SWE 363: Web Engineering & Development

Module 3
Cascading Style Sheets

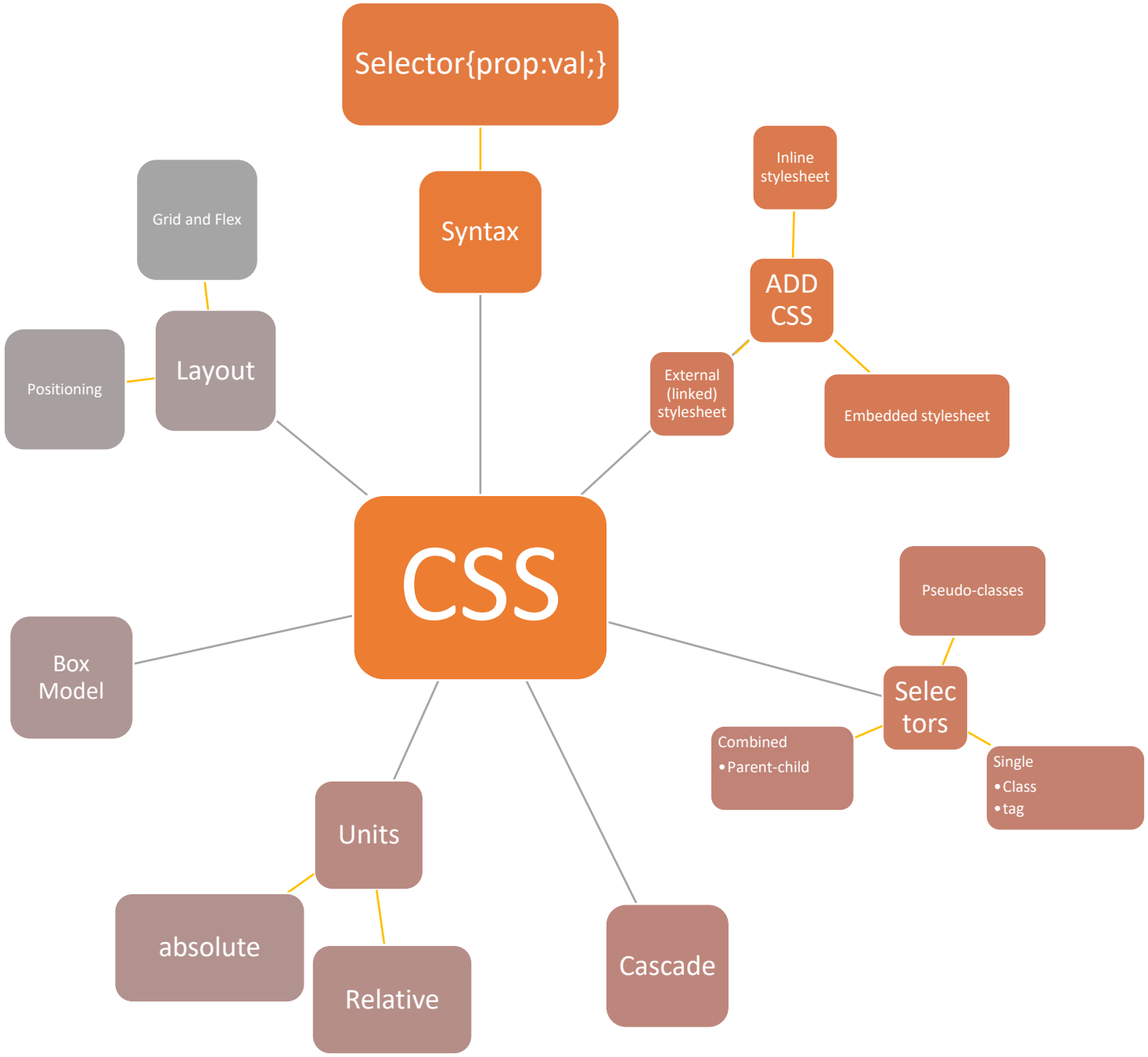
Objectives

Learn the basics of CSS

Learn the different CSS types

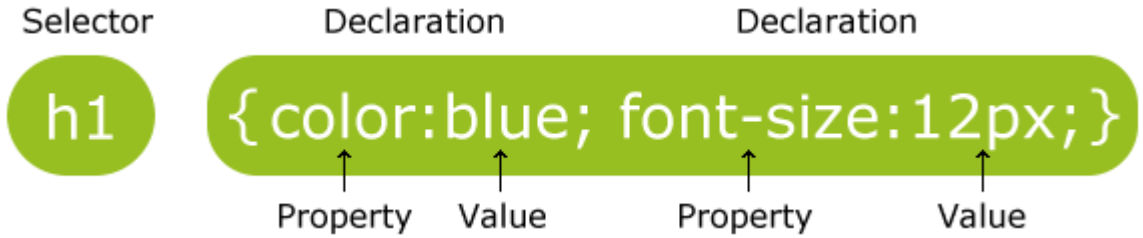
Use style sheets to separate presentation from content

Outline


Selector{prop:val;}
Syntax
Grid and Flex
Layout
Positioning
Inline stylesheet
ADD CSS
External (linked) stylesheet
Embedded stylesheet
CSS
Box Model
Pseudo-classes
Selec tors
Combined
• Parent-child
Single
• Class
• tag
Units
absolute
Relative
Cascade

CSS Syntax

CSS Rules: Selectors & Declarations



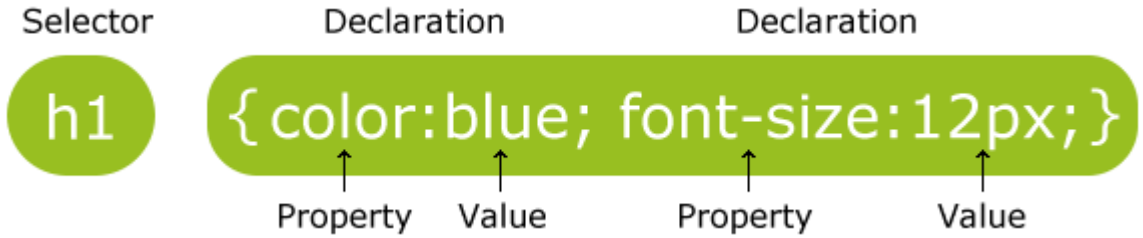
Rule

- A CSS Style Sheet is basically a collection of rules, each rule *contains 2 parts*:
  - *Selector (s)*
  - *Declaration (s)*, which describe how these elements should be displayed.
- Each rule starts with a **selector**
  - The *selector* identifies *which element or elements* in the HTML document will be affected by the declarations in the rule
  - selectors are *html tags* to be affected

```
selector, ..., selector {
   property: value;
   . . .
   property: value
}
```

CSS Syntax

CSS Rules: Properties & Values



- Declarations consist of 2 parts: a property and a value.
  - property and value describe the appearance of that tag;
  - Properties and values are separated with a colon (:)
  - a semicolon must be used between different declarations.
- The **unit** of any given value is dependent upon the property.
  - Some property values are from a predefined list of keywords.
  - Others are values such as length measurements, percentages, numbers without units, color values.
- If the value is multiple words, put **"quotes"** around the value

Adding CSS to HTML

Three ways

- Inline 

Code	output
`<p style="color: red;">This is a paragraph</p>`	This is a paragraph

- Embedded in the <head> 

Code	output

`<head><style>`
`h2 { background-color: yellow; }`
`</style></head>`
`<body><h2>My Title</h2>`
	My Title

- External: linked in the <head> 
  -

```
<head>
    <link rel="stylesheet" href="test.css" />
</head>
```

CSS Selectors

The *selector* identifies *which element or elements* in the HTML document will be affected by the declarations in the rule

Simple Selectors

Classification	Description	HTML	CSS Selector syntax
Type Selector	Selects all elements that have the given node name.	Ingredients: Banana Steps … Tips	h2
Class selector	Selects all elements that have the given *class* attribute	<h2>Ingredients:</h2> Banana <h2 class="st">Steps</h2> … <h2 class="st">Steps</h2>	**.st**
ID Selector	Selects an element based on the value of its *id* attribute. There should be only one element with a given ID in a document.	Ingredients: Banana Steps	**#in1**

Pseudo classes

```
a:link    { color: #FF0000; } /* unvisited link */
a:visited { color: #00FF00; } /* visited link */
a:hover   {  /* mouse over link */
  color: #FF00FF;
  cursor: pointer; /* can set new pointer icon, usually a "hand" */
}
```

Class	Description
:hover	An element that has the mouse over it
:visited	A link that has already been visited
:first-child	An element that is the first one to appear inside another
:nth-child(N)	Applies to every Nth child of a given parent

https://www.w3schools.com/cssref/sel_firstchild.asp
https://www.w3schools.com/cssref/sel_nth-child.asp

Combined Selectors

Combinator	Description	Example

Selector list “,”	selects all the matching nodes	section, p, h3 {
		color: green;
		font-size: 14pt;
		}

Descendant combinator “ “	The " " (space) combinator selects nodes that are descendants of the first element.	div span{} will match all elements that are inside a element.
Child combinator “>”	The > combinator selects nodes that are direct children of the first element.	ul > li{} will match all elements that are nested directly inside a element.

Combinator exercise

```
<section>
  <p>no pressure, no diamonds</p>
  <p>impossible is for the unwilling</p>
  <article>
 <p>try, try again</p>
 <p>take the risk or lose the chance</p>
 <div>
   <p>that's enough</p>
 </div>
  </article>
</section>
```

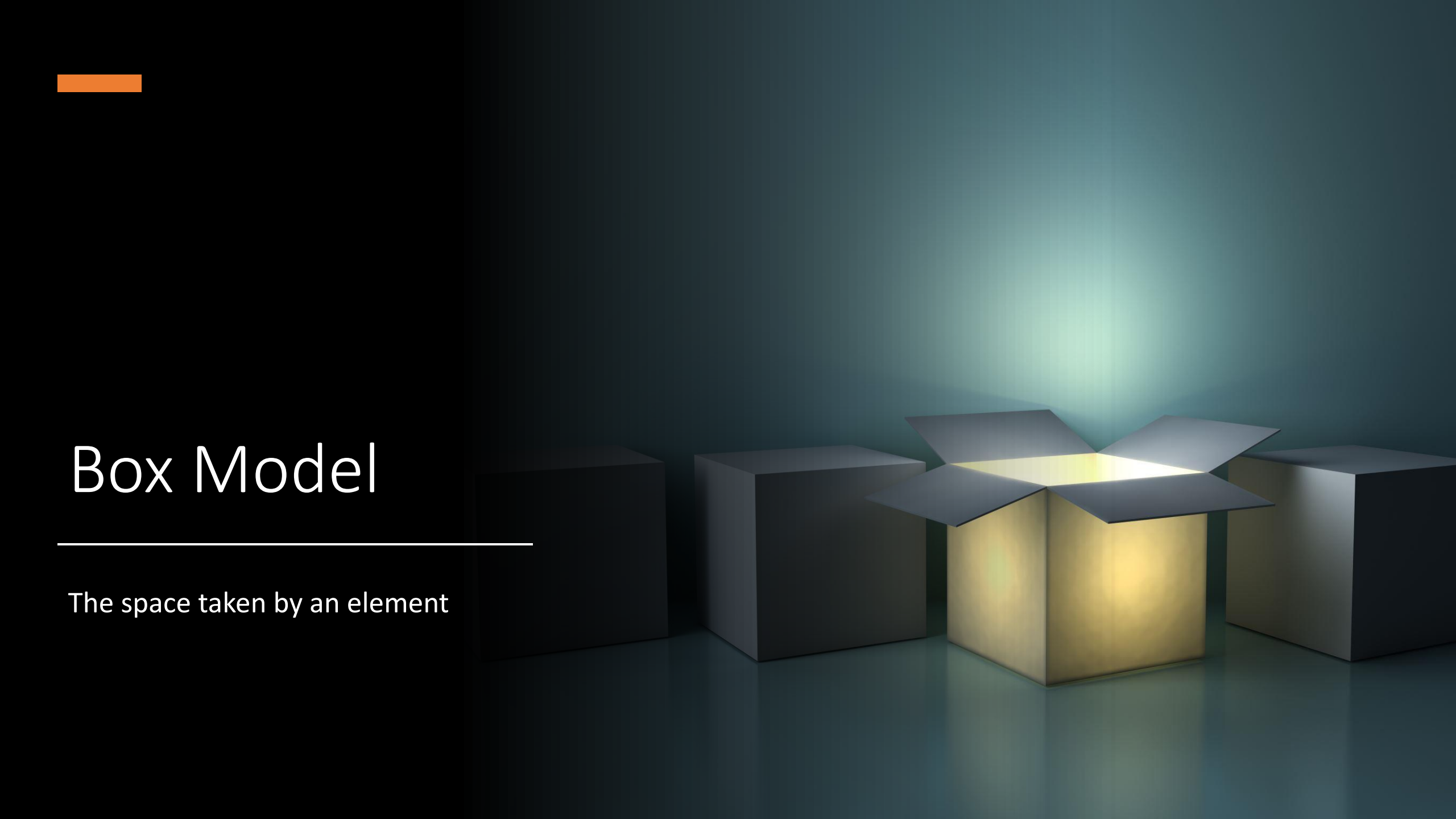
- What selector would select all the <p> tags?
- What selector would select the <p> tags highlighted in red?
- What selector would select the <p> tags highlighted in blue?
- What selector would select the <p> tag highlighted in pink?

https://www.w3schools.com/css/css_combinators.asp

A nice game for practicing selectors

Box Model

The space taken by an element

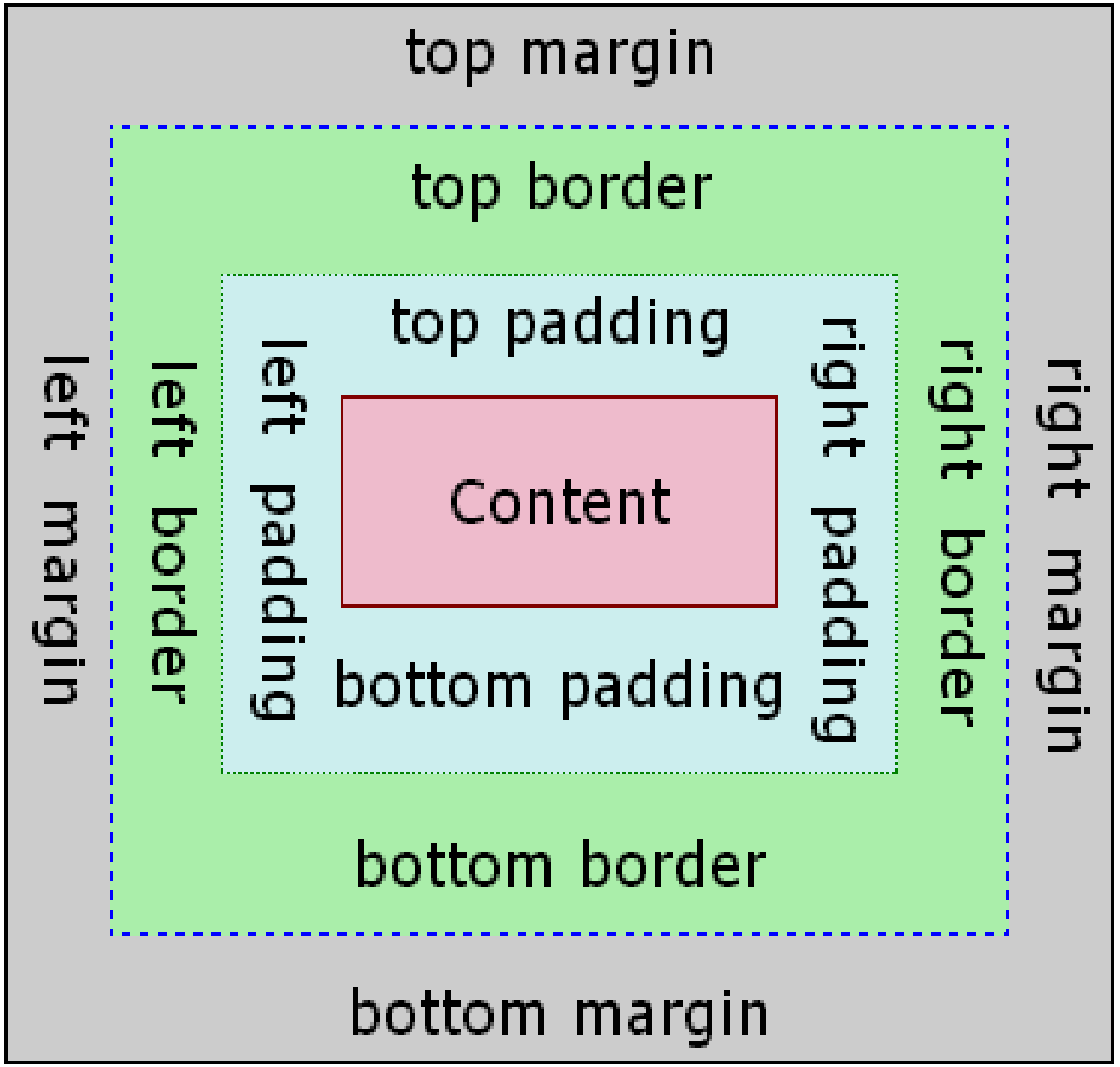


Box model properties

Margin: (*outside*) space between different elements

Border: (*optionally visible*) line that separates elements

Padding: (*inside*) space between element content and border

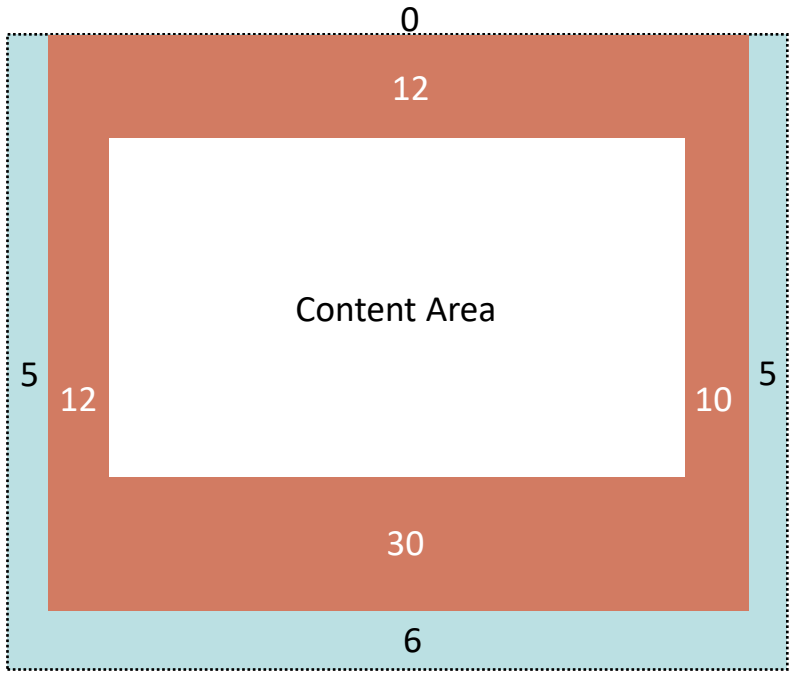


CSS Box Model

CSS Shorthand: Margin & Padding

- For margin and padding (and others), CSS provides a number of shorthand properties that can save on writing lines and lines of code.
  - Instead of writing this:

```
#container {
        margin-top: 0;
        margin-right: 5px;
        margin-bottom: 6px;
        margin-left: 5px;
        padding-top: 12px;
        padding-right: 10px;
        padding-bottom: 30px;
        padding-left: 12px;
}
```

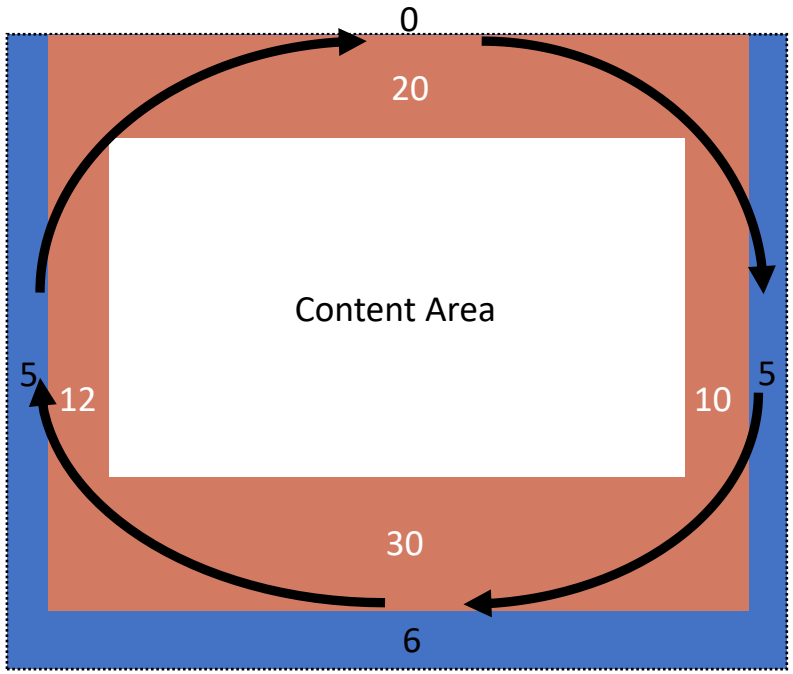


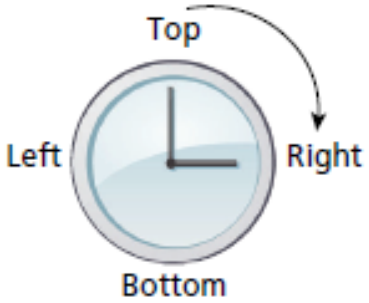
CSS Box Model

CSS Shorthand: Margin & Padding

- …Its much easier to write this:
- The sequence order is always clockwise, starting from the top

```
#container {
        padding: 20px 10px 30px 12px;
        margin: 0px 5px 6px 5px;
}
```





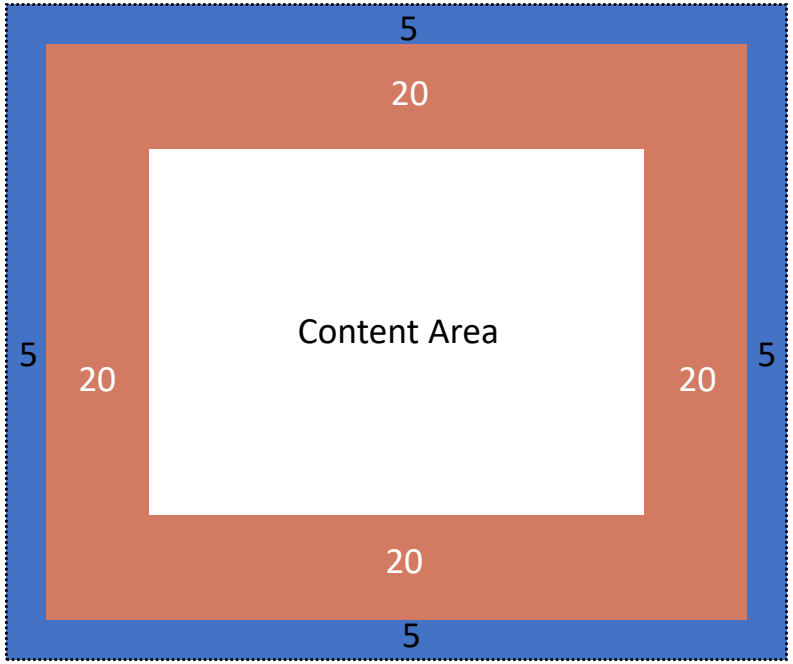
CSS Box Model

CSS Shorthand: Margin & Padding

- You can also apply just one value, example:

```
#container {
        padding: 20px;
        margin: 5px;
}
```

Which will apply the value specified equally on all 4 sides



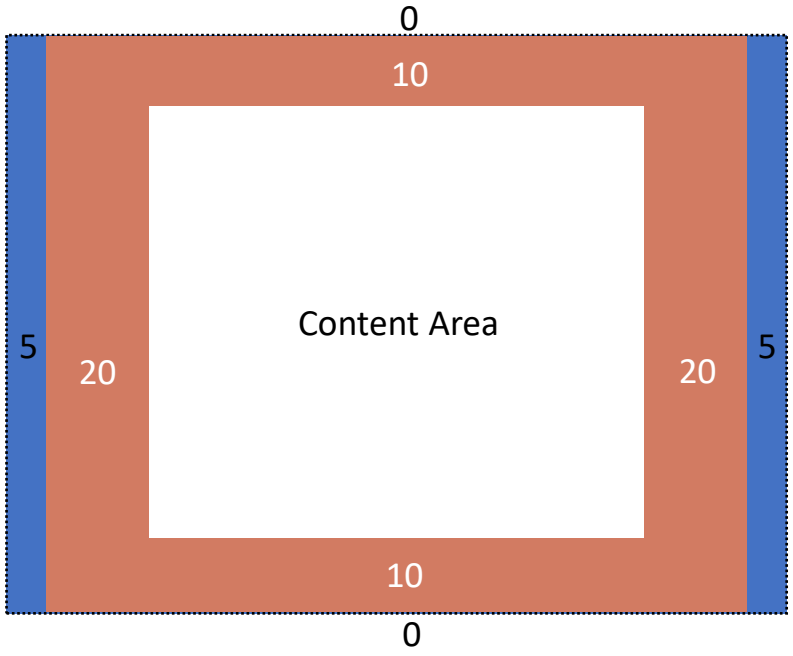
CSS Box Model

CSS Shorthand: Margin & Padding

- And you can apply two values, example:

```
#container {
        padding: 10px 20px;
        margin: 0px 5px;
}
```

- The first value is applied to the *top* and *bottom*
- The second value is applied to the *left* and *right*

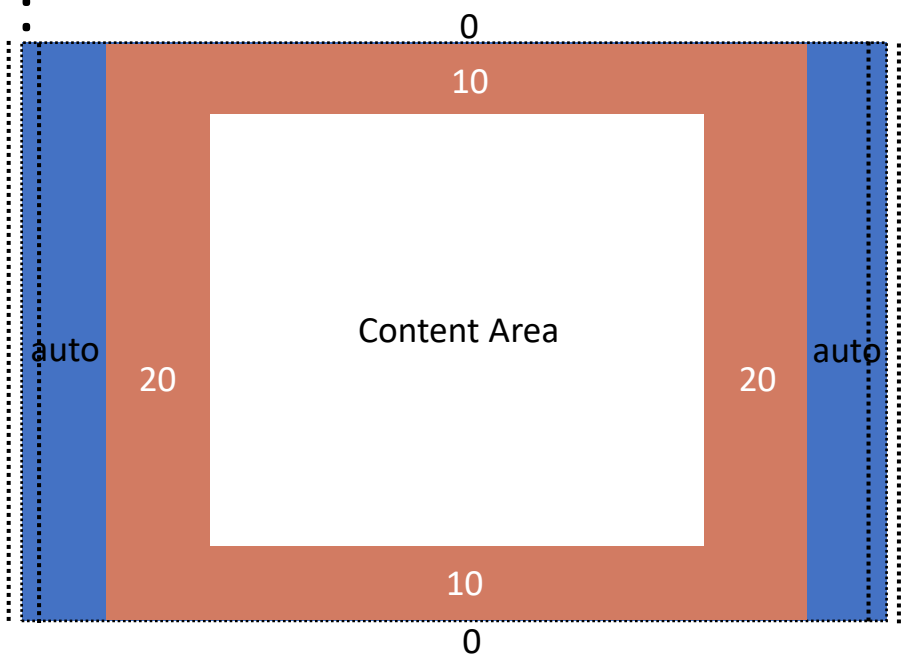


CSS Box Model

CSS Shorthand: Margin & Padding

- A useful value to remember is ‘auto’:

```
#container {
        padding: 10px 20px;
        margin: 0px auto;
}
```

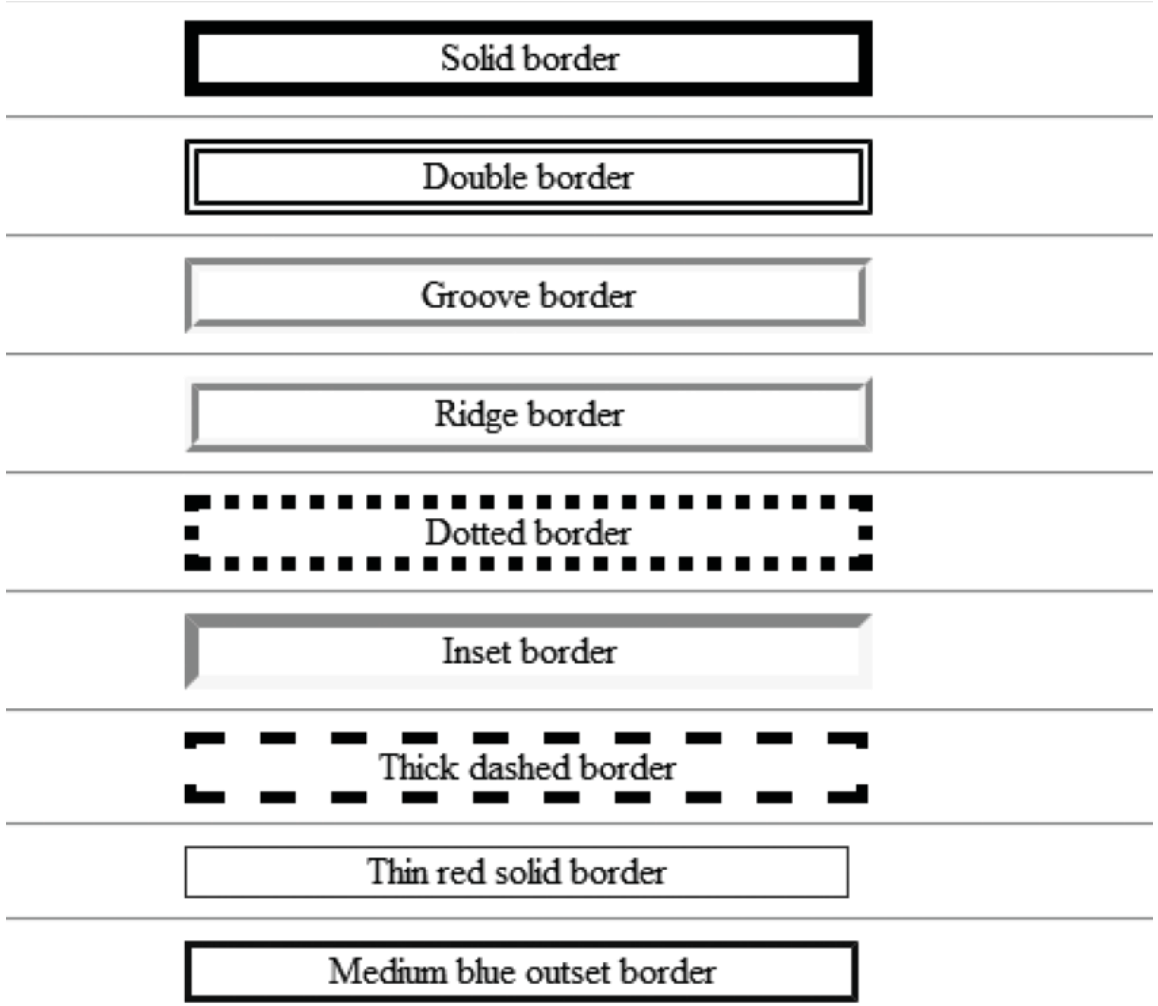


- Usually applied to the left & right areas of the margin property, auto is useful for centering a block container element in the browser window
  - The element will then take up the specified width, and the remaining space will be split equally between the left and right margins

https://www.w3schools.com/css/css_boxmodel.asp

CSS Box Model Borders

- The core border properties are:
  - Width: *absolute* (px, in, cm, or ‘thin’, ‘medium’, ‘thick’), or *relative* (em)
  - Style: dotted, dashed, solid, double, groove, ridge, inset, outset, hidden, etc.
  - Color: ‘blue’, ‘red’, #FF9900, etc.
- The border is a shorthand for border-left, border-top ….



https://www.w3schools.com/css/css_border.asp

Rounded Corners..

Property	Description
border-radius	A shorthand property for setting all the four border-*-*-radius properties
border-top-left-radius	Defines the shape of the border of the top-left corner
border-top-right-radius	Defines the shape of the border of the top-right corner
border-bottom-right-radius	Defines the shape of the border of the bottom-right corner
border-bottom-left-radius	Defines the shape of the border of the bottom-left corner

How CSS Work

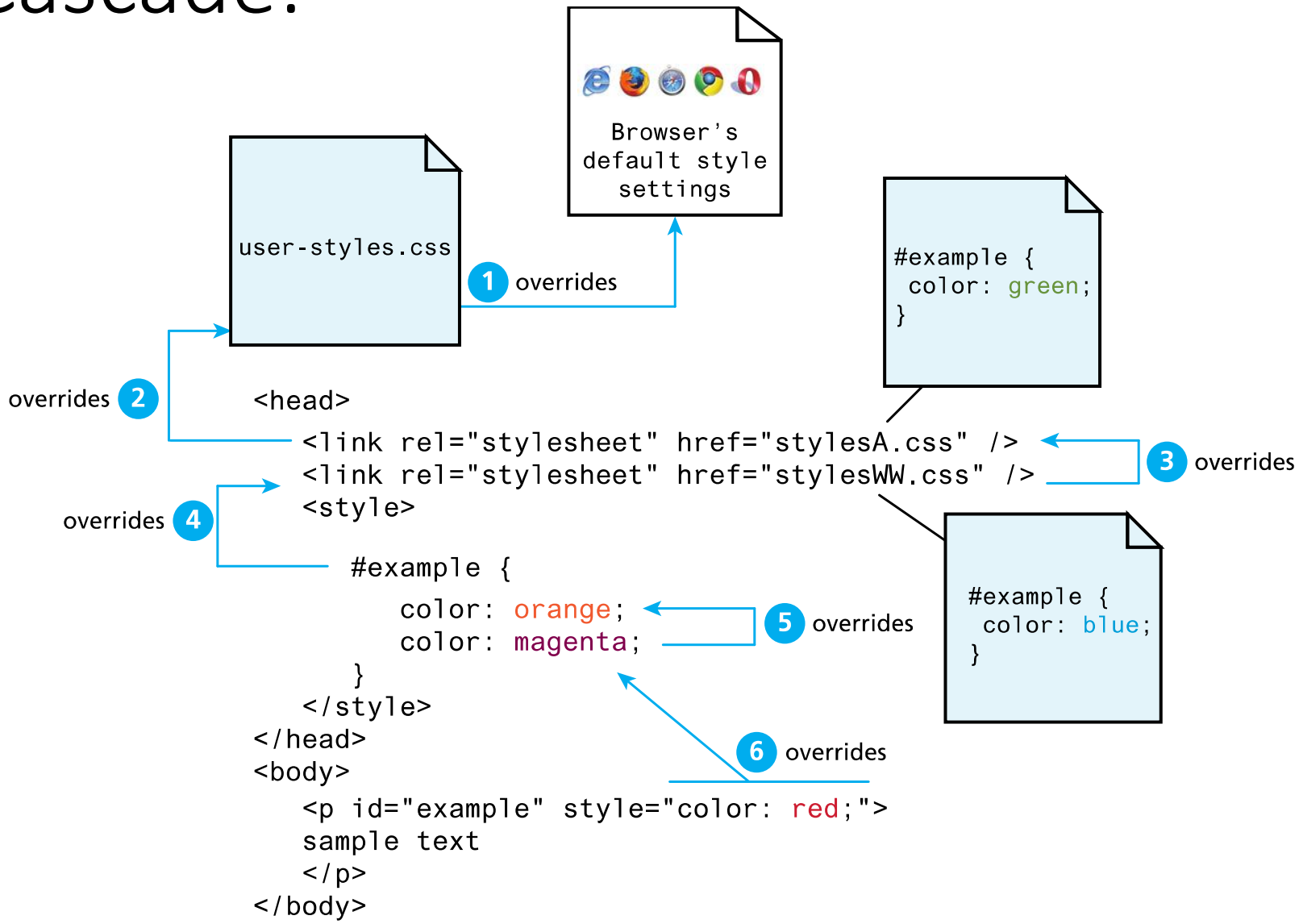
Cascade

Inheritance

Specificity

The Cascade:

Location



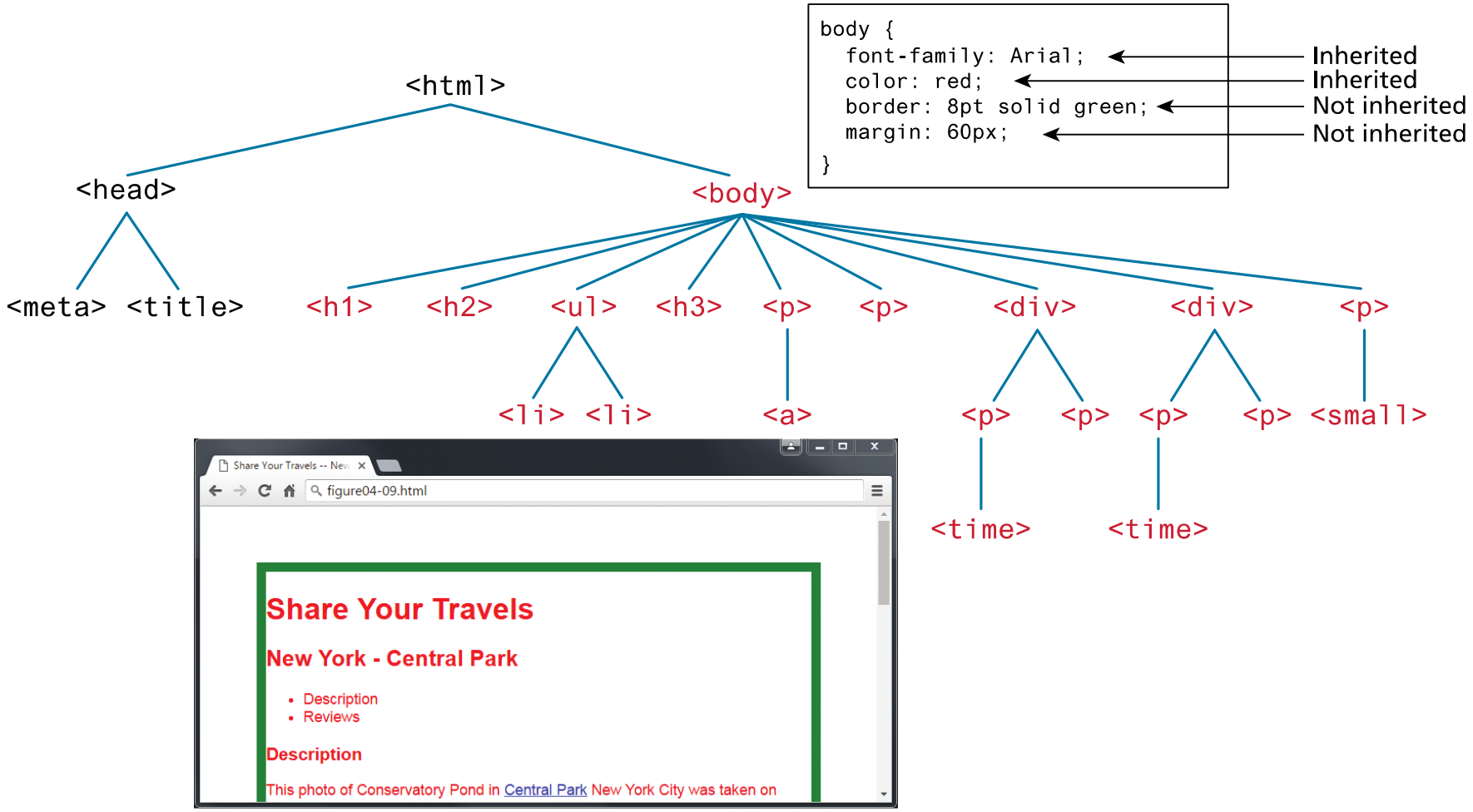
The Cascade:

Inheritance

- Many (but not all) CSS properties affect not only themselves but their descendants as well.
  - Are inheritable:
    - Font,
    - color,
    - list, and
    - text properties
  - Not inheritable:
    - layout,
    - sizing,
    - border,
    - background, and
    - spacing properties

The Cascade:

Inheritance



The Cascade:

Specificity

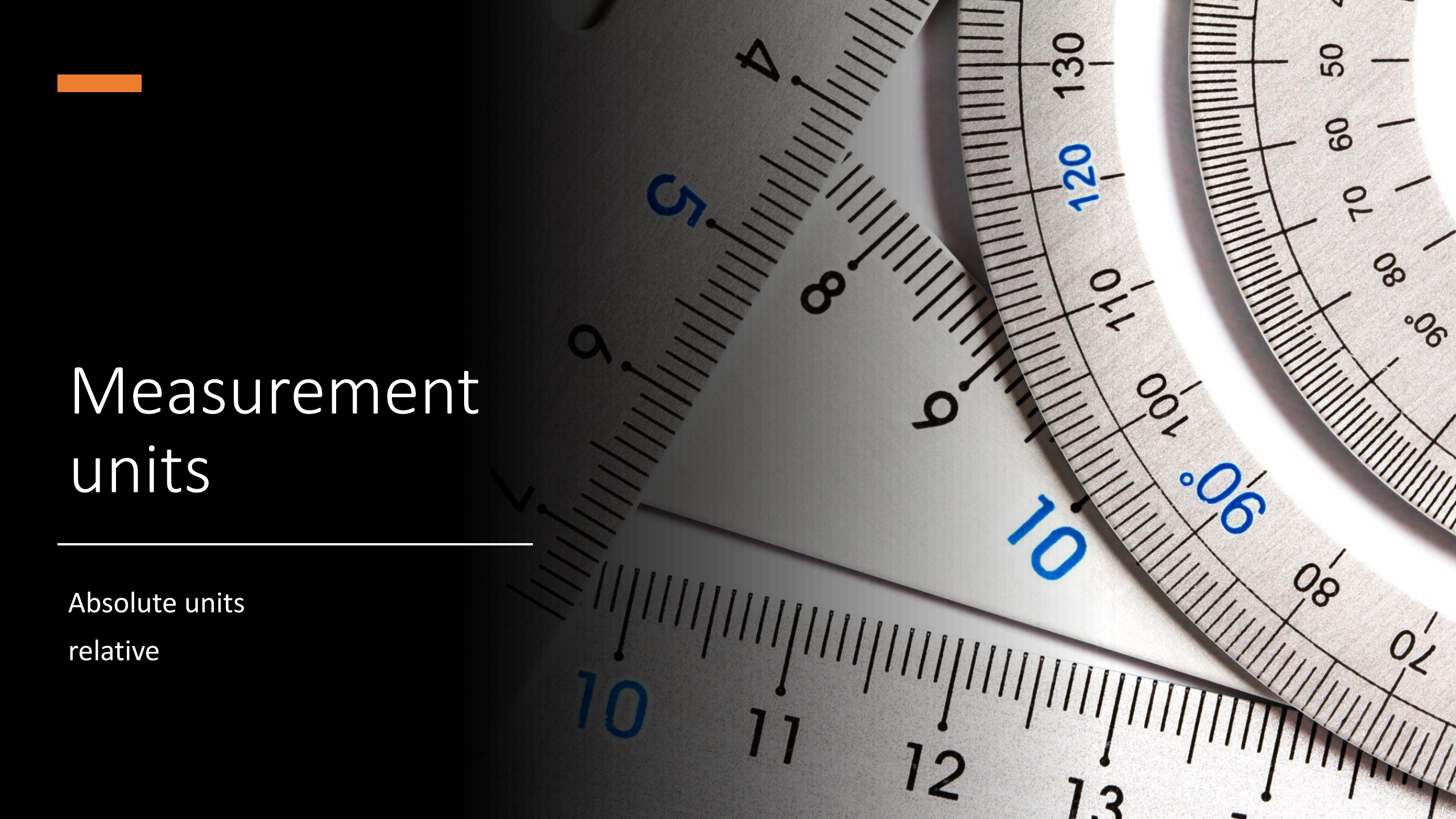
- Specificity is how the browser determines which style rule takes precedence when more than one style rule could be applied to the same element.
- In CSS, the more specific the selector, the more it takes precedence (i.e., overrides the previous definition).
- Properties defined for child and descendant elements have a higher specificity than properties defined for parent and ancestor elements.
- Conflicts are resolved in favor of properties with a higher specificity, so the child's styles take precedence.
- Here is a nice article explaining specificity

https://www.w3schools.com/css/css_specificity.asp

Measurement units

Absolute units

relative



Use relative units when possible

Font relative units

- 5 em , 10 rem

Window relative units

- 20 vw, 10 vh

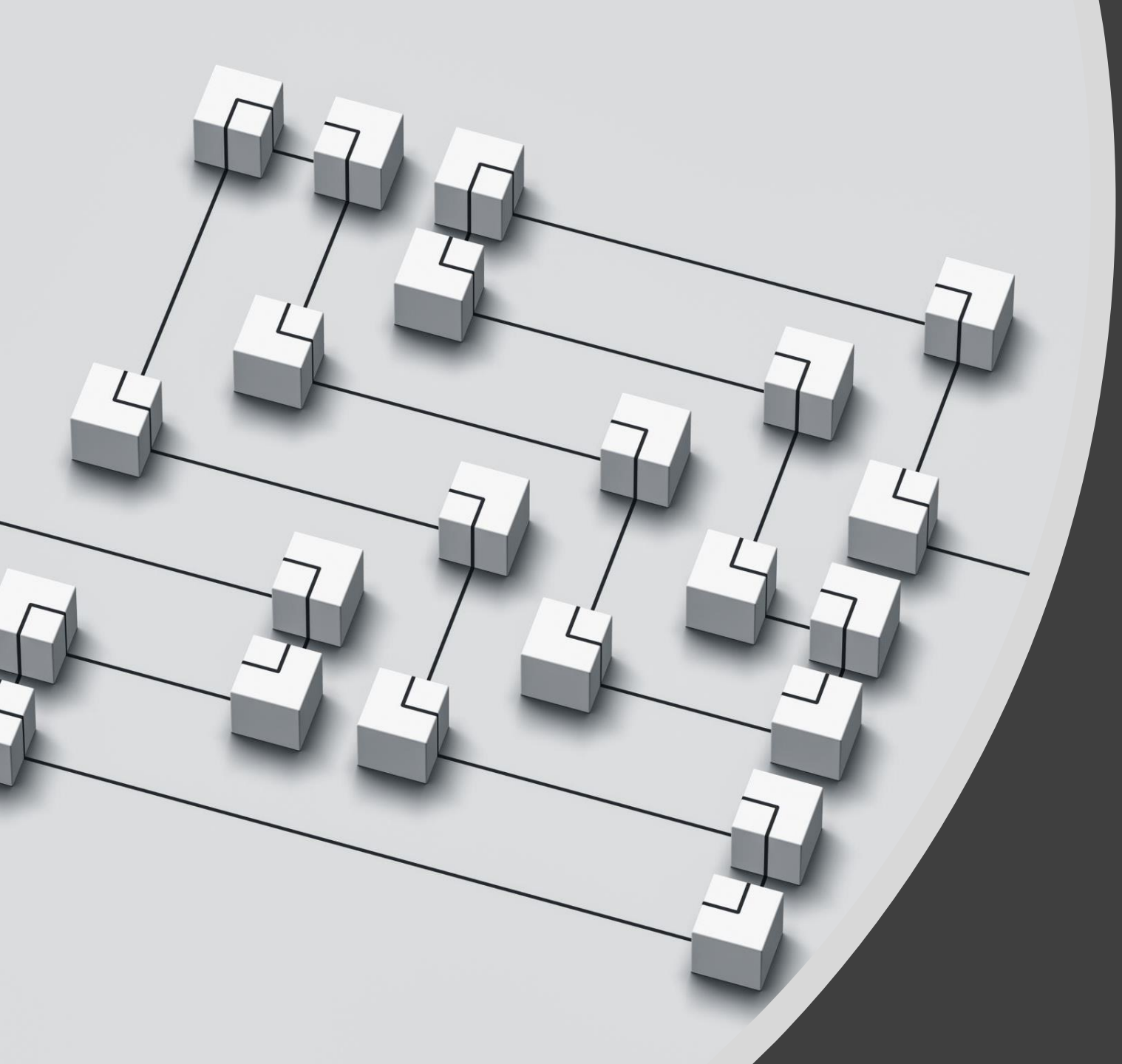
Percentages relative

- 50%

Absolute units

- 50 px, 20 cm

https://www.w3schools.com/cssref/css_units.asp



CSS Layout

Positioning

Grid

Positioning Elements

position: static

- Default value. Elements render in order, as they appear in the document flow

position: fixed

- Puts an element at an exact position within the browser window

position: absolute

- Puts an element at an absolute position based on the location of the element's parent container

position: relative

- Makes children positioned relative to the parent container
- Handy for sticking a footer to the bottom of a page, for example

position: sticky

- A "hybrid" - toggles between relative and fixed depending on scroll position

Another good explanation is here

https://www.w3schools.com/css/css_positioning.asp

The display property

inline:	• Displays an element as an inline element, spanning the width/height of its content. Any height and width properties will have no effect.
block:	• Displays an element as a block element. It starts on a new line, and takes up the width of the parent container.
initial:	• Initial or default display value.
none:	• The element is completely removed.
flex:	• Displays an element as a block-level flex container.
grid:	• Displays elements in a 2-dimensional grid.

https://www.w3schools.com/cssref/pr_class_display.asp

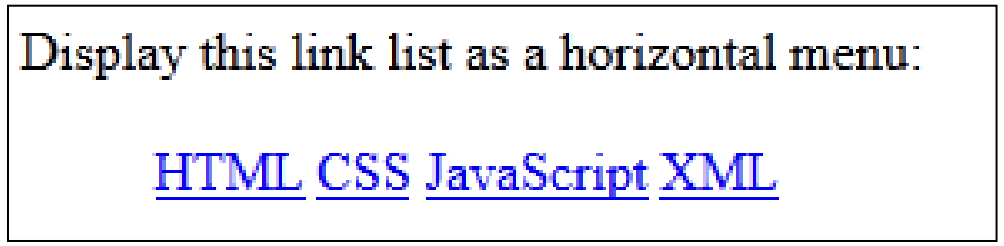
Display Property Example 1

```
<html>
<head>
<style type="text/css">
li {
display:inline;
}
</style>
</head>
<body>

<p>Display this link list as a horizontal menu:</p>

<ul>
<li><a href="/html/default.asp" target="_blank">HTML</a></li>
<li><a href="/css/default.asp" target="_blank">CSS</a></li>
<li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
<li><a href="/xml/default.asp" target="_blank">XML</a></li>
</ul>

</body>
</html>
```

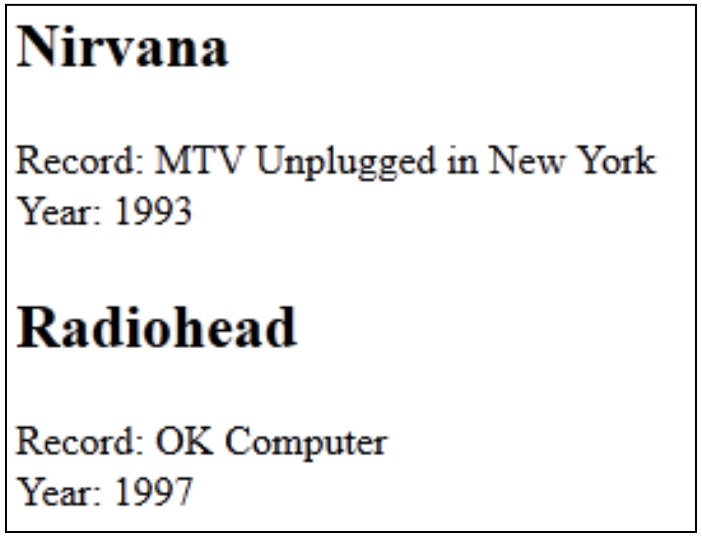


Display Property Example 2

```
<html>
<head>
<style type="text/css">
span
{
display:block;
}
</style>
</head>
<body>

<h2>Nirvana</h2>
<span>Record: MTV Unplugged in New York</span>
<span>Year: 1993</span>
<h2>Radiohead</h2>
<span>Record: OK Computer</span>
<span>Year: 1997</span>

</body>
</html>
```



Flex

- One-dimensional layout model, that offers space distribution between items


```
.box {
  display: flex;
}
```

```
<div class="box">
  <div>One</div>
  <div>Two</div>
  <div>Three has extra text
  </div>
</div>
```

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

Flex

- Flex-direction


```
.box {
  display: flex;
  flex-direction: row-reverse;
}
```

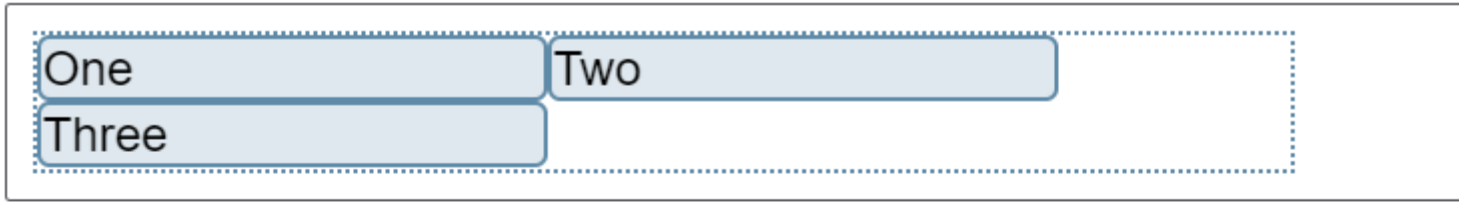
```
<div class="box">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</div>
```

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_flex-direction

Flex

- Flex-wrap



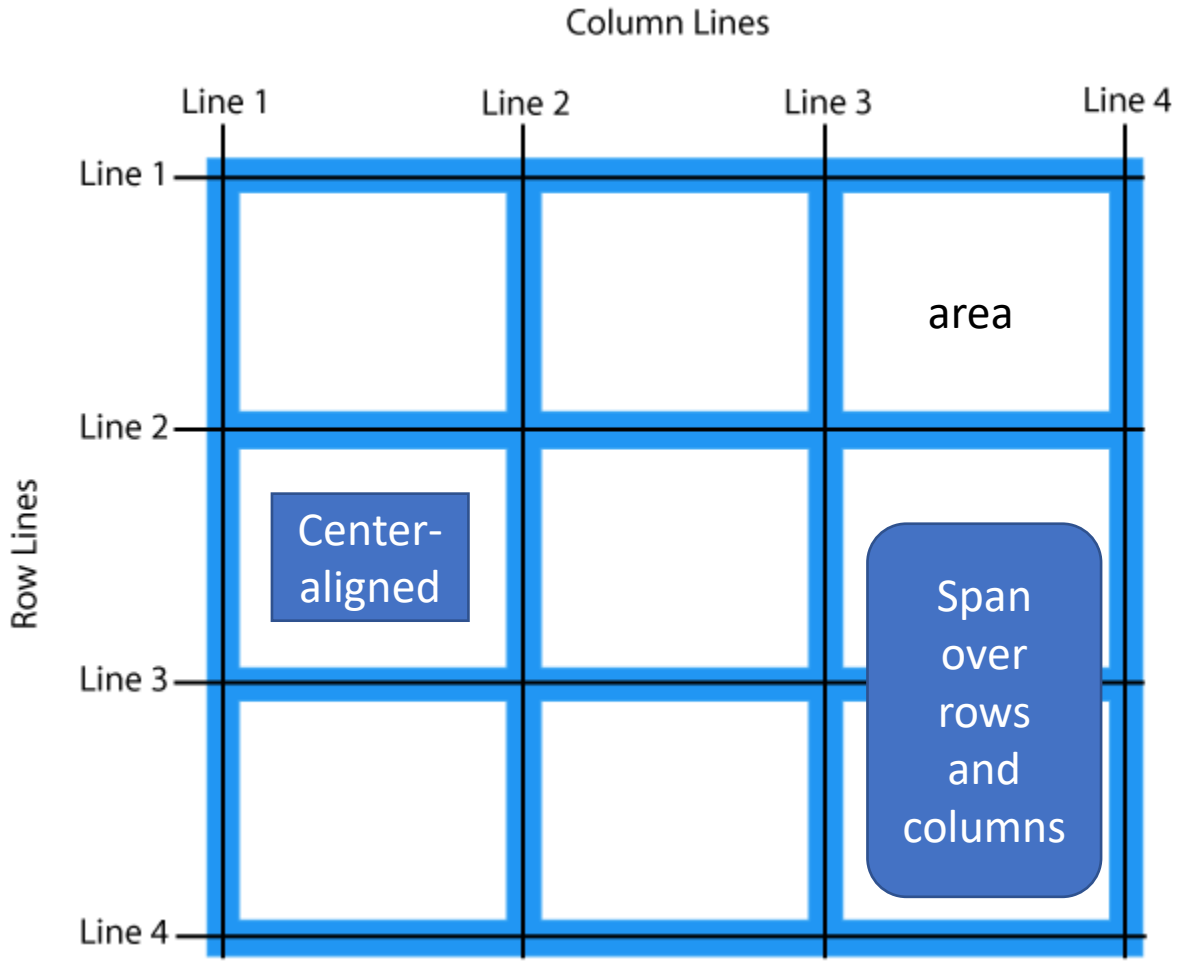

```
.box {
  display: flex;
  flex-wrap: wrap;
}
```

```
<div class="box">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</div>
```

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

Grid

- A grid layout consists of a parent element, with one or more child elements.
- Fixed and flexible track sizes (px)
- Item placement: line-numbers, area
- Items alignment to the cell
- Spanning over more cells
- Gaps



https://www.w3schools.com/css/css_grid.asp
https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout

Grid

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>

<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

1	2	3
4	5	6
7	8	9

https://www.w3schools.com/css/tryit.asp?filename=trycss_grid

Grid

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-column-start: 1;
  grid-column-end: 3;
}
</style>
```

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
</div>
```

1		2
3	4	5
6	7	8

https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_lines