

CSS3 AND BOOTSTRAP 4

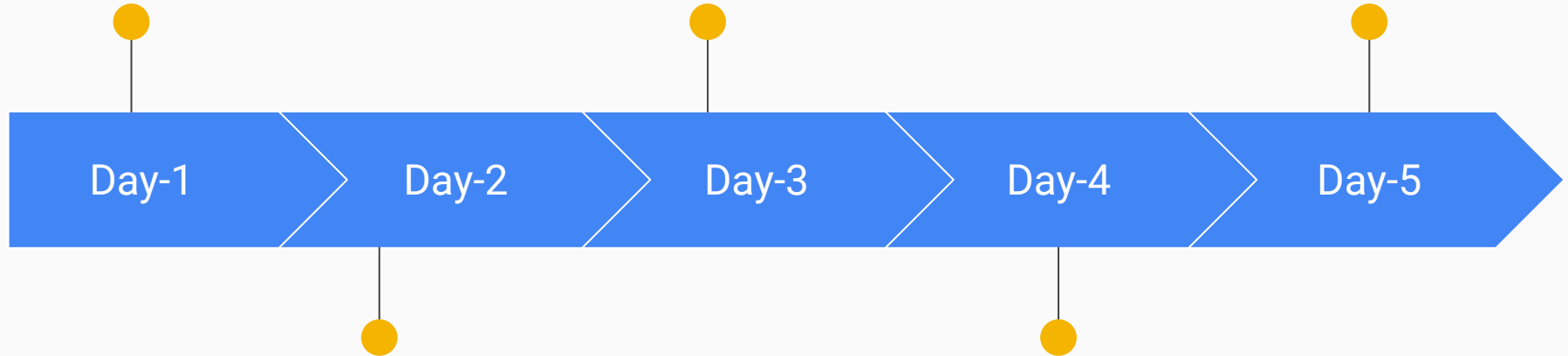
5-10 Days Training



CSS 3 Basics and
implementations
usages

CSS 3 Effects and
2D,3D,animations.

Website layouts and
all class
implementation



CSS 3 Concepts and
tasks

Bootstrap 4 basics
and usages

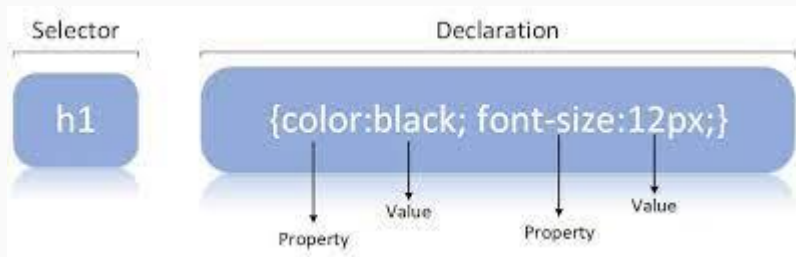
Day-1

- CSS Introduction
- CSS Syntax
- CSS Selectors
- CSS Colors
- Border,margin,padding
- Box module
- Outline
- fonts, Icons
- List, Table
- Display, Max width

What is CSS?

- CSS stands for Cascading Style Sheets
- 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

Syntax



Selectors

Selectors

1] Simple selectors (select elements based on name, id, class,*) :- `p{}`, `.test{}`, `#test{}`.

2] Pseudo-class selectors (select elements based on a certain state) :- `selector:pseudo-class`
`{property: value;}`

3] Pseudo-elements selectors (select and style a part of an element)

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

4] Attribute selectors (select elements based on an attribute or attribute value):

`Selector[attribute="value"]`

```
a[target] {
```

```
  color: purple; }
```

Colors,padding,margin,box model

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

Margin:-Margins are used to create space around elements, outside of any defined borders.

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



Height and Width:The CSS `height` and `width` properties are used to set the height and width of an element.

Display

Display: none;

Visibility Property:

```
h2.a {  
    visibility: visible;  
}
```

```
h2.b {  
    visibility: hidden;  
}
```

display:block

Outline, fonts,Icons, List, Table

Outline: An outline is a line drawn outside the element's border.

Fonts,Icons: <https://fonts.google.com/>, <https://fontawesome.com/v4.7.0/icons/>

List and Table

Unordered Lists:

- Coffee
 - Tea
 - Coca Cola
-
- Coffee
 - Tea
 - Coca Cola

Ordered Lists:

1. Coffee
 2. Tea
 3. Coca Cola
-
- I. Coffee
 - II. Tea
 - III. Coca Cola

```
table {  
  
    border: 1px solid black;  
  
}
```


Day-1 : Task

Box Model Implementation:

Display Property Implementation

Margin,Padding, Borders, Outline

List style and Table Example.

Fonts and Icon.

Day-2

- Float
- Positions
- CSS Inline-block
- CSS Pseudo-classes
- CSS Image Sprites
- CSS Forms
- CSS Specificity
- CSS The !important Rule
- CSS Website Layout

CSS Forms

Using CSS to style an HTML Form

First Name Last Name Country

Using CSS to style an HTML Form

First Name

Last Name

Country

CSS Specificity

If there are two or more conflicting CSS rules that point to the same element, the browser follows some rules to determine which one is most specific and therefore wins out.

Every selector has its place in the specificity hierarchy. There are four categories which define the specificity level of a selector:

Inline styles - An inline style is attached directly to the element to be styled. Example: `<h1 style="color: #ffffff;">.`

IDs - An ID is a unique identifier for the page elements, such as `#navbar`.

Classes, attributes and pseudo-classes - This category includes `.classes`, `[attributes]` and pseudo-classes such as `:hover`, `:focus` etc.

CSS The !important Rule

The `!important` rule in CSS is used to add more importance to a property/value than normal.

```
#myid {background-color: blue;}
```

```
.myclass {background-color: gray;}
```

```
p {  
  
    background-color: red !important;  
  
}
```

Float and Position Property

CSS Layout - float and clear

The `float` property is used for positioning and formatting content

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

Position Property

The `position` property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

The display: inline-block Value

Compared to `display: inline`, the major difference is that `display: inline-block` allows to set a width and height on the element.

Also, with `display: inline-block`, the top and bottom margins/paddings are respected, but with `display: inline` they are not.

Compared to `display: block`, the major difference is that `display: inline-block` does not add a line-break after the element, so the element can sit next to other elements.

CSS Combinators

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

Child Selector (>)

The child selector selects all elements that are the children of a specified element.

Adjacent Sibling Selector (+)

The adjacent sibling selector is used to select an element that is directly after another specific element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

General Sibling Selector (~)

The general sibling selector selects all elements that are siblings of a specified element.

Website Layout

Header

Navigation Menu

Content

Main Content

Content

Footer

Day-3

- CSS Text Effects
- CSS 2D and 3D Transforms
- CSS Animations
- CSS Multiple Columns
- CSS Media Queries
- CSS Flexbox
- CSS Responsive

Media Queries, Flexbox

CSS3 Introduced Media Queries

Media queries in CSS3 extended the CSS2 media types idea: Instead of looking for a type of device, they look at the capability of the device.

Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as iPhone and Android phones).

```
@media screen and (min-width: 480px) {  
  body {background-color: lightgreen;}
```

Flexbox

Before the Flexbox Layout module, there were four layout modes:

- Block, for sections in a webpage
- Inline, for text
- Table, for two-dimensional table data
- Positioned, for explicit position of an element

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

Text Effects, !important rule, Animations

CSS 2D Transforms Methods

With the CSS `transform` property you can use the following 2D transformation methods:

- `translate()` - Move from Current Position
- `rotate()`
- `scaleX()`
- `scaleY()`
- `scale()`
- `skewX()`
- `skewY()`
- `skew()`

CSS The !important Rule

The `!important` rule in CSS is used to add more importance to a property/value than normal.

```
#myid {background-color: blue;}
```

```
.myclass {background-color: gray;}
```

```
p {background-color: red !important;}
```

CSS 3D Transforms Methods

With the CSS `transform` property you can use the following 3D transformation methods:

- `rotateX()`
- `rotateY()`
- `rotateZ()`

Animations, Multi-column Layout

CSS Animations

CSS allows animation of HTML elements without using JavaScript or Flash!

In this chapter you will learn about the following properties:

- `@keyframes`
- `animation-name`
- `animation-duration`
- `animation-delay`
- `animation-iteration-count`
- `animation-direction`
- `animation-timing-function`
- `animation-fill-mode`
- `animation`

CSS Multi-column Layout

- `column-count`
- `column-gap`
- `column-rule-style`
- `column-rule-width`
- `column-rule-color`
- `column-rule`
- `column-span`
- `column-width`

Daily Ping

Lorem ipsum

dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim

veniam, quis nostrud exercitation 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. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

Day-4

- Bootstrap 4 Introduction
- Bootstrap 4 Containers
- Bootstrap 4 Grid System
- Bootstrap 4 Progress Bars
- Bootstrap 4 Spinners
- Bootstrap 4 Cards
- Nav Menus
- Bootstrap 4 Forms
- Bootstrap 4 Input Groups
- Plugins

Bootstrap 4



Bootstrap Container

Containers are used to pad the content inside of them, and there are two container classes available:

1. The `.container` class provides a responsive fixed width container
2. The `.container-fluid` class provides a full width container, spanning the entire width of the viewport

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
max-width	100%	540px	720px	960px	1140px

Container Padding

By default, containers have 15px left and right padding, with no top or bottom padding. Therefore, we often use spacing utilities, such as extra padding and margins to make them look even better. For example, `.pt-3` means "add a top padding of 16px":

Bootstrap 4 Grid System

Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

.col- (extra small devices - screen width less than 576px)

.col-sm- (small devices - screen width equal to or greater than 576px)

.col-md- (medium devices - screen width equal to or greater than 768px)

.col-lg- (large devices - screen width equal to or greater than 992px)

.col-xl- (xlarge devices - screen width equal to or greater than 1200px)

The classes above can be combined to create more dynamic and flexible layouts.

Note: Each class scales up, so if you wish to set the same widths for sm and md, you only need to specify sm.

Bootstrap 4 Jumbotron

A jumbotron indicates a big grey box for calling extra attention to some special content or information.

Claritech Solutions

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile-first projects on the web.

`jumbotron jumbotron-fluid`

Bootstrap 4 Alerts

Bootstrap 4 provides an easy way to create predefined alert messages:

Success! This alert box indicates a successful or positive action.



Info! This alert box indicates a neutral informative change or action.



Warning! This alert box indicates a warning that might need attention.



Danger! This alert box indicates a dangerous or potentially negative action.



Primary! This alert box indicates an important action.



Secondary! This alert box indicates a less important action.



Dark! Dark grey alert box.



Light! Light grey alert box.



Button Styles

Bootstrap 4 provides different styles of buttons:

Basic

Primary

Secondary

Success

Info

Warning

Danger

Dark

Light

Link

Example

```
<button type="button" class="btn">Basic</button>
<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-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-link">Link</button>
```

Spinners

Colored Spinners

Use any **text color utilites** to add a color to the spinner:



Example

```
<div class="spinner-border text-muted"></div>
<div class="spinner-border text-primary"></div>
<div class="spinner-border text-success"></div>
<div class="spinner-border text-info"></div>
<div class="spinner-border text-warning"></div>
<div class="spinner-border text-danger"></div>
<div class="spinner-border text-secondary"></div>
<div class="spinner-border text-dark"></div>
<div class="spinner-border text-light"></div>
```

Bootstrap Cards

A card in Bootstrap 4 is a bordered box with some padding around its content. It includes options for headers, footers, content, colors, etc.

```
<div class="card">  
  <div class="card-body">Basic card</div>  
</div>
```

Contextual Cards

To add a background color the card, use contextual classes (`.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` and `.bg-light`).

Bootstrap Cards

A card in Bootstrap 4 is a bordered box with some padding around its content. It includes options for headers, footers, content, colors, etc.

```
<div class="card">  
  <div class="card-body">Basic card</div>  
</div>
```

Contextual Cards

To add a background color the card, use contextual classes (`.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` and `.bg-light`).