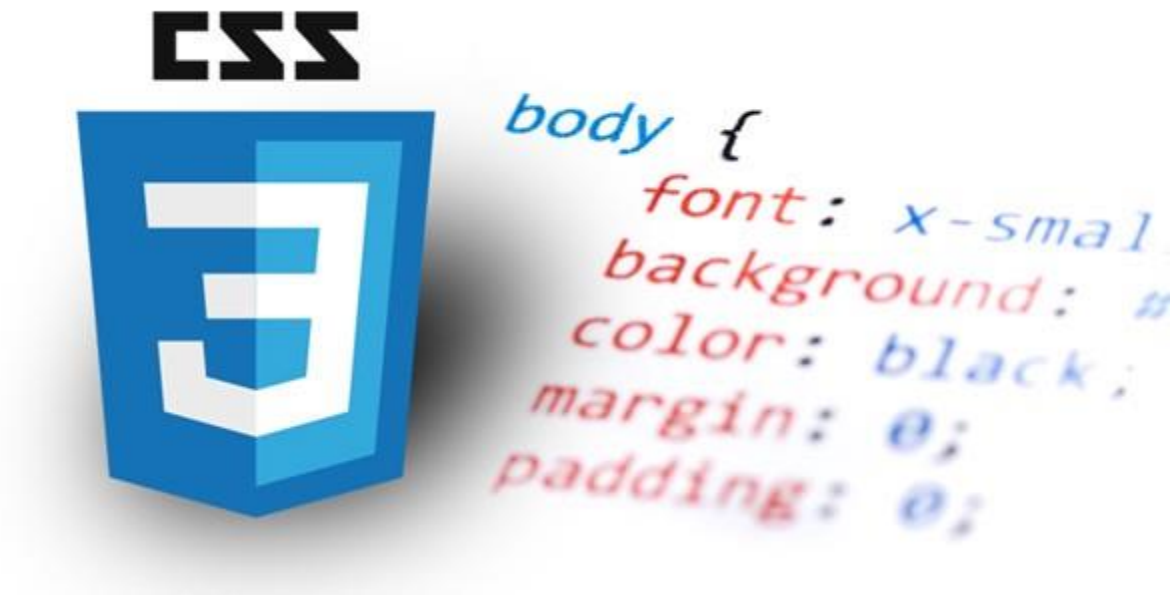

CSS
body {
font: x-smal
background:
color: black;
margin: 0;
padding: 0;

Outline

1. CSS Syntax
2. CSS Selectors
3. Basic Text Styles
4. Margin, Border and Padding

CSS Syntax

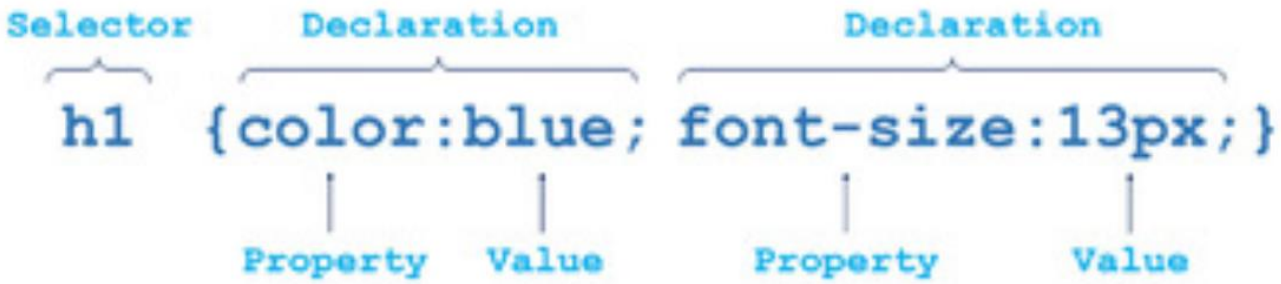
Back

CSS – Cascading Style Sheets

- CSS is used to control the **presentation** (*look and feel*) and **layout** (*positioning*) of web page elements
- Allows **separating** web page content from its design and **visual appearance**
- Used in conjunction with HTML
  - **HTML** is used for describing the **content** of a web page, **CSS** is used for describing its **presentation**
- CSS a flexible, cross-platform, **standards-based** styling language developed by the **W3C**

Style Sheets Syntax

- Stylesheets consist of **rules** that describe the styling to be applied
  - Each rule has **selectors** and declarations. A declaration specifies a **property** and its **value**



  - Declarations are separated by `semicolons`
  - Properties and values are separated by `colons`
- Comment in CSS **/* comment */**

Ways to incorporate CSS in an HTML document

- **Inline** – style included as the attribute of an HTML tag:

```
<p style="color:sienna;margin-left:20px;">This is a paragraph.</p>
```

- **Embedded** – CSS rules are contained in the head section:

```
<head>
<style>
p {color:sienna; margin-left:20px;}
body {text-align:center;}
</style>
</head>
```

- **External** - separate .css file referenced in the HTML:

HTML source code:

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

</head>
```

'main.css':

```
p {color:sienna;
margin-left:20px;}
body {text-align:center;}
```

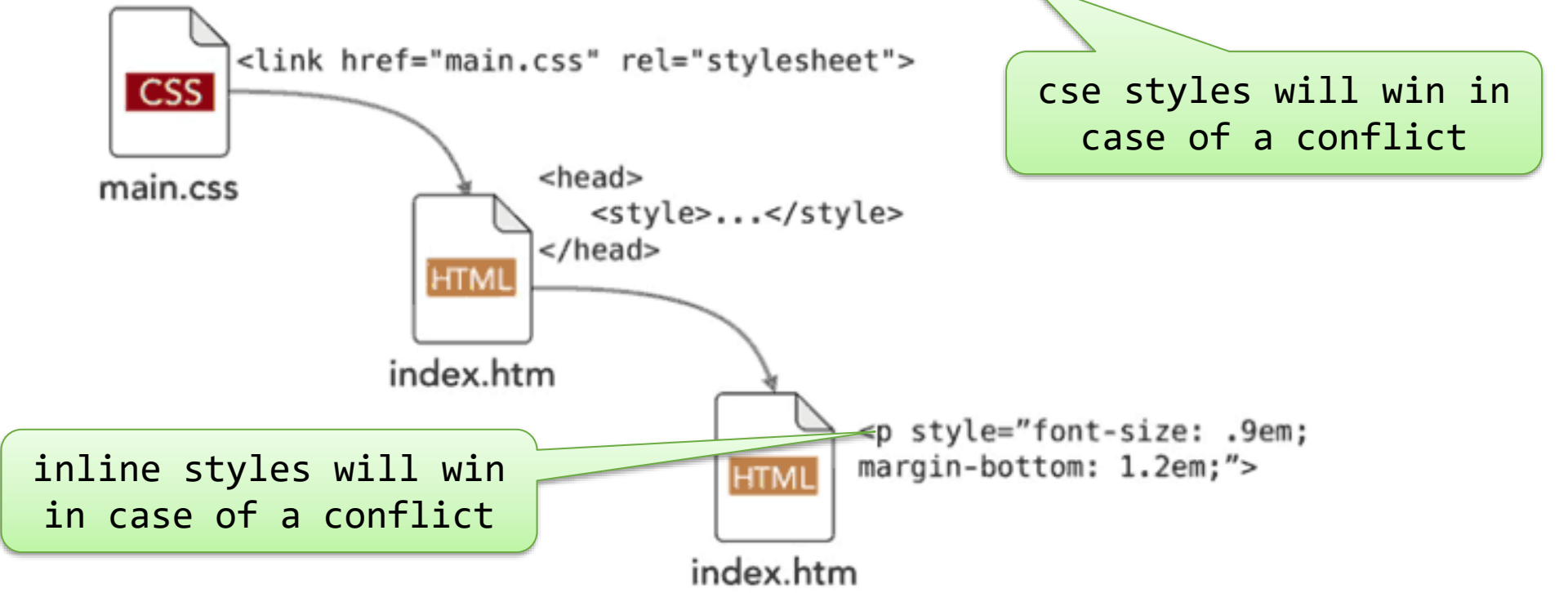
√ Ensure **consistent look and feel**

√ Improve **reusability** and **maintainability**

The “Cascade”

- The **last rule applied wins**: Styles are applied in the order that they're found, and last styles always overwrite earlier styles in case of a conflict

```
<link rel="stylesheet" href="styles/qu-styles.css">
<link rel="stylesheet" href="styles/ceng-styles.css">
<link rel="stylesheet" href="styles/cse-styles.css">
```



Inheritance

- Inheritance means that child elements will **inherit** the properties applied to a parent element

- Body selector has a font, font-size, and color applied to it:
  - Every element **inherits** those styles from the body selector
  - The style applied to h1 **overrides** the inherited color => h1 element appears in red berry color

CSS Selectors

Selectors: used to select elements to style on an HTML page

- **Element Selectors** - matches a HTML element directly
  - Apply page-wide

  e.g., `p { font-family: verdana; }` applies the style to all <p> tags

- **Class Selectors**

  - Defines a **named** style (prefix the name with dot (.))
  - Can apply to any page element using the class attribute

  e.g., `.redBorder {border: 1px solid red}` defines a style named redBorder

```
<p class='redBorder'>Using the class attribute to
apply the redBoder style to this paragrpah</p>
```

- **ID Selectors**

  - Apply to one specific tag
  - Use hash (#) followed by the tag id to select the element to be styled
  - Good for linking to specific part of a page

  e.g., `#errorMsg { color: red; }` apply the style to the element with id **errorMsg**

Universal

- [*]: Selects all elements
  - Used to override the browser default styles

```
* {
        font-family: "Times New Roman";
        color: blue;
}
```

An element can have more than one class

```html
<h1 class="red bold">Red and Bold</h1>
<h1 class="blue bold">Blue and Bold</h1>
<h1 class="bold">Bold</h1>
<h1 class="blue">Blue</h1>
```

```css
.red {
  color: red;
}

.blue {
  color: blue;
}

.bold {
  font-weight: bold;
}
```

- Class Selector selects all elements whose **class attribute** matches the selector
  - The selector starts with a period (.) followed by the class name
  - An element can have more than one class

Combined Selectors

element, element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> descendant elements inside elements
element > element	div > p	Selects all <p> elements that are **direct child** of element

e.g.,

```
li a {text-decoration: none}
```

- This will match all <a> tags that are inside of <li>

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

Attribute Selectors

- Selects the elements whose attribute-value pair matches the selector

[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 *= "qu"]	Selects every <a> element whose href attribute value contains "qu"

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

Pseudo-classes

- Pseudo-classes to define **element state**
  - `:link`, `:hover`, `:visited`, `:active`

`a:hover { color: red; }` `->` Style link on mouse over

`:link` pseudo-class selects anchors tags that were not visited

- Pseudo-elements to **insert content** around the selected element

```
p:before { content: "«"; }
```

Insert « before the content of each <p> element

```
p:after { content: "»"; }
```

Insert » after the content of each <p> element

Structural Pseudo-classes

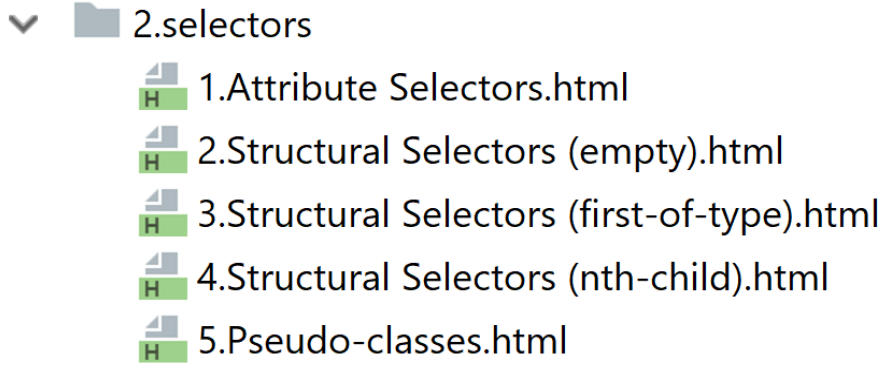
:first-child	tr:**first-child**	First row of an HTML table
:last-child	tr:**last-child**	Last row of an HTML table
:nth-child(*n*)	tr:**nth-child(2)**	Second row of an HTML table
:nth-last-child(*n*)	tr:**nth-last-child(2)**	Second row of an HTML table, counting from the last row
:nth-child(odd)	tr:**nth-child(odd)**	Every odd row of an HTML table
:nth-child(even)	tr:**nth-child(even)**	Every even row of an HTML table

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

Selectors Summary

- A style consists of a selector, followed by property/value pairs
- Selectors:
  - Element Selectors
  - Class Selectors
  - ID Selectors
  - Combined Selectors
  - Attribute selectors
  - Pseudo-elements
  - Structural pseudo-classes

Examples



Specificity

```html
<h1 id="my-heading" class="red">What colour am I?</h1>
```

```css
.red {
  color: red;
}

#my-heading {
  color: blue;
}

h1 {
  color: green;
}
```

- We have 3 selectors for the h1 with different colors
  - What will the color be?

  => h1 would be **blue** because the ID selector is more specific than the others
- The style for a **more specific selector** will be **applied** to the element

Basic Text Styles

Text-related CSS Properties

- `color` – specifies the color of the text
- `font-size` – size of font: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`… or numeric value
- `font-family` – comma separated font names
  - Example: `verdana`, `sans-serif`, …
  - The browser loads the first one that is available
- `font-weight` can be `normal`, `bold`….

Text-related CSS Properties (2)

- `font-style` – styles the font
  - Values: `normal`, `italic`, `oblique`
- `text-decoration` – decorates the text
  - Values: `none`, `underline`, `line-trough...`
- `text-align` – defines the alignment
  - Values: `left`, `right`, `center`, `justify`

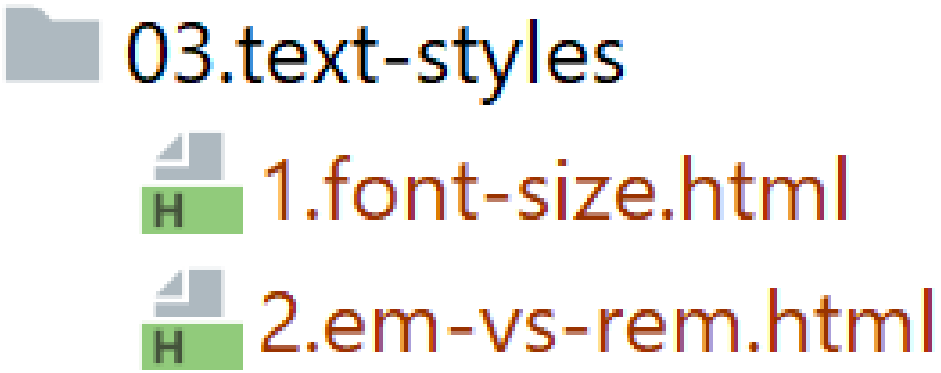
Font Sizing

Two ways to define font sizes in CSS

- **Fixed Font Sizing**: defines the size of fonts using absolute units such as points (**`pt`**) or pixels (**`px`**)
- **Relative Font Sizing**: defines the size of fonts using relative units such as **`em`** and **`rem`** units
  - This allows fonts to scale appropriately to different resolutions, browsers or platforms
  - **`em`** is relative to the font-size of its direct or nearest parent
  - **`rem`** is relative to the font-size of the html (root) element

Relative Font Sizing

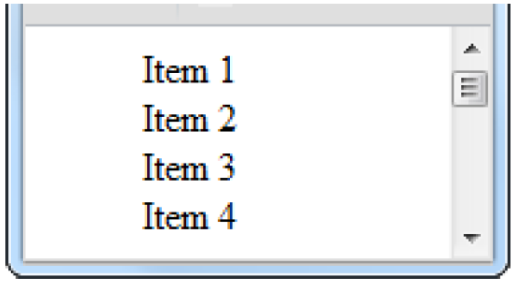
```
body {font-size: .8em;
   font-family: Verdana, Helvetica, Sans-Serif;}
h1   {font-size: 1.2em;}
div  {font-size: .8em;}
li   {font-size: .8em;}
p    {font-size: .8em;}
```



Styles for Lists

- List properties are used to define the look and feel of the list items
  - Values for <ul>: circle, square,…
  - Values for <ol>: upper-roman, lower-alpha
  - Values for both: none

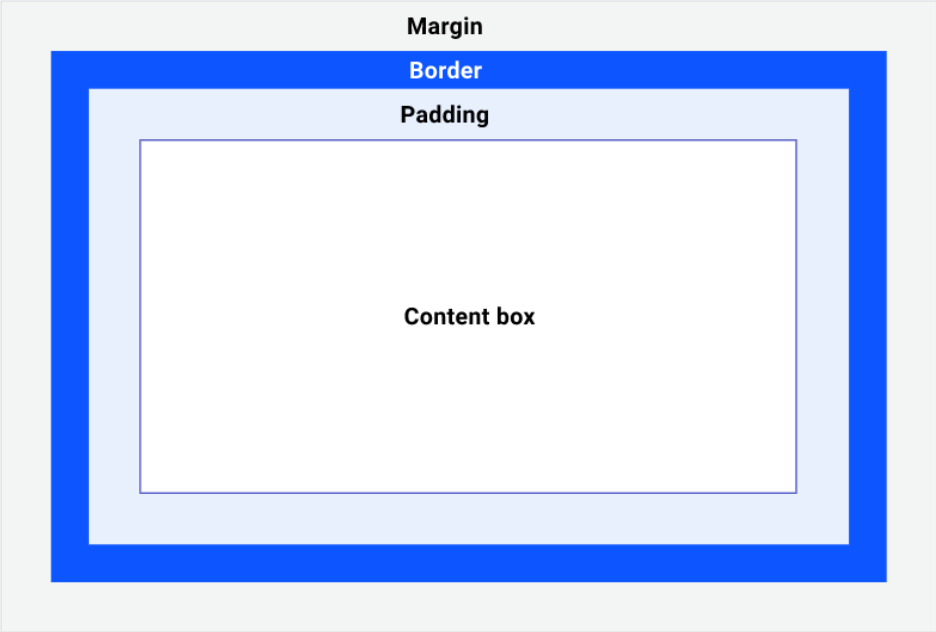
```
ul
{
    list-style-type:none;
}
```

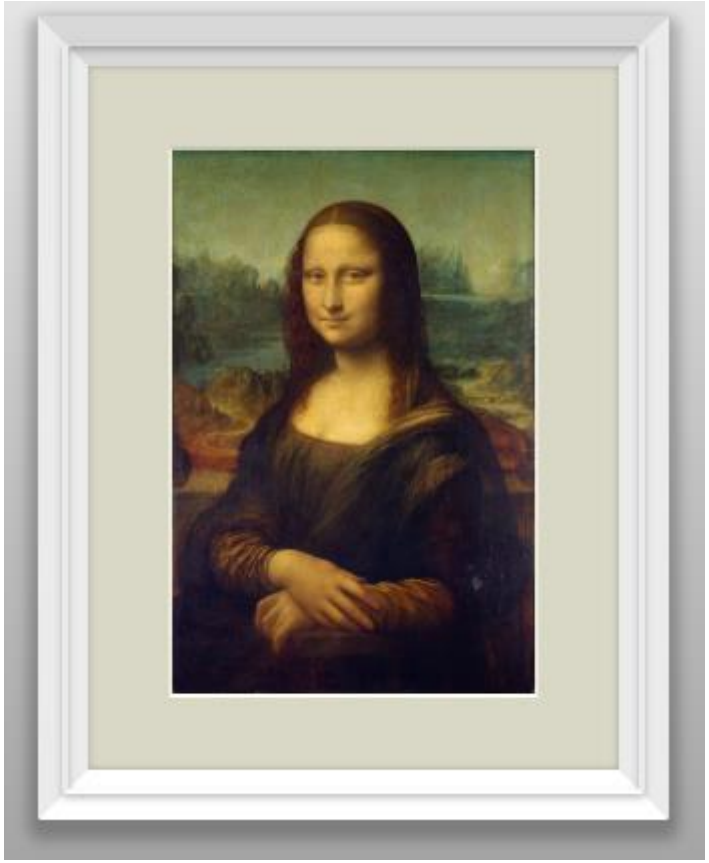


Practice …

- Use the W3Schools try-it-yourself editor to try styling each of these properties
- Background
  http://www.w3schools.com/css/css_background.asp
- Text
  http://www.w3schools.com/css/css_text.asp
- Fonts
  http://www.w3schools.com/css/css_font.asp
- Lists
  https://www.w3schools.com/css/css_list.asp

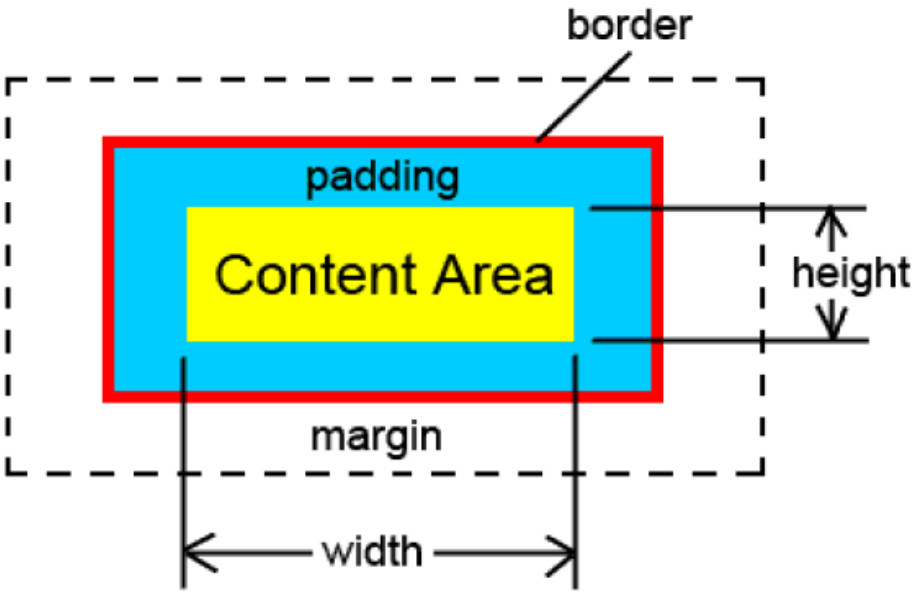
Margin, Border and Padding





Back

Box Model



- Each tag is a box, and its properties can be styled:
  - **Margin** - the space that separates the boxes
  - **Border** - the line around each edge of the box
  - **Padding** - the space between the border and the contents

Margin and Padding

- Margin and padding define the spacing around the element
  - Numerical value, e.g. **`0.8rem`**
  - Can be defined for each of the four sides separately: **margin-top**, **padding-left**, … or using short rules:
- margin: 0.8rem;
  - Sets all four sides to have margin of **0.8rem**
- margin: 10px 20px;
  - top and bottom to 10px, left and right to 20px
- margin: 1px 3px 5px 7px;
  - top, right, bottom, left (clockwise from top)
- Same for padding

Borders

- Border style:

```
border-width:1px;
border-color:red;
border-style:solid;
```

  - **border-width**: thin, medium, thick or numerical value
  - **border-color**: color alias or RGB value
  - **border-style**: none, dotted, dashed, solid, double, …

- Shorthand rule for setting border properties:

```
border: 1px solid red;
```

- Can specify different borders for the sides using: border-top, border-left, border-right, border-bottom

Rounded Corner & Centered Box

```
.centered-rounded-box {
    display: flex;
    column-gap: 1rem;
    align-items: center;

    padding: 0.5rem;
    border: 1px solid saddlebrown;
    border-radius: 15px;

    width:80%;
    margin: 0 auto;
}
```

Rounded Corner

Center Content

References

- CSS Tutorials http://www.w3schools.com/css/
- CSS Course https://web.dev/learn/css/
- Cheat sheet https://htmlcheatsheet.com/css/
- CSS developer guide

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

- Selectors

http://code.tutsplus.com/tutorials/the-30-css-selectors-you-must-memorize--net-16048