(255, 0, 0, 0.5);}
```

1.6.4 HEX Value

A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.
Range is 0 to FF(same as 0-255)

Syntax:
#rrggbb

3 Digit HEX Value
The 3-digit hex code is a shorthand
for some 6-digit hex codes.
The 3-digit hex code has the following form:
#rgb

```
h2{background-color: #1d0505;;}
```

#ff0000 - red; (#f00)
#00ff00 - green; (#0f0)
#0000ff - blue; (#00f)

#ffffff - white (#fff)
#000000 - black (#000)

1.6.5 HSL Values

In CSS, a color can be specified using hue, saturation, and lightness (HSL) .
Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.
Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.
Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

Syntax:
hsl(*hue, saturation, lightness*)

```
h2{background-color: hsl(255, 0%, 0%);}
```

1.6.6 HSLA Values

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.
The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all)

Syntax:

hsla(*hue, saturation, lightness, alpha*)

```
h2{background-color: hsla(255, 0%, 0%,0);}
```

1.6.7 CSS Colors

1.6.7.1 Background colors
1.6.7.2 Text colors
1.6.7.3 Border colors

```
h2{background-color: hsla(255, 0%, 0%,0);}
h2{color: red;}
h2{border: 2px solid red;}
```

Background color

Font color

Border color

1.7 CSS Specificity

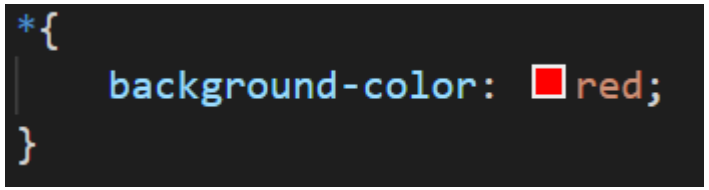
is the means by which browsers decide which CSS property values are the most relevant to an element and, therefore, will be applied.

1.7.1 Specificity scoring

Each selector rule gets a scoring. You can think of specificity as a total score and each selector type earns points towards that score. The selector with the highest score wins.

1.7.2.Universal selector

A universal selector (*) has no specificity and gets **0** points. This means that any rule with 1 or more points will override it



1.7 CSS Specificity

1.7.3 Element or pseudo-element selector

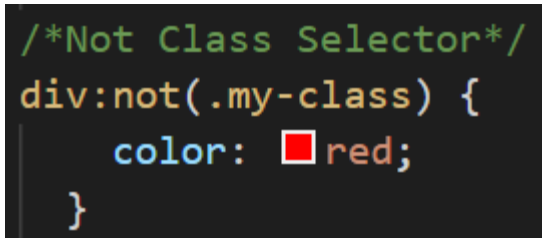
An element (type) or pseudo-element selector gets **1** point of specificity .

```
/*Type Selector*/
div{
    color: red;
}
/*Pseudo Element Selector*/
::selection{
    background-color: blueviolet;
}
```

1.7 CSS Specificity

1.7.5 Not Selector

The :not() pseudo-class itself adds nothing to the specificity calculation. However, the selectors passed in as arguments do get added to the specificity calculation.



Example: The above sample would have **11** points of specificity because it has

- one type selector (div) and
- one class inside the :not()

1.7 CSS Specificity

1.7.6 ID Selector

An ID selector gets **100** points of specificity.

```
/*ID Selector*/
#myId{
    background-color: brown;
}
```

1.7 CSS Specificity

1.7.7 Inline style attribute

CSS applied directly to the style attribute of the HTML element, gets a specificity score of **1,000** points.

```
<!--Inline Styling-->
<h2 style="background-color: aqua;">Inline Styling</h2>
```

1.7.8 !important attribute

- !important at the end of a CSS value gets a specificity score of **10,000** points.
- This is the highest specificity that one individual item can get.

```
/*Important Selector*/
.my-class-important {
    color: red !important;
  }
```

1.7.9 CSS Specificity - Quiz

What is the specificity score of `a[href="#"]` ?

- ○ 1
- ○ 5
- ○ 11

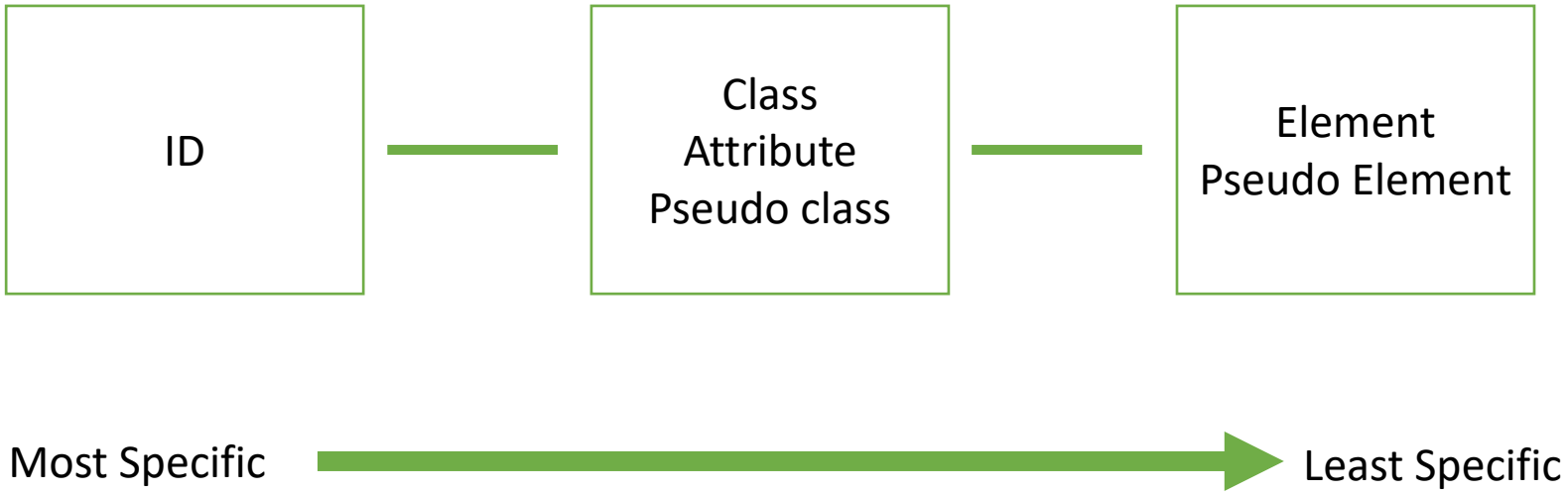
1.7.10 CSS Specificity - Quiz

Write a class with highest number of specificity score for the given element

```
<a class="my-class another-class" href="#">A link</a>
```

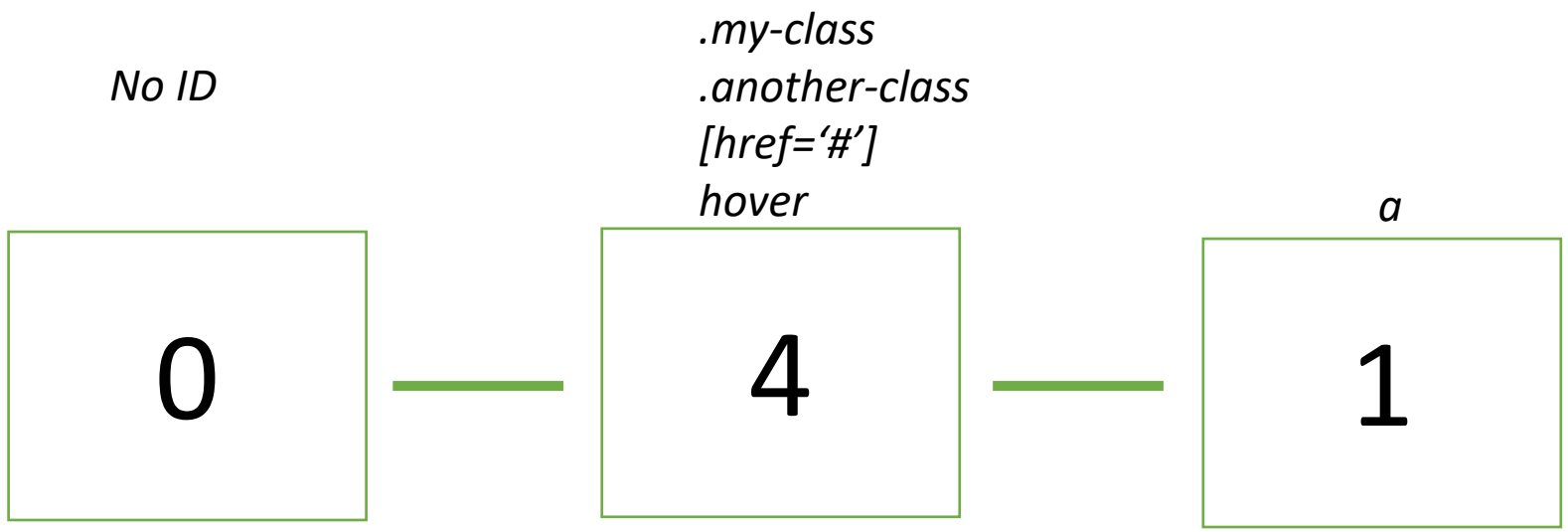
1.8 Visualizing specificity

In diagrams and specificity calculators, the specificity is often visualized like this:



1.8 Visualizing specificity

In diagrams and specificity calculators, the specificity is often visualized like this:




```
a.my-class.another-class[href='#']:hover{
    background-color: aqua;
}
```

1.8.1 Matching specificity

If multiple rules have identical specificity, the newest one is selected by CSS to have its rules applied.

1.8.2 Visualizing Specificity Quiz

Write the specificity for the below CSS classes.

section#specialty.dark

#specialty:hover li.dark

[data-state-rad].dark#specialty:hover

li#specialty section.dark

1.9 CSS Borders

The CSS border properties allow you to specify the style, width, and color of an element's border.

CSS Border Style

The border-style property specifies what kind of border to display.

CSS Border Width

The border-width property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick

```
.border-10px{
    border-width: 10px;
}
.border{
    border-width: 10px 20px 30px 40px;
}
.border-shorthand{
    border-width: 10px 20px;
}
```

→ Top , right , bottom , left all same

→ Top ,right ,bottom , left all different

→ Top & Bottom are same, Left & right are same

1.9 CSS Borders

CSS Border Color

The border-color property is used to set the color of the four borders.

CSS Rounded Borders

The border-radius property is used to add rounded borders to an element

1.10 CSS Margins

- The CSS margin properties are used to create space around elements, outside of any defined borders.

Property	Description
margin	A shorthand property for setting the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

1.10.1 CSS Margin Collapse

The CSS margin properties are used to create space around elements, outside of any defined borders.

```
margin: 20px; /*All sides are equal*/
margin: 10px 20px 30px 40px; /*Top Right Bottom Left are different*/
margin: 10px 20px; /*Top & Bottom are same , Left & Right are same*/
```

1.11 CSS Padding

- Padding is used to create space around an element's content, inside of any defined borders.

Property	Description
padding	A shorthand property for setting all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element

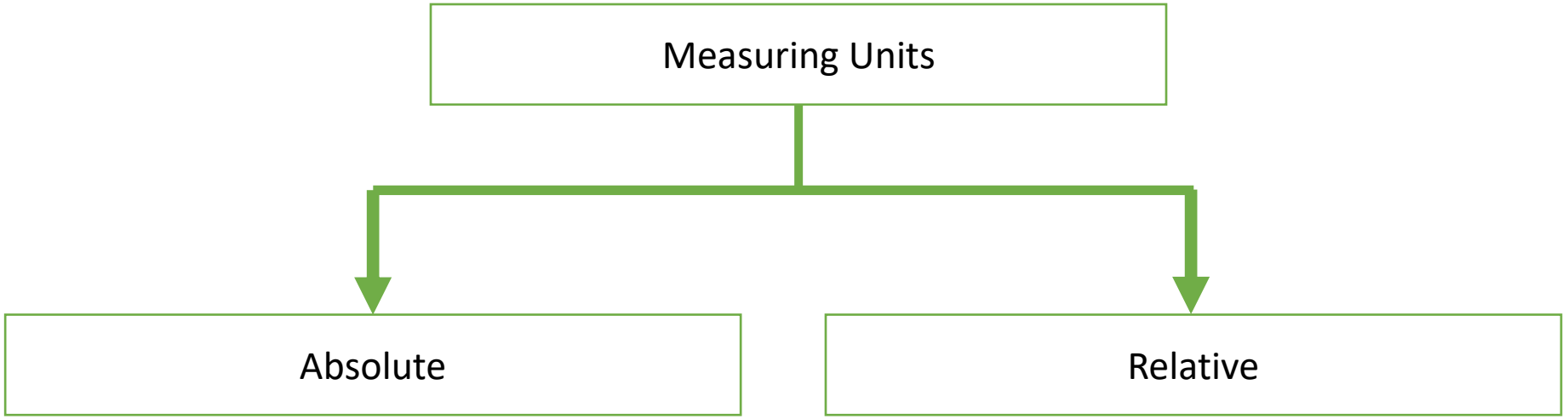
1.9 CSS Box Model

The CSS box model is essentially a box that wraps around every HTML element.

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

1.10 Measurement Units

CSS has several different units for expressing a length.



1.10.1 Absolute Units

- The absolute length units are fixed, and a length expressed in any of these will appear as exactly that size.
- Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

1.10.2 Relative Units

- Relative length units specify a length relative to another length property. Relative length units scale better between different rendering medium.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element