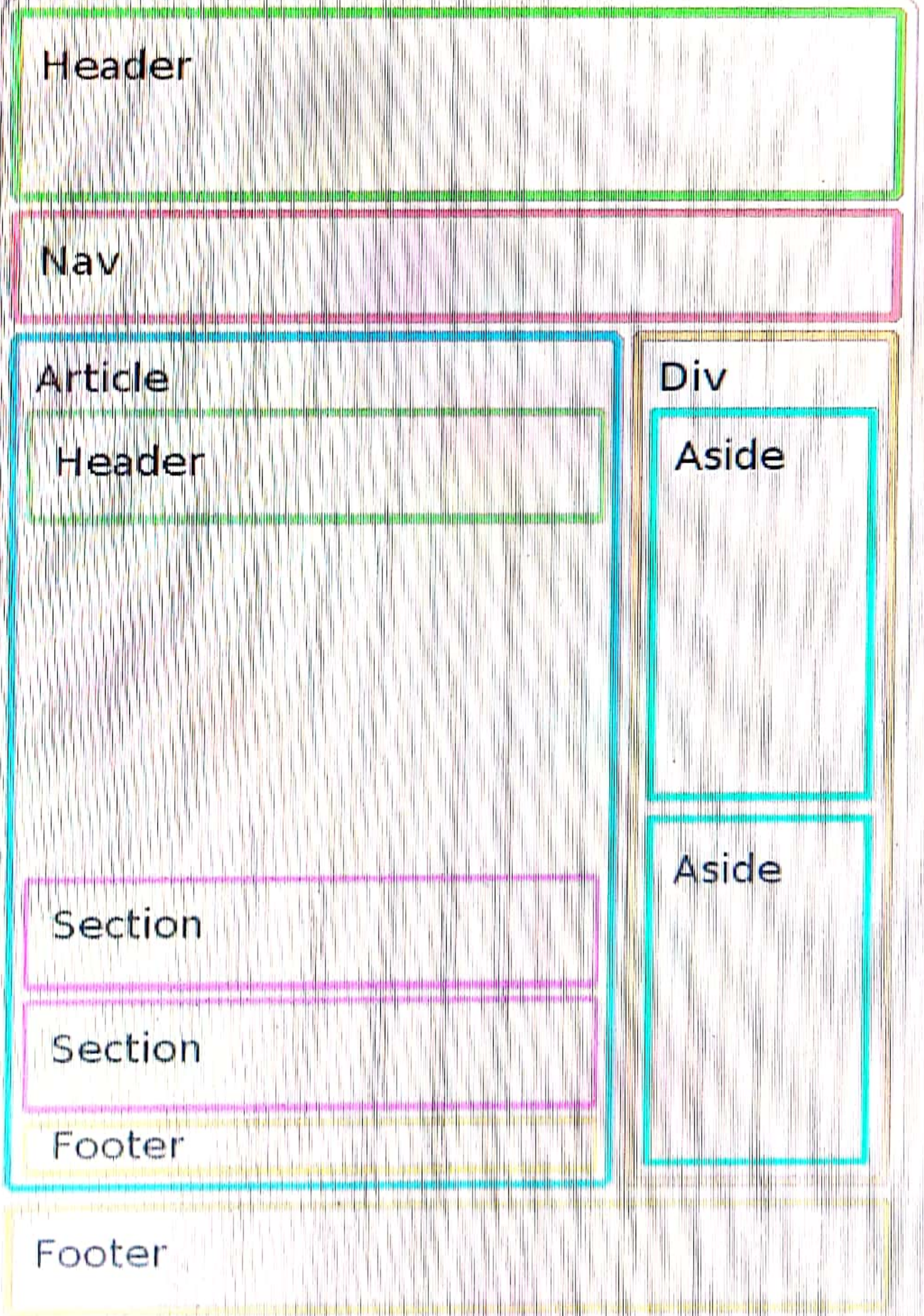


CMS (Content Management Software)

- A Content Management System (CMS) manages the creation and modification of digital content.
- CMS is a software that facilitates creating, editing, organizing, and publishing content.
- It typically supports multiple users in a collaborative environment.

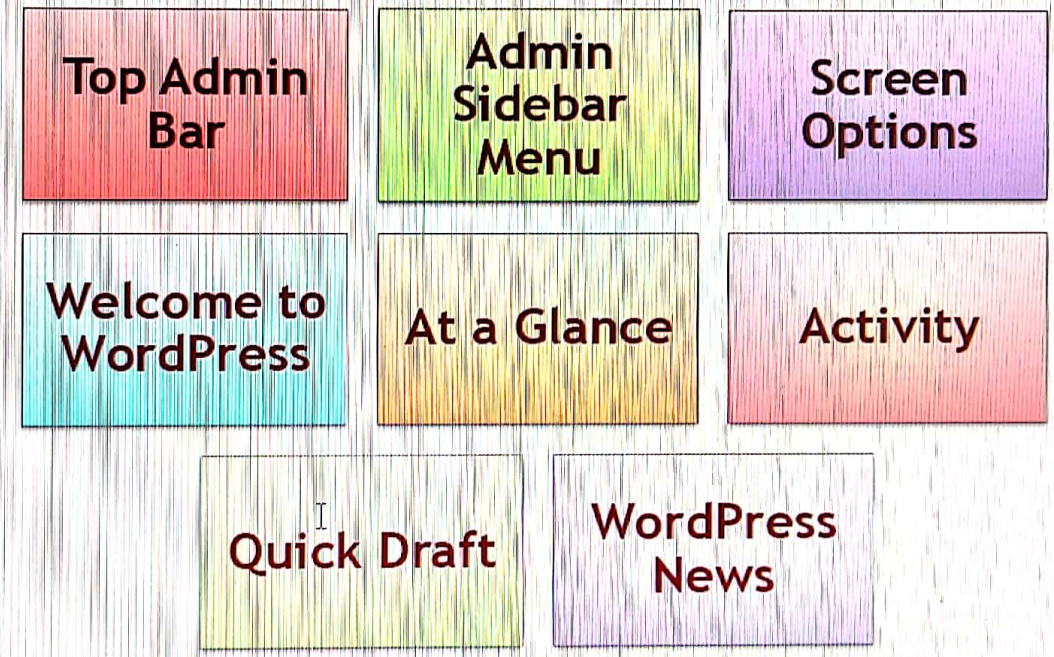
Examples of popular CMSs

- **WordPress**
  - https://wordpress.org
- **Drupal**
  - https://www.drupal.org
- **Joomla**
  - https://www.joomla.org
- **ExpressionEngine**
  - https://expressionengine.com
- **Typo3**
  - https://typo3.org
- **ImpressPages**
  - https://www.impresspages.org

CMS (Content Management Software)

- WordPress (WordPress.org) is a Content Management System (CMS) based on PHP and MySQL that is usually used with the MySQL or MariaDB database servers but can also use the SQLite database engine.
- WordPress is a Content Management System, that allows you to create and publish your content on the web. Although it is mostly used for web publishing, it can be used to manage content on an intranet, or in a single computer.
- WordPress allows users to have full control over the files, documents as well as the design and display of the content

Dashboard Features



Ways to define Styles

- Exter
- Interr
- Inline

External style sheet

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

Internal style sheet

```
<head>
<style type="text/css">
hr {color: sienna}
p {margin-left: 20px}
</style>
</head>
```

Inline style

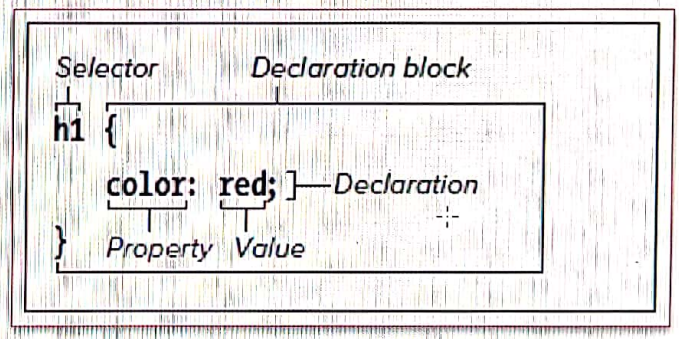
```
<p style="color: sienna; margin-left: 20px"> This is a paragraph </p>
```

Cascading Style Sheet (CSS)

- CSS is a language that describes the style of an HTML document.
- A Style Sheet is simply a text file that contains one or more rules that determine– through properties and values – how certain elements in your Web page should be displayed.
- CSS stands for Cascading Style Sheets.
- Styles are normally stored in Style Sheets.
- External style sheets can save a lot of work.
- External style sheets are stored in CSS files.
- Multiple style definitions will cascade into one.

CSS Style rule

- Each style rule in a style sheet has two main parts: the selector, which determines which elements are affected, and the declaration block, made up of one or more property/value pairs (each constitutes a declaration), which specifies just what should be done.



Selector

- Type selector:
  - Element type, such as body, p, hr, etc.
  - Multiple element types using the same style are separated by comma
    - h1, h2, h3, h4, h5, h6 {background-color:purple}
- ID selector:
  - #p1, #s1 {background-color: blue}
  - …
  - …
  - id values are case-sensitive

CSS Box Model

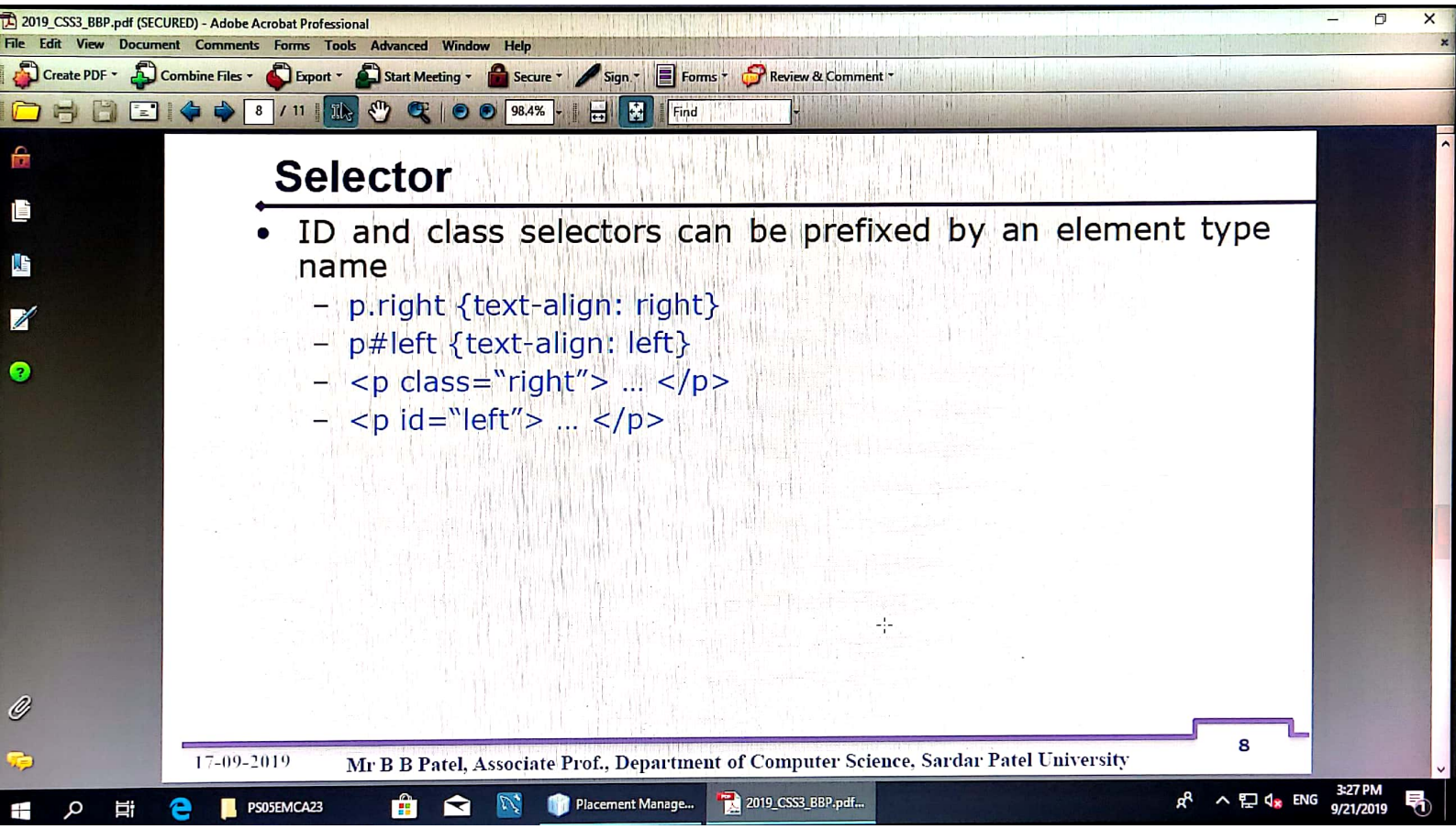
- CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself.
- It is used to create the design and layout of web pages.
- It can be used as a toolkit for customizing the layout of different elements.
- The web browser renders every element as a rectangular box according to the CSS box model.
- The CSS box model is essentially a box that wraps around every HTML element.

Selector

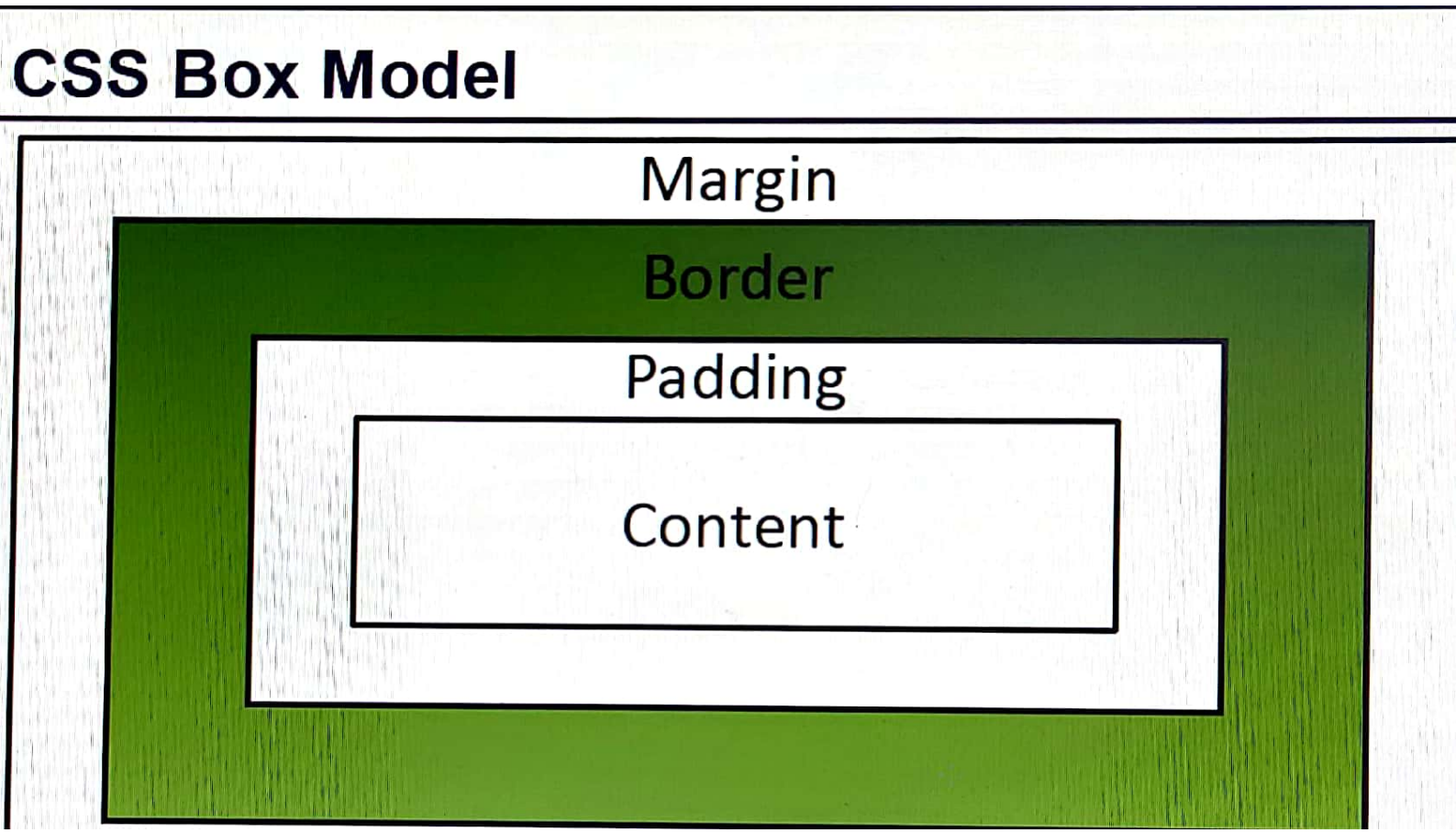
- Class selector:
  - .myitalic {font-style: italic}
  - .myred {color: red}
  - …
  - class values are case sensitive
  - multiple classes can be applied, separated by space
  - All but a few elements, such as html, head, and elements that appear as content of head, have the class attribute

CSS3 Modules

- Box Model
- SelectorsBackgrounds and BordersText
- Effects2D/3D
- TransformationsAnimations
- Multiple Column Layout
- User Interface
- Speech module
- Hyperlink Presentation

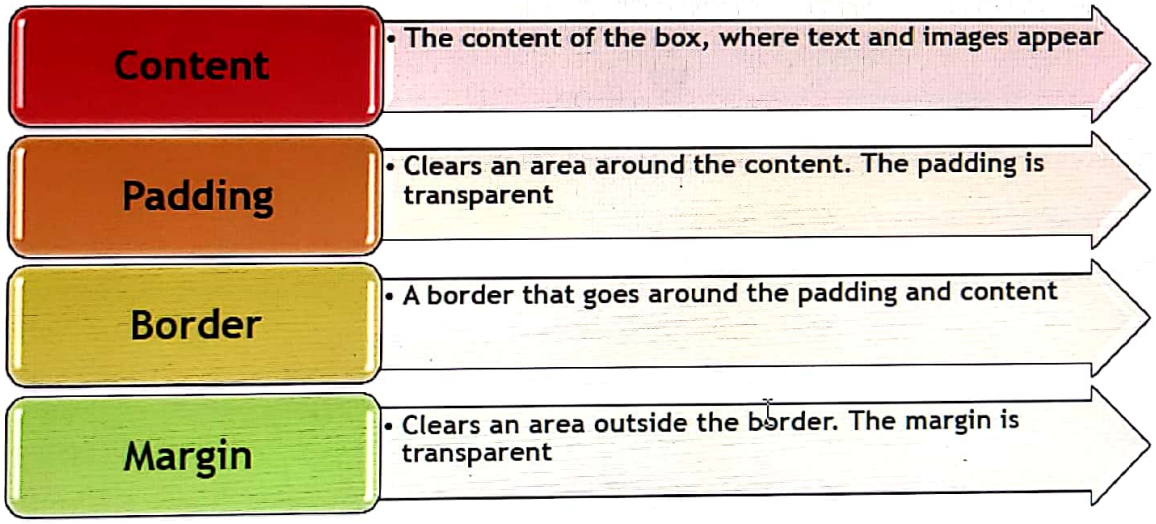


CSS Box Model



CSS Box Model

- Box-Model has multiple properties in CSS. Some of them are given below:



Accessibility

HTML5 makes creating accessible sites easier for two main reasons:

[1] semantics and

[2] ARIA. The new (some currently available) HTML headings like <header>, <footer>, <nav>, <section>, <aside>, etc. allow screen readers to easily access content.

Before, your screen readers had no way to determine what a given was even if you assigned it an ID or Class. With new semantic tags screen readers can better examine the HTML document and create a better experience for those who use them.

ARIA is a W3C spec that is mainly used to assign specific "roles" to elements in an HTML document – essentially creating important landmarks on the page: header, footer, navigation or article, via role attributes. This has been well overlooked and widely under-used mostly due to the fact that it wasn't valid, however, HTML5 will validate these attributes. Also, HTML5 will have built in roles that can't be over-ridden making assigning roles a no brainer.

Font Paragraph

Video and Audio Support

Forget about Flash Player and other third party media players, make your videos and audio truly accessible with the new HTML5 <video> and <audio> tags. Getting your media to play correctly has always been pretty much a nightmare, you had to use the <embed> and <object> tags and assign a huge list of parameters just to get the thing visible and working correctly. Your media tags just become these nasty, huge chunks of confusing code segments. HTML5's video and audio tags basically treat them as images; <video src="url"/>. But what about all those parameters like height, width and autoplay? No worries my good man, just define those attributes in the tag just like any other HTML element: <video src="url" width="640px" height="380px" autoplay/>.

It's actually that dead simple, however because old evil browsers out there don't like our HTML5 friend, you'll need to add a little bit more code to get them working correctly... but this code isn't nearly as gnarly and messy as the <object> and <embed> tags:

```
<video poster="myvideo.jpg" controls>
 <source src="myvideo.m4v" type="video/mp4" />
 <source src="myvideo.ogg" type="video/ogg" />
 <embed src="/to/my/video/player"></embed>
</video>
```

Smarter Storage

One of the coolest things about HTML5 is the new local storage feature. It's a little bit of a cross between regular old cookies and a client-side database. It's better than cookies because it allows for storage across multiple windows, it has better security and performance and data will persist even after the browser is closed. Because it's essentially a client side data base you don't have to worry about the user deleting cookies and it is been adopted by all the popular browsers.

Local storage is great for many things, but it's one of HTML5 tools that are making web apps possible without third party plugins. Being able to store data in the user's browser allows you to easily create those app features like: storing user information, the ability to cache data, and the ability to load the user's previous application state. If you are interested in getting started with local storage, check out Christian Heilmann's great 24 Ways article from last year

Better Interactions

Awe, we all want better interactions, we all want a more dynamic website that responds to the user and allows the user to enjoy/interact your content instead of just look at it. Enter <canvas>, the drawing HTML5 tag that allows you to do most (if not more) interactive and animated possibilities than the previous rich internet application platforms like Flash.

Beyond <canvas>, HTML5 also comes with a slew of great APIs that allow you to build a better user experience and a beefier, more dynamic web application — here's a quick list of native APIs:

- Drag and Drop (DnD)
- Offline storage database
- Browser history management
- document editing
- Timed media playback

Game Development

Doctype

```
<!DOCTYPE html>
```

Yup that's it, that is the doctype, nothing more, nothing less. Pretty simple right? No more cutting and pasting some long unreadable line of code and no more dirty head tags filled with doctype attributes. You can simply and easily type it out and be happy. The really great thing about it though, beyond the simplicity, is that it works in every browser clear back to the dreaded IE6.

Cleaner Code

If you are passionate about simple, elegant, easy to read code then HTML5 is the beast for you. HTML5 allows you to write clear and descriptive code, semantic code that allows you to easily separate meaning from style and content. Consider this typical and simple header code with navigation:

```
<div id="header">
 <h1>Header Text</h1>
 <div id="nav">
  <ul>
   <li><a href="#">Link</a></li>
   <li><a href="#">Link</a></li>
   <li><a href="#">Link</a></li>
  </ul>
 </div>
</div>
```

So this code is pretty clean and simple? But with HTML5 you can clean this up even more and at the same time give your markup more meaning:

```
<header>
 <h1>Header Text</h1>
 <nav>
  <ul>
   <li><a href="#">Link</a></li>
```

It's the Future, Get With It!

The number one reason why you should start using HTML5 today is this: it's the future, start using it now so you don't get left behind. HTML5 is not going anywhere and as more and more elements get adopted more and more companies will start to develop in HTML5. HTML5 is essentially just HTML, it's not scary, it's not anything you really need to figure out or relearn — if you're developing XHTML strict right now you are already developing in HTML5 so why not take full advantage of it's current capability?

You really don't have any excuses not to adopt HTML5 and begin your new love affair with it. Truly, the only real reason I prefer to use HTML5 is just to write cleaner code, all the other benefits and fun features I haven't even really jumped into yet, but that is the great thing about it, you can just start using it right now and not even change the way you design. So, start using it right now, whether you are just simplifying and making your markup more semantic OR you are gonna build some sick new mobile game that will take over the world — who knows, maybe you can start selling stuffed animal versions of your gaming characters too.

2. Getting Simple – No more Type attribute for script and link

This is how you define your <script> and <link> tags in HTML4.

```
1 <link rel="stylesheet" type="text/css" href="style.css" />
2 <script type="text/javascript" src="script.js"></script>
```

In HTML5, you don't have to specify the MIME type value for your <script> and <link> tag. All scripts and styles are assumed to be type="text/javascript" and type="text/css". You can simply write your code as:

```
1<link rel="stylesheet" href="style.css" />
```

1. A new DOCTYPE declaration

Having problem remembering long and complicated HTML4 / XHTML 1.0 doctype like below?

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

If so, then you will surely love the new doctype by HTML5, which looks simple, clean and easy to memorize. Here it is:

```
<!DOCTYPE html>
```

2. Getting Simple – No more Type attribute for script and link

This is how you define your <script> and <link> tags in HTML4.

```
<link rel="stylesheet" type="text/css" href="style.css"
/>
<script type="text/javascript" src="script.js"></script>
```

```
1<link rel="stylesheet" href="style.css" />
2<script src="script.js"></script>
```

3. Semantic Structure – header , footer & nav

Previously, we added an `id` or `classes` to HTML structure and perhaps, it can be serve as pseudo-semantic structure. But by nature, `div` have no semantic structure even after added an `id`.

```
1<div id="header">
2     ...
3</div>
4<div id="nav">
5     <ul>...</ul>
6</div>
7<div id="footer">
8     ...
</div>
```

4. Semantic Structure – article vs section

HTML5 also offers new <article> and <section> tags to help us create semantic content.

```
1<section>
2     ...
3</section>
```

<section> tag defines sections in a HTML such as headers, footers, or any other sections of content.

```
1<article>
2     ...
3</article>
```

The <article> tag is used to specify independent and self-contained content.

Hereby, HTML5 were created to help us better explain the semantic structure. New HTML5 tags like <header>, <nav> and <footer> can be used to replace the mark-up above, with provide semantic structure to content.

```
1<header>
2      ...
```

```
3</header>
4<nav>
5      <ul>...</ul>
6</nav>
7<footer >
8      ...
9</footer>
```

5. New HTML5 Form Input Types and Attributes

HTML5 has introduced 13 new input types and several new attributes for <form> and <input> tags. However, not all browsers are fully support HTML5 new input types and attribute. By the way, you might have to try out these amazing HTML5 form, some new features has been added for instance the browser-based validation, build-in placeholder and new input types.

```
<formid="myform">

    Name: <input name="name" required placeholder="Your
1 name" pattern="[A-z]{7}" />
2   <br/>
3   Email: <input type="email" name="email" required
4 placeholder="email@inwebson.com"/>
5   <br/>
6   URL: <input type="url" name="url"
  placeholder="Homepage URL"/>
7   <br/>
8   Age: <input type="number" name="age" min="18"
9 max="99" />
10  <br/>
11  Description: <textarea name="desc"
  placeholder="Describe yourself here..."></textarea>
```

```
11      Description: <textarea name="desc"
  placeholder="Describe yourself here..."></textarea>
12      <br/>
13      <input type="submit" value="Submit" />
  </form>
```

The form above using new HTML5 input types like *email*, *url* and *number*, and new HTML5 input attributes like *required*, *pattern(regexp)*, *min*, *max* and *placeholder*.

6. HTML5 Canvas

The most exciting feature, HTML5 <canvas> tag allows us to render 2D shapes or graphics on web page with help of JavaScript. Moreover, we're able to create an animation using HTML5 Canvas.
The HTML below shows a simple HTML5 Canvas declaration and use JavaScript

7. HTML5 Audio and Video Support

Previously, to embed an audio or video file on web page we have to rely on third party plugin like Flash through `<object>` and `<embed>` tags. To view the media, user requires administrator privileges to install this plugin as well as the browser has to be able to support this plugin.

```
<object width="400" height="300">
    <param name="movie" value="video.mp4" />
    ...
    <embed src="video.mp4" type="application/x-shockwave-flash" ... ></embed>
</object>
```

HTML5 now introduces a new way to embed media via the `<audio>` and `<video>` tags. With HTML5, at least we don't have to worry about plugins

new HTML5 input attributes like *required*, *pattern(regexp)*, *min*, *max* and *placeholder*.

6. HTML5 Canvas

The most exciting feature, HTML5 <canvas> tag allows us to render 2D shapes or graphics on web page with help of JavaScript. Moreover, we're able to create an animation using HTML5 Canvas.
The HTML below shows a simple HTML5 Canvas declaration and use JavaScript to draw a blue rectangle on it.

```
1<canvas id="myCanvas" width="200" height="200"></canvas>
2
3<script>
4     var c=document.getElementById("myCanvas");
5     var ctx=c.getContext("2d");
6     ctx.fillStyle="#0000FF";
7     ctx.fillRect(0,0,150,150);
8</script>
```

7. HTML5 Audio and Video Support

8. Editable HTML5 Content

HTML5 has offers another cool new attribute – `contenteditable`. You can make your content editable with adding the `contenteditable` attribute to it. This feature will be more useful if paired with HTML5 Local Storage (will be explain in below).

```
1<div contenteditable="true">
2     Any content here will be editable...
3</div>
```

9. HTML5 Local Storage

Now, no more web browser's cookies storage needed (perhaps, depend on your usage)! This is because HTML5 has introduces a new way to store the data in user's browser, known as client-side storage. HTML5 offers two new methods for storing data on user's browser, which are `localStorage` and `sessionStorage`.

```
1<script>
```

deleted when the browser's window is closed.

10. HTML5 Custom Data- Attribute

Have you ever use custom attributes inside a tag to store arbitrary data for the purpose of JavaScript? Or store these arbitrary data using `class` or `rel` attribute rather than creating custom attributes for the purpose of valid HTML markup.

```
1<div id="div1" class="style" time="3" order="1"></div>

2<div id="div1" class="style" time="5" order="3"></div>

3<div id="div1" class="style" time="2" order="2"></div>
```

Thanks to HTML5, we now able to create custom attributes on all HTML elements with the prefix `data-` and simultaneously, give us valid HTML markup.

```
1<div id="div1" class="style" data-time="3" data-
 order="1"></div>
2
 <div id="div1" class="style" data-time="5" data-
```

9. HTML5 Local Storage

Now, no more web browser's cookies storage needed (perhaps, depend on your usage)! This is because HTML5 has introduces a new way to store the data in user's browser, known as client-side storage. HTML5 offers two new methods for storing data on user's browser, which are `localStorage` and `sessionStorage`.

```
1 <script>
2     localStorage.variableName = "value";
3     alert(localStorage.variableName);
4     localStorage.removeItem("variableName");
5     alert(localStorage.variableName);
6 </script>
```

type=" "	Description
text	Defines a one-line text input field
password	Defines a one-line password input field
submit	Defines a submit button to submit the form to server
reset	Defines a reset button to reset all values in the form.
radio	Defines a radio button which allows select one option.
checkbox	Defines checkboxes which allow select multiple options form.
button	Defines a simple push button, which can be programmed to perform a task on an event.
file	Defines to select the file from device storage.
image	Defines a graphical submit button.

HTML5 added new types on <input> element. Following is the list of types of elements of HTML5

type=" "	Description
color	Defines an input field with a specific color.
date	Defines an input field for selection of date.
datetime-local	Defines an input field for entering a date without time zone.
email	Defines an input field for entering an email address.
month	Defines a control with month and year, without time zone.
number	Defines an input field to enter a number.
url	Defines a field for entering URL
week	Defines a field to enter the date with week-year, without time zone.
search	Defines a single line text field for entering a search string.
tel	Defines an input field for entering the telephone number.

HTML5 Tags

- <article>
- <aside>
- <audio>
- <bdi>
- <canvas>
- <datalist>
- <details>
- <dialog>
- <embed>
- <figcaption>
- <figure>
- <footer>
- <header>
- <keygen> (DEPRECATED)
- <main>
- <mark>
- <menuitem>
- <meter>
- <nav>
- <output>
- <progress>
- <rp>
- <rt>
- <ruby>

Structural Elements

article

HTML5 <article> tag defines space for an article. It's way easier to use this element instead of <p> if you need a specific formatting for long text.

Example

```
<article>
  <h1>Fun Fact</h1>
  <p>Fun fact: most of the fun facts on the Internet are not actually fun.</p>
</article>
```

aside

The <aside> tag describes the content which should be set to the side. It can be used as a sidebar or a part of additional text.

Example

```
<aside>
  <h4>Lake</h4>
  <p>Oxford lake is a lake in the state.</p>
</aside>
```

bdi

HTML5 <bdi> feature allows creating mirrored text (or a part of a text). It means that the text will be written not from left to right, but vice versa. It is very useful when writing in languages such as Arabic, or simply adding some whimsicality.

Example

```
<ul>
  <li>Player <bdi>One</bdi>: 66 points</li>
  <li>Player <bdi>Two</bdi>: 86 points</li>
  <li>Player <bdi>Three</bdi>: 69 points</li>
</ul>
```

details

dialog

<dialog> defines the dialog box in your website. However, Microsoft Edge and Internet Explorer **do not** support this tag.

Example

```
<dialog open id="DialogExample">
  <p>Here is some text for example.</p>
</dialog>
```

figcaption

The <figcaption> element defines the caption for <figure> element. This tag goes together with <figure> element.

Example

```
<figure>
  <figcaption>Dog</figcaption>
  <img src="image.png" alt="The Bread Dog" width="300" height="300">
</figure>
```

figure

The new HTML5 <figure> tag sets space for isolated content. It can be photos, diagrams, codes, videos, etc.

Example

```
<figure>
  <figcaption>Logo</figcaption>
  <img src="image.png" alt="Logo" width="300" height="300">
</figure>
```

footer

<footer> is one of the new HTML5 features for describing your content better. This tag sets space for footnotes.

Example

```
<footer>
```

date	Defines an input field for selection of date.
datetime-local	Defines an input field for entering a date without time zone.
email	Defines an input field for entering an email address.
month	Defines a control with month and year, without time zone.
number	Defines an input field to enter a number.
url	Defines a field for entering URL.
week	Defines a field to enter the date with week-year, without time zone.
search	Defines a single line text field for entering a search string.
tel	Defines an input field for entering the telephone number.

- <keygen> (DEPRECATED)
- <main>
- <mark>
- <menuitem>
- <meter>
- <nav>
- <output>
- <progress>
- <rp>
- <rt>
- <ruby>
- <section>
- <source>
- <summary>
- <svg>
- <time>
- <track>
- <video>
- <wbr>

details

HTML5 <details> tag describes the details of your website or content in general. The details can either be visible to everyone or hidden.

Example

```
<details>
  <summary> right</summary>
  <p>- by your company</p>
  <p>Content governed by  right.</p>
</details>
```

header

The <header> tag allows you to set the header - the main title - of your website. It is similar to the title element, but is visible in the website.

Example

```
<header>
  <h1>JavaScript</h1>
  <h3>What is JavaScript?</h3>
  <p>Today we are going to talk about JavaScript.</p>
</header>
```

main

Example

```
<footer>
  <address>
    Postal Address: Door No.00, Street, City, State, Country.
  </address>
  <p> right © 2018 All rights reserved.</p>
</footer>
```

```
<p>Karma    points:    <meter    optimum="30"    high="80"    max="100"
value="85">85</meter></p>
<p>Gas in Tanker: <meter low="20" max="100" value="11">11%</meter></p>
<p>Animals Petted: <meter low="10" high="60" min="0" max="50" value="43"
title="Animals">Petting</meter></p>
<p>Satisfaction:          <meter          max="100"          optimum="100"
value="100">100%</meter></p>
```

nav

The new <nav> element describes a special space for navigation links on your website.

main

<main> depicts space for the main content of a webpage.

Example

```
<main id="content" class="group" role="main">
```

mark

The <mark> element highlights the text (or part of it) on your site.

Example

```
<p>The mark tag is <mark>useful</mark> when you need to highlight important
information.</p>
```

menuitem

The <menuitem> element creates a menu item, which can be accessed through a pop-up menu on your website.

Example

```
<img    src="image.png"    id="doggo"    width="250"    alt="Space    Doggo"
contextmenu="sunmenu">

<menu type="context" id="sunmenu">
  <menuitem label="Zoom In" icon="Myzoom-in.png" onclick="zoomin()">
  <menuitem label="Zoom Out" icon="Myzoom-out.png" onclick="zoomout()">
  <menuitem          label="Reload          Image"          icon="Myreload.png"
onclick="window.location.reload();">
</menu>
```

meter

<meter> element depicts a scalar measurement in a defined range on your website.

Karma points:

Gas in Tanker:

Example

```
<nav>
  <ul>
    <li><a          href="https://www.bitdegree.org/tag/gamified/">Gamified
Courses</a></li>
    <li><a href="https://www.bitdegree.org/tutorials/">Tutorials</a></li>
    <li><a      href="https://www.bitdegree.org/course/learn-solidity-space-
doggos/">Space doggo courses</a></li>
    <li><a      href="https://www.bitdegree.org/tag/game-dev/">Game      Dev
Courses</a></li>
  </ul>
</nav>
```

progress

<progress> defines a progress bar for a task, which is included on your web page.

Example

```
<progress value="60" max="100"></progress>
```

rp

<rp> tag describes what is shown when the browser does not support <ruby>element included in your website.

Example

```
<ruby>
   吹禁
  <rp> ( </rp>
  <rt>こうかく</rt>
  <rp> ) </rp>
</ruby>
```

rt

The <rt> element describes the explanation for Asian typography so your web page can display Asian symbols correctly.

Example

```
ruby
```

section

<section> is used to set a part of the content for one section element.

Example

```
<section>
  <h1>Section Heading</h1>
  <p>The section tag can contain any elements.</p>
  <img src="image.png" alt="section example">
</section>
```

summary

The <summary> element is used for defining a title for the <details> element on your web page.

Some details

Example

```
<details>
  <summary>Some details</summary>
  <p>Provide more info about the details here.</p>
</details>
```

time

<time> is one of the HTML5 new tags for defining time and date on your website.

Example

```
<h2>The premiere show starts at <time>21:00</time> today.</h2>
```

```
  王 <rt>ji</rt>
</ruby>
```

ruby

The <ruby> element is used to define annotation for Asian typography so your website can display Asian symbols correctly.

Example

```
<ruby>攻殻
  <rp> ( </rp>
  <rt>こうかく</rt>
  <rp> ) </rp>
  機動隊
  <rp> ( </rp>
  <rt>きどうたい</rt>
  <rp> ) </rp>
</ruby>
```

section

<section> is used to set a part of the content for one section element.

Example

```
<section>
  <h1>Section Heading</h1>
  <p>The section tag can contain any elements.</p>
  <img src="image.png" alt="section example">
</section>
```

summary

The <summary> element is used for defining a title for the <details> element on your web page.

Some details

wbr

<wbr> is one of the HTML5 new features, which creates a space for a possible line break when a website is watched on different-sized screens.

Example

```
<p>To learn programming, you must be <wbr>Motivated</wbr> and ready.</p>
```

Graphics

canvas

<canvas> is one of the new HTML5 elements, which separates some specific space for JavaScript graphics that you may want to include on your site.

Example

```
<canvas id="canvas01" width="400" height="300" style="border: 1px solid #eee;"></canvas>
```

svg

<svg> defines a specific space for SVG graphic drawing on your web page.

Example

```
<svg width="200" height="200">
  <rect width="200" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />
</svg>
```

Media

audio

<audio> sets a specific area for including audio content into your website.

Example

```
<audio controls>
  <source src="audio-tag-example.mp3" type="audio/mpeg">
  <source src="audio-tag-sample.wav" type="audio/wav">
  <source src="audio-tag-demo.ogg" type="audio/ogg">
  Audio tag is not supported in this browser.
</audio>
```

embed

<embed> depicts a specific space for external plug-ins on your website.

Example

```
<embed src="https://www.youtube.com/embed/10r9ozshGVE">
```

source

The <source> element defines a source for video or audio elements.

Example

```
<audio controls>
  <source src="melody.mp3" type="audio/mpeg">
  <source src="melody.ogg" type="audio/ogg">
  This text will be displayed if the browser does not support the audio element.
</audio>
```

track

The <track> tag sets an audio track for <video> or <audio> elements on your website.

track

The <track> tag sets an audio track for <video> or <audio> elements on your website.

Example

```
<video controls width="325" height="245">
  <source src="random_video.mp4" type="video/mp4">
  <source src="track_video.mp4" type="video/mp4">
  <track   src="english_subtitles.vtt"   kind="subtitles"   srclang="en"
label="English">
  <track  src="lithuanian_subtitles_lt.vtt"  kind="subtitles"  srclang="lt"
label="Lithuanian">
</video>
```

video

The new HTML5 <video> tag sets a specific place for adding video material to your website to make your content more interesting and dynamic.

Example

```
<video controls width="400" height="300">
  <source src="video-tag-example.mp4" type="video/mp4">
  <source src="video-tag-sample.webm" type="video/webm">
  <source src="video-tag-demo.ogg" type="video/ogg">
  Video tag is not supported in this browser.
</video>
```