WEB PROGRAMMING 06016322

BY DR BUNDIT THANASOPON

CSS REVISION

WHAT IS CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets

- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**

- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once

- External stylesheets are stored in **CSS files**

CSS Saves a Lot of Work!

The style definitions are normally saved in external .css files.

With an external stylesheet file, you can change the look of an entire website by changing just one file!

Welcome to My Homepage
Use the menu to select different Stylesheets

- Stylesheet 1
- Stylesheet 2
- Stylesheet 3
- Stylesheet 4
- No Stylesheet

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
Stylesheet1, Stylesheet2, Stylesheet3, Stylesheet4.

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
No Stylesheet.

Side-Bar

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

CSS SYNTAX

- CSS Syntax - a CSS rule-set consists of a selector and a declaration block:

- 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.

- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

CSS SELECTORS

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

```css
p {
  text-align: center;
  color: red;
}


#para1 {
  text-align: center;
  color: red;
}
/* This is a single-line comment */
```

```css
.center {
  text-align: center;
  color: red;
}


p.center {
  text-align: center;
  color: red;
}


h1, h2, p {
  text-align: center;
  color: red;
}
```

THREE WAYS TO INSERT CSS

- There are three ways of inserting a style sheet:
  - **External style sheet**

```
<head>
    <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

  - **Internal style sheet**

```
<head>
    <style>
    h1 {
            color: maroon;
            margin-left: 40px;
    }
    </style>
</head>
```

  - **Inline style:** `<h1 style="color:blue;margin-left:30px;">This is a heading</h1>`

CSS BORDERS

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

- Properties of border

  – Border style, for example, dotted, dashed, solid double, groove, none, hidden, etc.

  – Border-width property specifies the width of the four borders (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-color property is used to set the color of the four borders.

- Border - Shorthand Property
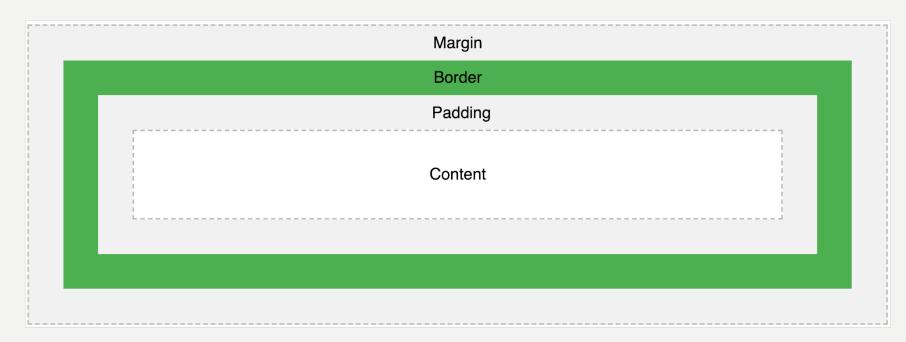
```
p { border: 5px solid red; }
```

CSS MARGINS AND PADDING

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

- CSS has properties for specifying the margin for each side of an element:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left

- Values can be "auto" or "px, pt, cm, etc." or "%" or "inherit"

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

- CSS has properties for specifying the padding for each side of an element:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left

- Values can be "px, pt, cm, etc." or "%"or "inherit"

CSS BOX MODEL

- All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

Margin

Border

Padding

Content

WIDTH AND HEIGHT OF AN ELEMENT

- **Important:** When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**. To calculate the full size of an element, you must also add padding, borders and margins.

- Assume we want to style a <div> element to have a total width of 350px.

- ```
  div {
      width: 320px;
      padding: 10px;
      border: 5px solid gray;
      margin: 0;
  }
  ```

- 320px (width) + 20px (left + right padding) + 10px (left + right border) + 0px (left + right margin) **= 350px**

CSS TABLE

- To specify table borders in CSS, use the border property.

```css
table, th, td { border: 1px solid black; }
```

- The border-collapse property sets whether the table borders should be collapsed into a single border:

```css
table { border-collapse: collapse; }
table, th, td { border: 1px solid black; }
```

- Use text-alignment and vertical-align to align content in table cells:

```css
td {
    height: 50px;
    vertical-align: bottom;
    text-alignment: center;
}
```

- Use the :hover selector on <tr> to highlight table rows on mouse over:

```css
tr:hover {background-color: #f5f5f5;}
```

CSS TABLE

- For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

- A responsive table will display a horizontal scroll bar if the screen is too small to display the full content:

- Add a container element (like <div>) with overflow-x:auto around the <table> element to make it responsive:

```
<div style="overflow-x:auto;">
 <table>
  ... table content ...
 </table>
</div>
```

EXERCISE: CREATE A NICE TABLE

Firstname	Lastname	Savings
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300
Jack	Jacky	$30

CSS POSITION

- The position property specifies the type of positioning method used for an element.
- There are five different position values:
  - **static**: the element is always positioned according to the normal flow of the page.
  - **relative**: setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
  - **fixed**: the element always stays in the same place even if the page is scrolled.
  - **absolute**: the element is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
  - **sticky**: the element is positioned based on the user's scroll position.
- Elements are then positioned using the top, bottom, left, and right properties.

CSS FLOAT

- The CSS float property specifies how an element should float.

- The float property can have one of the following values:

    - left - The element floats to the left of its container

    - right- The element floats to the right of its container

    - none - The element does not float (will be displayed just where it occurs in the text). This is default

    - inherit - The element inherits the float value of its parent

BOOTSTRAP CSS

WHAT IS BOOTSTRAP?

- Bootstrap is an open source toolkit for developing with HTML, CSS, and JS.

- It provides front-end component library for building responsive, mobile-first websites

- It is easy to use and very popular!

LET'S GET STARTED!

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap 4 Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>

</body>
</html>
```

LAYOUT

- Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**.

```
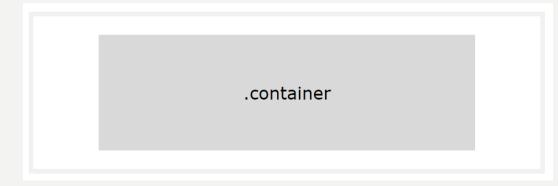<div class="container">
        <!-- Content here -->
</div>
```

```
<div class="container-fluid">
        <!-- Content here -->
</div>
```

.container

.container-fluid

RESPONSIVE BREAKPOINTS

- Bootstrap primarily uses the following media query ranges—or breakpoints

// Extra small devices (portrait phones, less than 576px) => xs

// Small devices (landscape phones, 576px and up) @media (min-width: 576px) { ... } => sm

// Medium devices (tablets, 768px and up) @media (min-width: 768px) { ... } => md

// Large devices (desktops, 992px and up) @media (min-width: 992px) { ... } => lg

// Extra large devices (large desktops, 1200px and up) @media (min-width: 1200px) { ... } => xl

GRID SYSTEM

- Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with flexbox and is fully responsive.

One of three columns	One of three columns	One of three columns

Copy

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

GRID OPTIONS

	Extra small <576px	**Small ≥576px**	**Medium ≥768px**	**Large ≥992px**	**Extra large ≥1200px**
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

EQUAL-WIDTH

- Add any number of unit-less classes for each breakpoint you need and every column will be the same width.

```
<div class="container">
  <div class="row">
    <div class="col"> 1 of 3 </div>
    <div class="col"> 2 of 3 </div>
    <div class="col"> 3 of 3 </div>
  </div>
</div>
```

SETTING ONE COLUMN WIDTH

- Auto-layout for flexbox grid columns also means you can set the width of one column and have the sibling columns automatically resize around it.

```
<div class="container">
  <div class="row">
    <div class="col"> 1 of 3 </div>
    <div class="col-6"> 2 of 3 (wider) </div>
    <div class="col"> 3 of 3 </div>
  </div>
</div>
```

VARIABLE WIDTH CONTENT

- Use col-{breakpoint}-auto classes to size columns based on the natural width of their content.

```
<div class="container">
  <div class="row">
    <div class="col"> 1 of 3 </div>
    <div class="col-md-auto"> 2 of 3 (wider) </div>
    <div class="col col-lg-2"> 3 of 3 </div>
  </div>
</div>
```

USE .W-100 TO BREAK A NEW LINE

```html
<div class="container">
  <div class="row">
    <div class="col"> 1 of 3 </div>
    <div class="col"> 2 of 3 </div>
    <div class=”w-100"></div>
    <div class="col"> 3 of 3 </div>
    <div class="col"> 3 of 3 </div>
  </div>
</div>
```

MIX AND MATCH

```html
<div class="container">
<!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-12 col-md-8">.col-12 .col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>
<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div> <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```

TYPOGRAPHY

- Headers

  `<p class="h1">`h1. Bootstrap heading`</p>`

  `<p class="h2">`h2. Bootstrap `<small class="text-muted">`heading `<small>``</p>`

  `<p class="h3">`h3. Bootstrap heading`</p>`

  `<p class="h4">`h4. Bootstrap heading`</p>`

  `<p class="h5">`h5. Bootstrap heading`</p>`

  `<p class="h6">`h6. Bootstrap heading`</p>`

- Display headings

  `<h1 class="display-1">`Display 1`</h1>`

  `<h1 class="display-2">`Display 2`</h1>`

  `<h1 class="display-3">`Display 3`</h1>`

  `<h1 class="display-4">`Display 4`</h1>`

TYPOGRAPHY

<p>You can use the mark tag to <mark>highlight</mark> text.</p>

<p>~~This line of text is meant to be treated as deleted text.~~</p>

<p><s>This line of text is meant to be treated as no longer accurate.</s></p>

<p><ins>This line of text is meant to be treated as an addition to the document.</ins></p>

<p><u>This line of text will render as underlined</u></p>

<p><small>This line of text is meant to be treated as fine print.</small></p>

<p>This line rendered as bold text.</p>

<p>This line rendered as italicized text.</p>

<p class="lead">Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus. </p>

TEXT ALIGNMENT

Try change this to "text-right"

```
<blockquote class="blockquote text-center">
  <p class="mb-0">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.
  </p>
  <footer class="blockquote-footer">
    Someone famous in <cite title="Source Title">Source Title</cite>
  </footer>
</blockquote>
```

IMAGES

- Responsive images
  - max-width: 100%; and height: auto; are applied to the image so that it scales with the parent element.

  `<img src="..." class="img-fluid" alt="Responsive image">`

- Image thumbnails

  `<img src="..." alt="..." class="img-thumbnail">`

- Aligning images

  `<img src="..." class="rounded float-left" alt="...">`

  `<img src="..." class="rounded float-right" alt="...">`

  `<img src="..." class="rounded mx-auto d-block" alt="...">`

TABLES

- Just add class "table" to table tag

    <table class="table">

- You can also invert the colors—with light text on dark backgrounds—with .table-dark.

    <table class="table-dark">

- Similar to tables and dark tables, use the modifier classes .thead-light or .thead-dark to make <thead>s appear light or dark gray.

    <thead class="thead-dark">

    <thead class="thead-light">

- Use .table-striped to add zebra-striping to any table row within the <tbody>.

    <table class="table table-striped">

CARDS

- A **card** is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options.

CARDS

```html
<div class="card" style="width: 18rem;">
  <img src="..." class="card-img-top" alt="...">
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of the card's content.
    </p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

CARDS - CONTENT TYPES

- **Body** – use it whenever you need a padded section within a card.

```
<div class="card">

  <div class="card-body">

  This is some text within a card body.

  </div>
</div>
```

- **Titles, text, and links**

```
<h5 class="card-title">Card title</h5>

<h6 class="card-subtitle mb-2 text-muted">Card subtitle</h6>

<p class="card-text">Text text text.</p>

<a href="#" class="card-link">Card link</a>
```

CARDS - CONTENT TYPES

- **Images** - .card-img-top places an image to the top of the card.

<img src="..." class="card-img-top" alt="...">

Cras justo odio
Dapibus ac facilisis in
Vestibulum at eros

- **List groups**

<div class="card" style="width: 18rem;">

<ul class="list-group list-group-flush">

  <li class="list-group-item">

    Cras justo odio

  <li class="list-group-item">

    Dapibus ac facilisis in

  <li class="list-group-item">

    Vestibulum at eros

</div>

BUTTONS

<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>

BUTTONS

- Fancy larger or smaller buttons? Add .btn-lg or .btn-sm for additional sizes.

`<button type="button" class="btn btn-primary btn-lg">Large button</button>`

`<button type="button" class="btn btn-primary btn-sm">Small button</button>`

- Create block level buttons – those that span the full width of a parent—by adding .btn-block.

`<button type="button" class="btn btn-primary btn-lg btn-block">Block level button</button>`

- Disabled state – Make buttons look inactive by adding the disabled boolean attribute to any <button> element.

`<button type="button" class="btn btn-lg btn-primary" disabled>Primary button</button>`

FORMS

```html
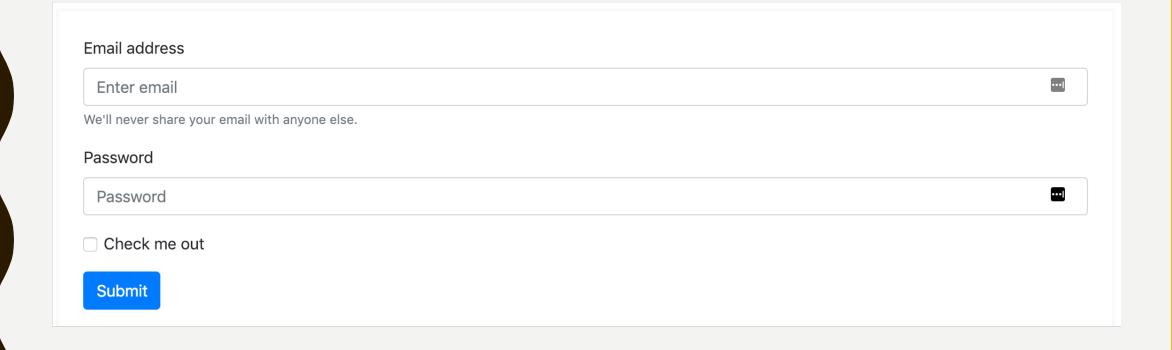<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">
    <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

FORMS

Email address

Enter email

We'll never share your email with anyone else.

Password

Password

☐ Check me out

Submit

FORMS

- Textual form controls—like <input>s, <select>s, and <textarea>s—are styled with the .form-control class.

<select class="form-control" id="exampleFormControlSelect1">

  <option>1</option>

  <option>2</option>

  <option>3</option>

</select>

<textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>

- Set heights using classes like .form-control-lg and .form-control-sm.

NAVS

Active   Link   Link   Disabled

- Navigation available in Bootstrap share general markup and styles, from the base .nav class to the active and disabled states.

```
<ul class="nav">
  <li class="nav-item"> <a class="nav-link active" href="#">Active</a> </li>
  <li class="nav-item"> <a class="nav-link" href="#">Link</a> </li>
  <li class="nav-item"> <a class="nav-link" href="#">Link</a> </li>
  <li class="nav-item"> <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a> </li>
</ul>
```