KodeGo

Intro to Web Development

What is HTML?

HyperText Markup Language

HTML describes the structure of the pages. It uses elements and tags.

It is the most basic building block of the Web.

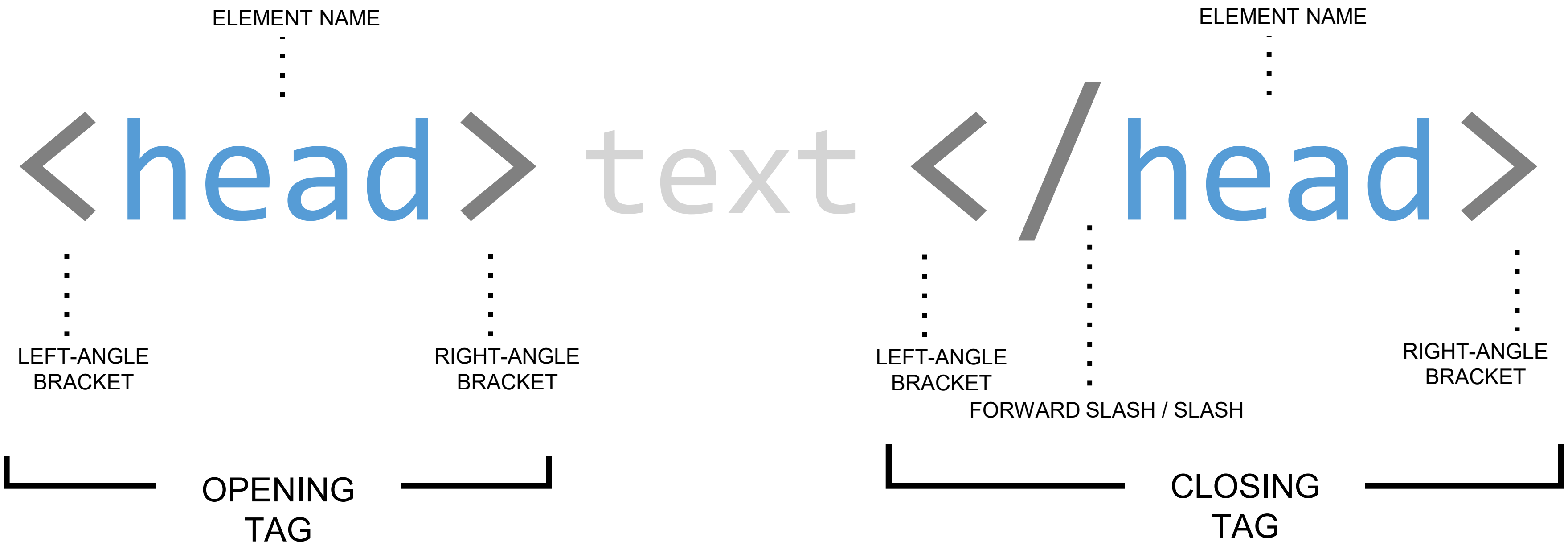
HTML Elements

Elements usually have opening and closing tags that surround and give meaning to the content.

Examples

- Body
- Head
- Title
- Meta

HTML Syntax



<head> text </head>

ELEMENT NAME

LEFT-ANGLE BRACKET

RIGHT-ANGLE BRACKET

OPENING TAG

ELEMENT NAME

LEFT-ANGLE BRACKET

FORWARD SLASH / SLASH

RIGHT-ANGLE BRACKET

CLOSING TAG

Kinds of Tags

Container Tags

always wrap around text or graphics and comes in a set with an opening and closing tag.

<title> My First Webpage </title>

Empty Tags

do not have to be wrapped around text and do not require a closing tag; stand alone

<meta>

Basic HTML Tags

Tag Name	Description
<!DOCTYPE html>	meant to act as links to a set of rules that the HTML page had to follow
<html></html>	the parent tag or root element of a webpage
<head></head>	first child of html tag. Site page information for web browsers and Search engines.
<title></title>	displayed in browsers tab, used only once inside head tag
<meta>	used to define the charset family, description, keywords, Author, robots and Geo Location of a website
<body></body>	used to create the page structure or content; structure includes Headings, Paragraphs, images, tables, division, etc of the website

HTML Structure

```html
<!DOCTYPE html>
<html>
<head>
    <title>My First HTML Website</title>
    <meta charset="UTF-8">
</head>
<body>
    Welcome to my first HTML Website!
</body>
</html>
```

Tags

Doctype
HTML Tag
Head Tag
Title Tag
Meta Tag
Body Tag

Basic HTML Tags

Tag Name	Use
<h1> </h1>	A section heading level 1. Headings are up to <h6> </h6>
<p> </p>	A paragraph tag
<br>	Line break
<hr>	Horizontal Rule
<b> </b>	Bold text
<!-- -->	Insert comment in the source code. A short description of code. Not displayed in the browser.

Nesting Elements

Elements can be placed within other elements.

Example

```
<p>My name is <strong>Raymart</strong></p>

 <p>My name is <strong>Raymart</p></strong>
```

Activity Time

Click the link provided in the chat box.

HTML List Tags

Unordered List

Non sequential list. List with bullets. In HTML5, type attribute of unordered list is deprecated.

UL Example 1: Bullet List

```
<ul>
    <li>Coffee</li>
    <li>Milk</li>
    <li>Tea</li>
</ul>
```

- Coffee
- Milk
- Tea

HTML List Tags

Ordered List

Sequential list. Use numbers, alphabets and Roman characters as list style.

OL Example 1: Uppercase Alphabet List

```
<ol type=”A”>
    <li>Coffee</li>
    <li>Milk</li>
    <li>Tea</li>
</ol>
```

1. Coffee
2. Milk
3. Tea

HTML List Tags

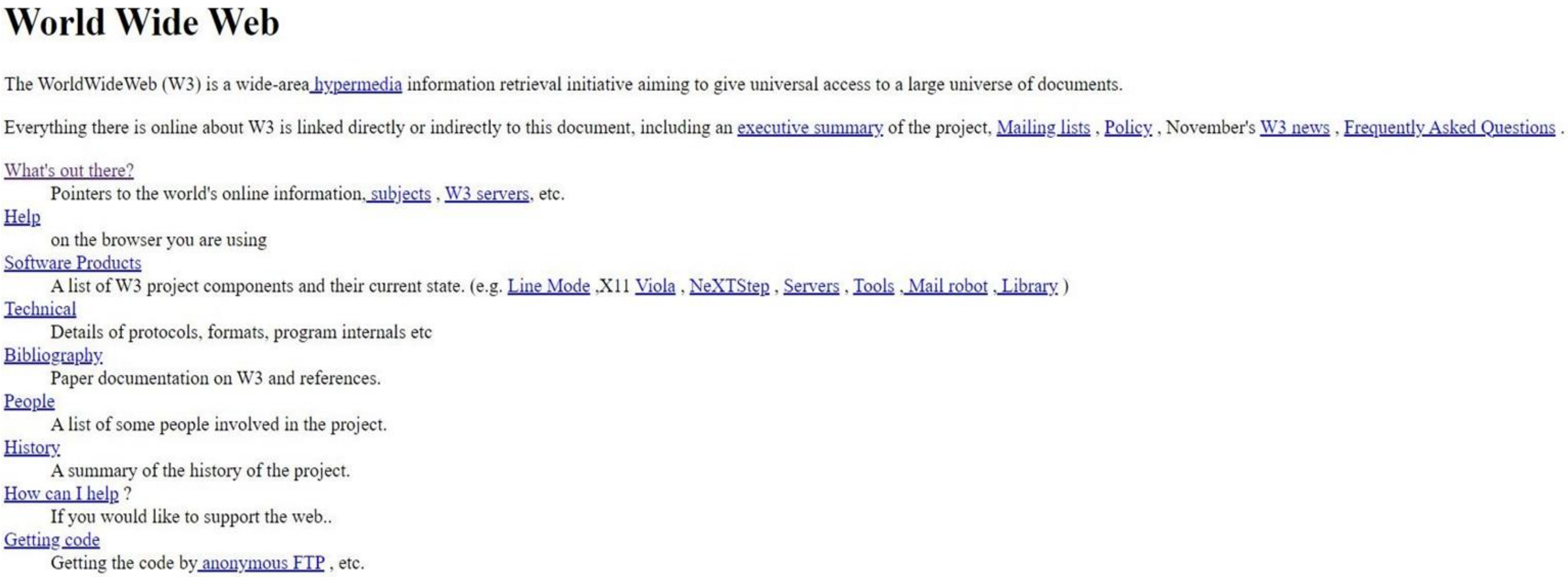
Description List

List with description term and description data.

Syntax:

```
<dl>
    <dt>Description Term</dt>
    <dd>Description Data</dd>
</dl>
```

The World’s First Website



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
    Pointers to the world's online information, subjects , W3 servers, etc.
Help
    on the browser you are using
Software Products
    A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
    Details of protocols, formats, program internals etc
Bibliography
    Paper documentation on W3 and references.
People
    A list of some people involved in the project.
History
    A summary of the history of the project.
How can I help ?
    If you would like to support the web..
Getting code
    Getting the code by anonymous FTP , etc.

Source: http://info.cern.ch/hypertext/WWW/TheProject.html

Exercise 2: Re-create The World’s First Website

1. Create a new html file name
2. Apply what you have learned today, about HTML tags and attributes.
3. Save page as B15_EXERCISE2_LASTNAME.html file on your system.
4. Double click the file and your page is live on browser.
5. Present your version of World’s First Website!

HTML Reference

https://www.w3schools.com/html/default.asp

https://developer.mozilla.org/en-US/docs/Learn/HTML

Learning Outcomes

What have you learned today?

Technical Skills

1. Introduction to Web Development
2. Introduction to Web Stacks
3. HTML Tags
4. HTML Attributes

Soft Skills

1. Communication Skill
2. Research
3. Problem-solving

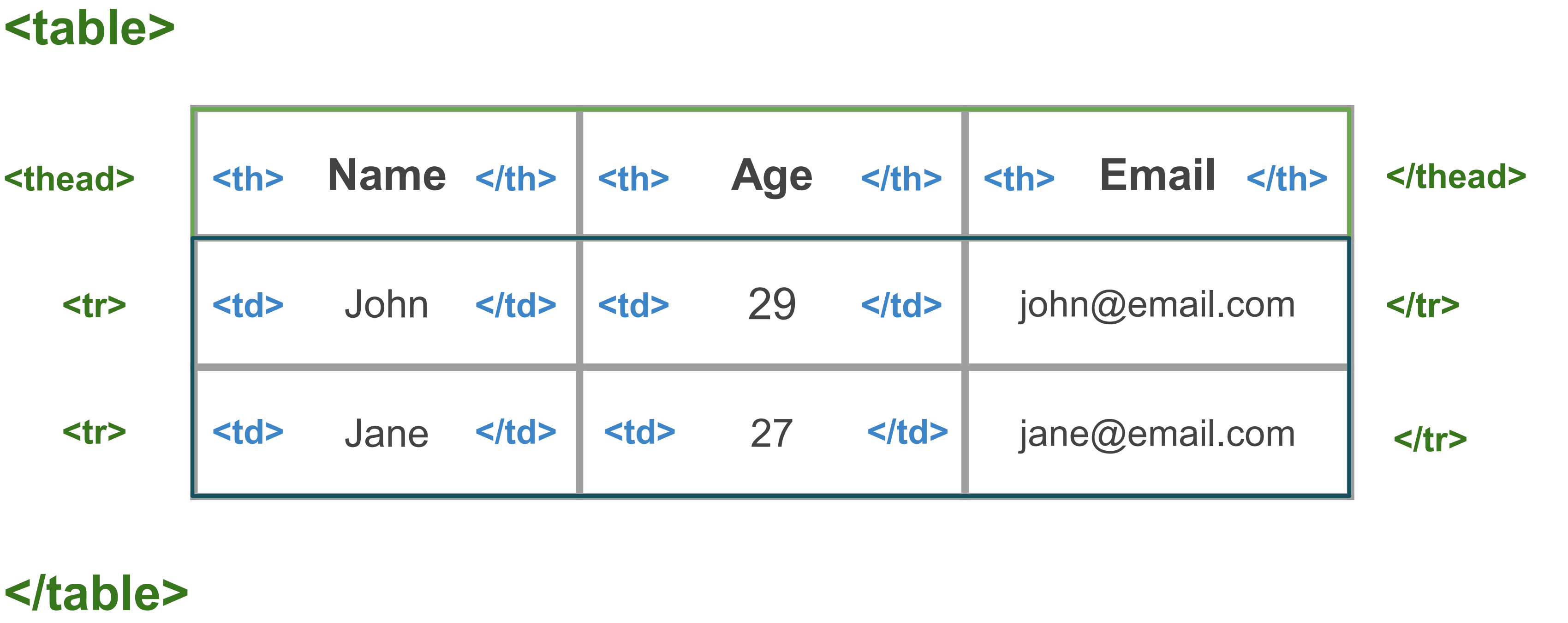
HTML Tables

HTML Tables are used to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

Note: To add borders to the table you can use the **border** attribute.

To define the width of the table you can also use the **width** attribute.

HTML Tables



<table>

<thead>	<th> **Name** </th>	<th> **Age** </th>	<th> **Email** </th>	</thead>
<tr>	<td> John </td>	<td> 29 </td>	john@email.com	</tr>
<tr>	<td> Jane </td>	<td> 27 </td>	jane@email.com	</tr>

</table>

KodeGo

HTML Table Head

Table Head is used to group the **header content** of an HTML table.

Table head is uses the tag:
<thead> </thead>

HTML Table Row

Table Row is used to define a **row** inside the table. The table row can contain **1** or more **<th>** or **<td>**.

Table Row is uses the tag:
<tr> </tr>

HTML Table Data

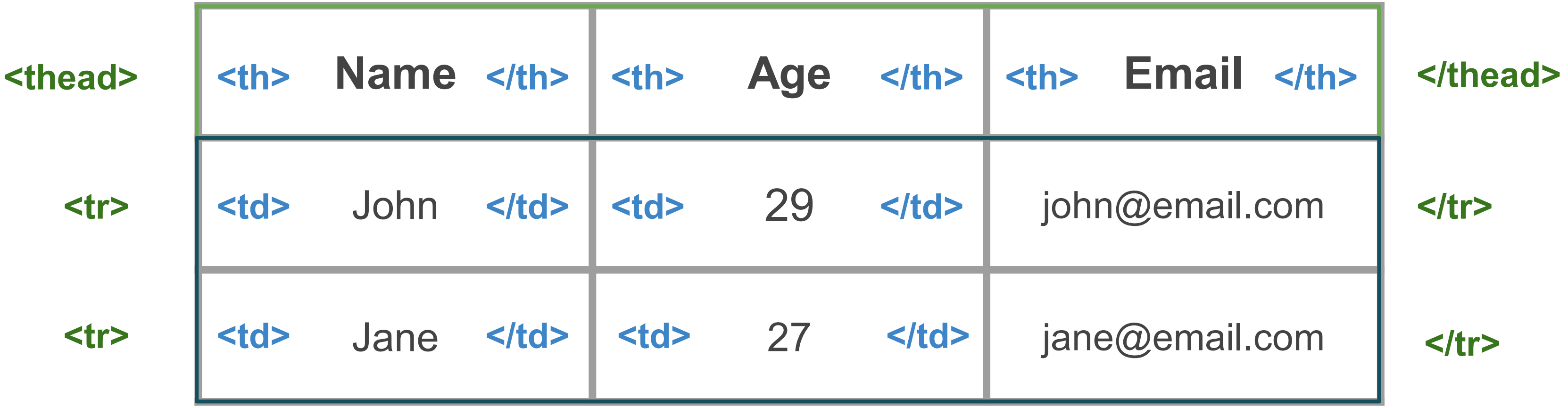
Table Data is used to define a **data cell** inside the table.

Table Data is uses the tag:
<td> </td>

<th>Name</th>	<th>Age</th>	<th>Email</th>
<td>John</td>	<td>29</td>	<td>john@email.com</td>
<td>Jane</td>	<td>27</td>	<td>jane@email.com</td>

HTML Tables

<table>



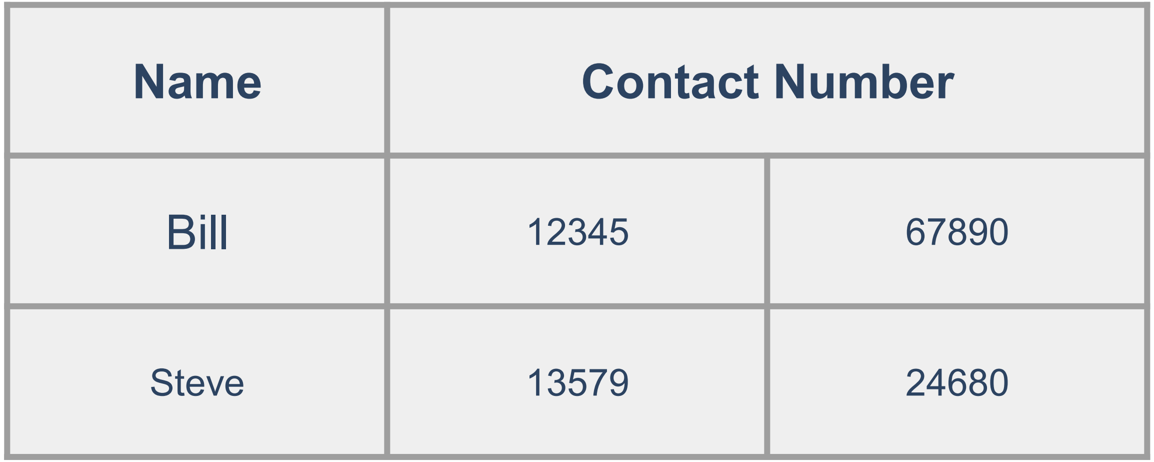
<thead>	<th> **Name** </th>	<th> **Age** </th>	<th> **Email** </th>	</thead>
<tr>	<td> John </td>	<td> 29 </td>	john@email.com	</tr>
<tr>	<td> Jane </td>	<td> 27 </td>	jane@email.com	</tr>

</table>

HTML Table Colspan

Colspan attribute is used to make a cell span for **more than 1 column**.

Syntax:
<td colspan=”2”>



Name	Contact Number	
Bill	12345	67890
Steve	13579	24680

HTML Table Rowspan

Rowspan attribute is used to make a cell span for **more than 1 row**.

Syntax:
<td rowspan=”2”>

NAME	Bill	Steve
Contact Number	12345	13579
	67890	24680

HTML Table Forms

HTML Forms are used to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.

HTML Structure

```
<form action="#" method="#">
              <label for="input-name">Full Name</label>
              <input type="text" name="full_name" id="input-name">
              <!-- other form elements here -->
</form>
```

HTML Forms

The form structure consists of an opening and closing tag of **<form>**. This is to declare that you are creating an **HTML Form**.

The **action** form attribute defines **where** should the form-data go after it is submitted.

While the **method** form attribute defines **how** the form-data is passed

HTML Forms

Labelling form controls are needed to make the form more **user-friendly**. As it shows the user what should be filled up inside the input field.

To label form controls:

```
<label for="input-id">Label Name</label>
```

Note: The **for attribute** should be **equal to the id of the input field**.

HTML Form Controls

Form Controls are used to ‘control’ the different types of data or values inputted by the user.

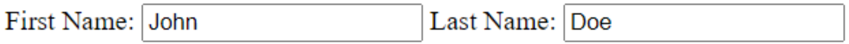
- Text Input
- Password Input
- Date
- Email
- Text Area
- Radio Button
- Checkbox
- Dropdown List Box
- Multiple Select Box
- File Input Box
- Submit Button
- Image Button
- Button
- Form Validation
- Placeholder

HTML Text Input

Syntax

```
<label for="first-name">First Name</label>
<input type="text" name="first-name"
id="first-name">

<label for="last-name">Last Name</label>
<input type="text" name="last-name" id="flast-
name">
```

Web Page



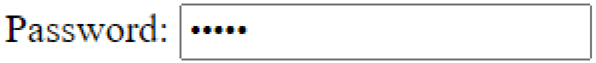
First Name: John Last Name: Doe

HTML Password Input

Syntax

```html
<label for="password">Password</label>
<input type="password" name="user_password"
id="password">
```

Web Page



Password: •••••

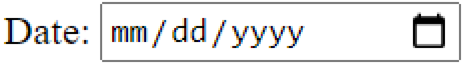
KodeGo

HTML Date Input

Syntax

```html
<label for="date">Date</label>
<input type="date" name="date" id="date">
```

Web Page



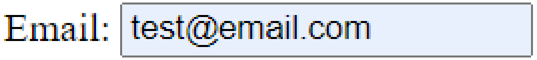
Date: mm / dd / yyyy

HTML Email Input

Syntax

```html
<label for="email">Email</label>
<input type="email" name="email" id="email">
```

Web Page



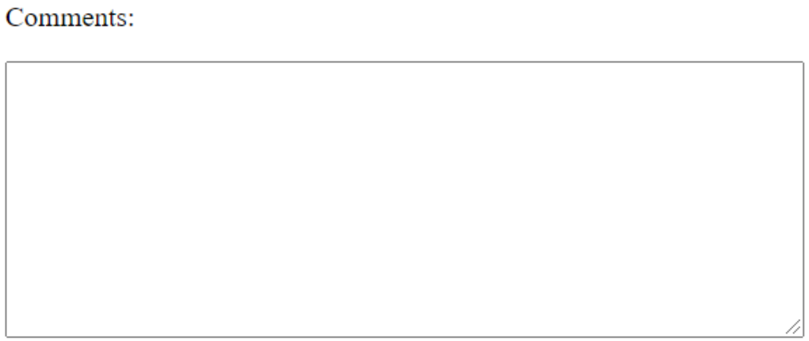
Email: test@email.com

HTML Text Area

Syntax

```
<textarea name="comments" id="comments"
cols="60" rows="10"></textarea>
```

Web Page



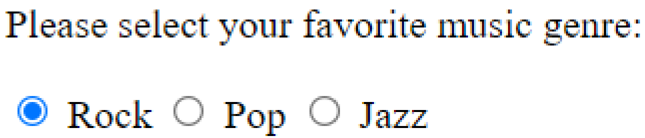
Comments:

HTML Radio Button

Syntax

```
<p>Please select your music genre:</p>
<input type="radio" name=“genre" id="rock"
value="rock">
<label for="rock">Rock</label>
<input type="radio" name=“genre" id=“pop"
value="pop">
<label for="rock">Rock</label>
<input type="radio" name=“genre" id=“jazz"
value="jazz">
<label for="rock">Rock</label>
```

Web Page



Please select your favorite music genre:

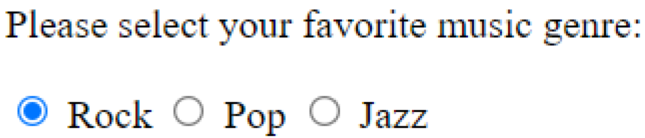
● Rock ○ Pop ○ Jazz

HTML Checkbox

Syntax

```
<p>Please select your favorite food:</p>
<input type="checkbox" name="food"
id="ramen" value="ramen">
<label for="ramen">Ramen</label>
<input type="checkbox" name="food"
id="sushi" value="sushi">
<label for="sushi">Sushi</label>
<input type="checkbox" name="food"
id="curry" value="curry">
<label for="curry">Curry</label>
```

Web Page



Please select your favorite music genre:

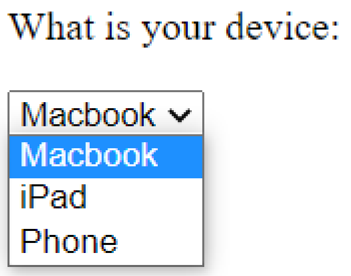
● Rock ○ Pop ○ Jazz

HTML Dropdown List

Syntax

```
<p>What is your device?</p>
<select name="device" id="device">
           <option
value="macbook">Macbook</option>
           <option value="ipad">iPad</option>
           <option
value="iphone">iPhone</option>
</select>
```

Web Page



What is your device:

Macbook ⌄

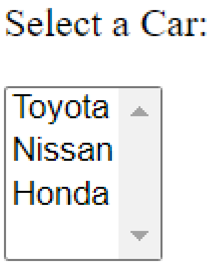
- Macbook
- iPad
- Phone

HTML Select Box

Syntax

```
<p>Select a Car:</p>
<select name="car" id="car" multiple>
           <option
value="toyota">Toyota</option>
           <option
value="nissan">Nissan</option>
           <option
value="honda">Honda</option>
</select>
```

Web Page



Select a Car:

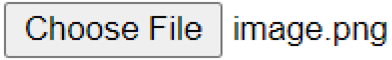
Toyota
Nissan
Honda

HTML Input Box

Syntax

```
<input type="file" name="file" id="file">
```

Web Page

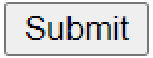


HTML Submit Button

Syntax

```
<input type="submit" name="submit" id="submit"
value="Submit">
```

Web Page

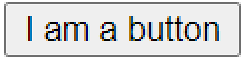


HTML Button

Syntax

```html
<button>I am a button</button>
```

Web Page



HTML Form Validation

HTML5 has the ability to validate most user data once submitted.

This is done by using validation attributes in the form elements.

- **required** = checks whether the input field is filled in or not before being submitted.
- **min** and **max** = limits the minimum or maximum value of the numerical type input fields.

HTML Form Validation

placeholder attribute provides a **small hint or description** for the users on what they will put in the input field.

Types of CSS

Inline CSS

uses the **style** attribute inside the HTML Elements.

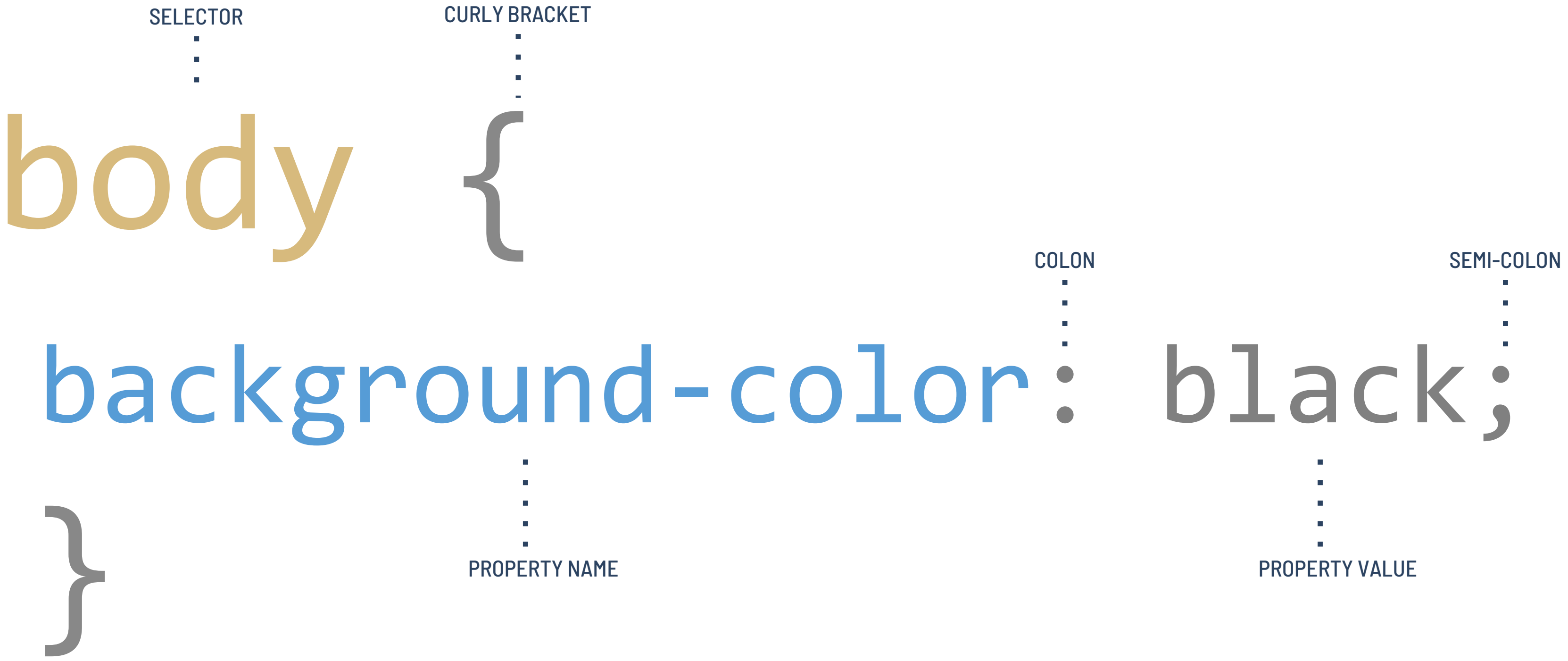
Internal CSS

uses the <**style**> element inside the <**head**> of the HTML page.

External CSS

uses an external CSS file which is linked by the <**link**> element inside the <**head**> of the HTML page.

CSS Syntax



```
body {
 background-color: black;
}
```

SELECTOR

CURLY BRACKET

COLON

SEMI-COLON

PROPERTY NAME

PROPERTY VALUE

Universal Selector

CSS

```css
* {
    color: blue;
}
```

NOTE:
THIS IS CALLED A UNIVERSAL SELECTOR.
YOU CAN USE THIS TO ADD STYLES TO ALL ELEMENTS

CSS SELECTORS - Element

HTML

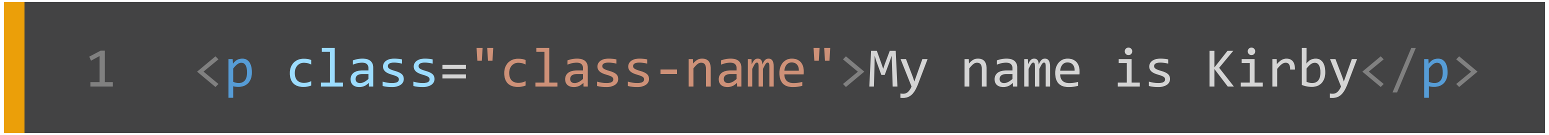
```html
<p>My name is Kirby</p>
```

CSS

```css
p {
	color: blue;
}
```

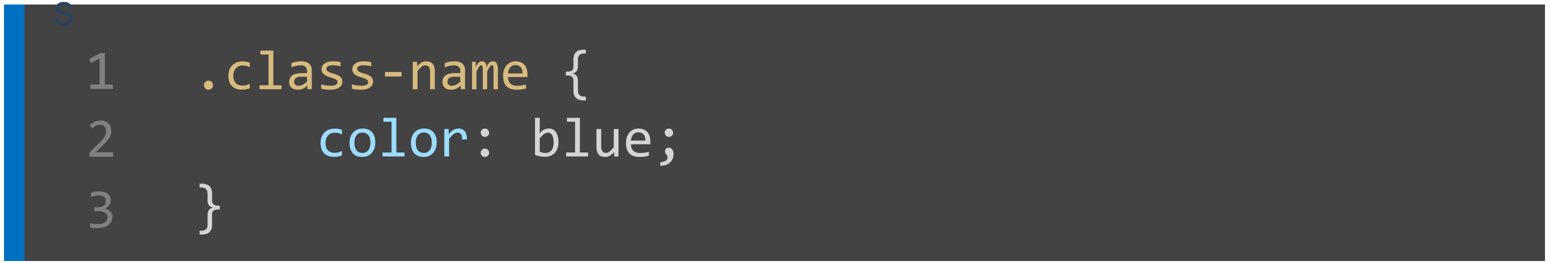
CSS SELECTORS - Class

HTML



```html
<p class="class-name">My name is Kirby</p>
```

CSS



```css
.class-name {
    color: blue;
}
```

CSS SELECTORS - ID

HTML

```html
<p id="id-name">My name is Kirby</p>
```

CSS

```css
#id-name {
	color: blue;
}
```

KodeGo

Class vs ID

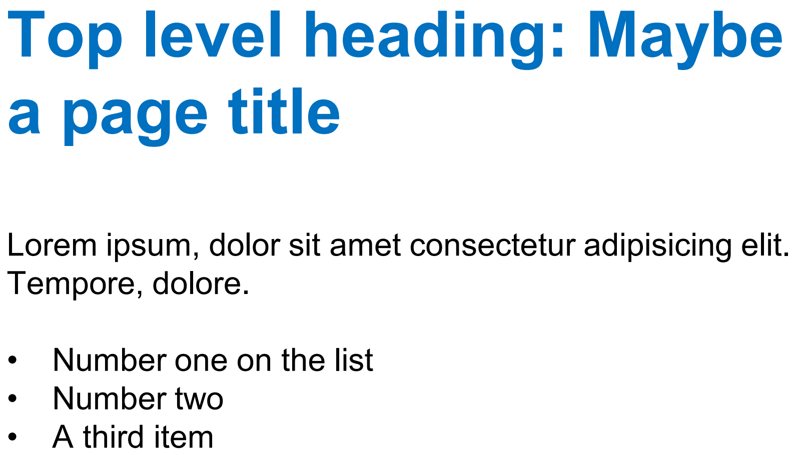
class attribute is used to identify more than one element.

id attribute is used to identify one unique element.

If the two selectors are identical, the latter of the two will take precedence.

Here you can see the **second selector** takes precedence over the first.

```
h1 {
    color: red;
}

h1 {
    color: blue;
}
```



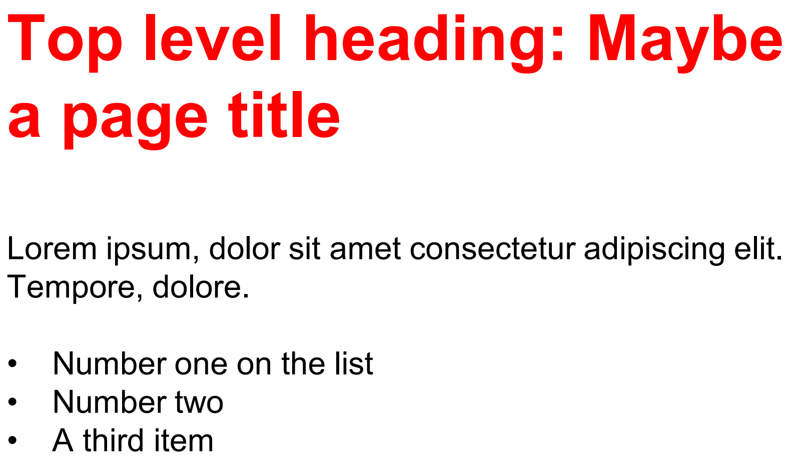
Top level heading: Maybe a page title

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Tempore, dolore.

- Number one on the list
- Number two
- A third item

If one selector is more specific than the others, the more specific rule will take precedence over more general ones.

```
h1.header1 {
    color: red;
}


h1 {
    color: blue;
}
```



Top level heading: Maybe a page title

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Tempore, dolore.

- Number one on the list
- Number two
- A third item

Select all **<li>** child of parent **<ul>**

HTML

```html
<ul class="nav-links">
   <li><a href="#">Home</a></li>
   <li><a href="#">Menu</a></li>
   <li><a href="#">Blog</a></li>
</ul>
```

CSS

```css
ul li  {
     color: blue;
}
```

Select all **<li>** child of parent **<ul>**

HTML

```html
<ul class="nav-links">
   <li><a href="#">Home</a></li>
   <li><a href="#">Menu</a></li>
   <li><a href="#">Blog</a></li>
</ul>
```

CS

```css
ul li  {
    color: blue;
}
```

OR using a parent class

```css
.nav-links li  {
    color: blue;
}
```

CSS Color

SALMON

rgb(250, 128, 114)

opacity: 1

rgba(250, 128, 114, 1)

#fa8072

color

the **color** property defines color of the text.

Example:

```
color: black;
```

CSS Colors

color name

all modern browsers support around **140 color name values.** However, this is not the practical way to use CSS colors

rgb

RGB color value specifies the **red, green, or blue intensity.** The intensity value should only be from **0 to 255.**

hexcode

hexcode is specified with:

#RRGGBB which means **RR for red value, GG for green value**, and **BB for blue value**. These hexadecimal values specifies the colors. All values should be in between **00 and FF**.

Exercise 3: Philippine Tourist Spots

1. Create a travel blog website that will help promote Philippine tourism.
2. Select one (1) city in the Philippines that you want to cover.
3. The blog should include the following:
    - Pictures of the location (tourist spots, historical places etc.)
    - Discuss the culture of the city (food, festival, rituals, transportation etc.)
    - Things to bring
    - Sample itinerary
    - Tips in visiting the city
    - Hyperlinks to hotels, social media sites and other related articles.
    - Comment section
4. Enhance the webpage using CSS formatting and other styles.
5. Insert the screenshots of your codes and sample web page output below.
6. Save your files following the format:

a. WD11_EXERCISE3_LastName_Firstname (docx or pdf format)
b. WD11_EXERCISE3_LastName_Firstname.html
c. WD11_EXERCISE3_LastName_Firstname.css

7. Upload your Exercise 3 files to Github pages.
8. Submit the github page link to the canvas

CSS Background Color

background-color

the **background-color** property defines color of the background of an element.

Example:

```
background-color: rgb(76, 175, 80)
```

Background Image

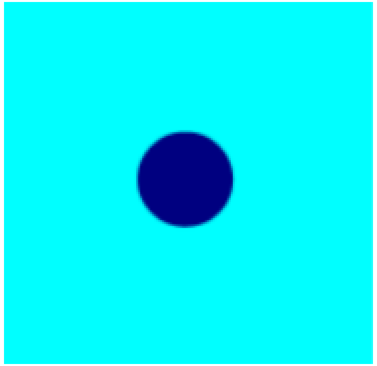
ADDING A BACKGROUND IMAGE

```
background-image: url(images/image.jpg);
```

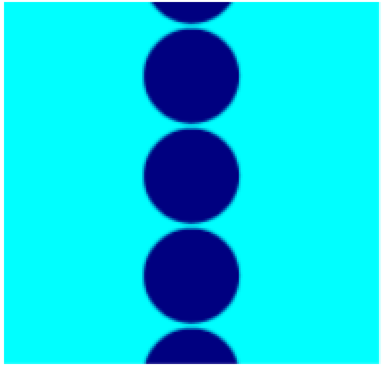
Background Repeat

```
background-repeat:
```

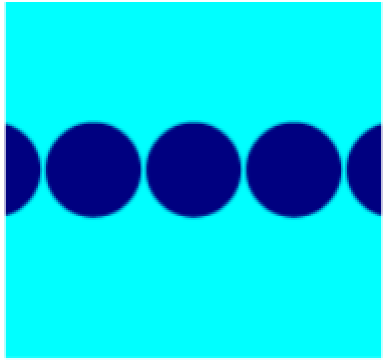
`no-repeat;`



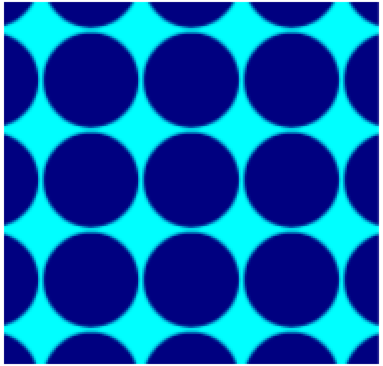
`repeat-y;`



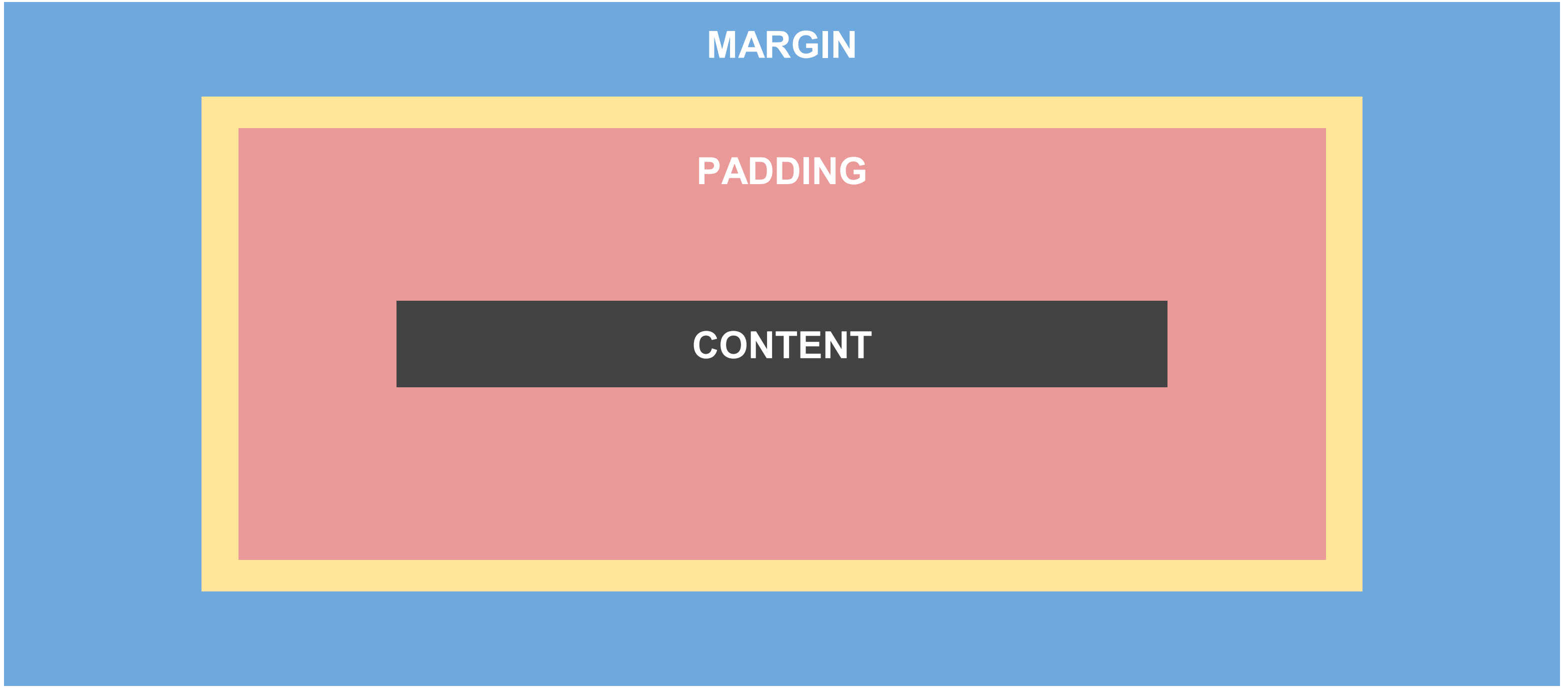
`repeat-x;`



`repeat;`



BOX Model



BORDER
is the yellow box.

Margin

margin

the margin property sets the margin for an element.

Note:
Margin values are applied clockwise

Margin

1 - value margin:

-the value applies the margin to all four sides.

Example:
margin: 10px;

2 - value margin:

- first value = top and bottom side
- second value = right and left side

Example:
margin: 10px 20px;

Margin

3 - value margin:

- **first** value = **top** side
- **second** value = **right** and **left** side
- **third** value = **bottom** side

Example:
margin: 10px 20px 30px;

4 - value margin:

- **first** value = **top** side
- **second** value = **right** side
- **third** value = **bottom** side
- **fourth** value = **left** side

Example:
margin: 10px 20px 30px 40px;

Margin

Note:

margin property is a shorthand.

To apply margin on individual sides you can use the following properties:

- margin-top
- margin-right
- margin-bottom
- margin-left

Padding

padding

the **padding** property sets the padding for an element.

Note:
Padding values are applied clockwise

Padding

1 - value padding:

-the value applies the padding to all four sides.

Example:
padding: 10px;

2 - value padding:

- **first** value = **top** and **bottom** side
- **second** value = **right** and **left** side

Example:
padding: 10px 20px;

Padding

3 - value padding:

- **first** value = **top** side
- **second** value = **right** and **left** side
- **third** value = **bottom** side

Example:
padding: 10px 20px 30px;

4 - value padding:

- **first** value = **top** side
- **second** value = right side
- **third** value = **bottom** side
- **fourth** value = **left** side

Example:
padding: 10px 20px 30px 40px;

Padding

Note:

padding property is a shorthand.

To apply padding on individual sides you can use the following properties:

- padding-top
- padding-right
- padding-bottom
- padding-left

Border

border

the **border** property adds a border to the element.

Border

border shorthand

```
border: border-width border-style border-color;
```

Padding

border-width:

sets the width of the border.

The value of the border-width can be: **px**

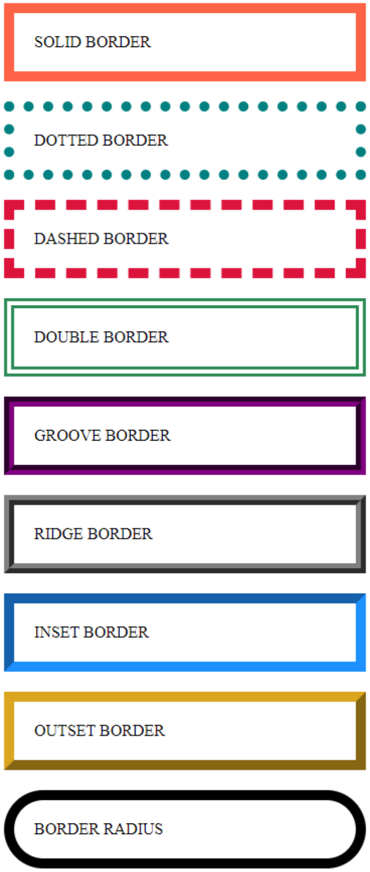
border-color:

sets the color of the border.

The value of the border-color can be: **color name, rgb, or hexcode**

Border Styles

```
.solid{
        border: 10px solid tomato;
}
.dotted{
        border: 10px dotted teal;
}
.dashed{
        border: 10px dashed crimson;
}
.double{
        border: 10px double seagreen;
}
.groove{
        border: 10px groove purple;
}
.ridge{
        border: 10px ridge gray;
}
.inset{
        border: 10px inset dodgerblue;
}
.outset{
        border: 10px outset goldenrod;
}
.radius{
        border: 10px solid black;
        border-radius: 50px;
}
```



SOLID BORDER

DOTTED BORDER

DASHED BORDER

DOUBLE BORDER

GROOVE BORDER

RIDGE BORDER

INSET BORDER

OUTSET BORDER

BORDER RADIUS

Padding

Note:

To apply border on individual sides you can use the following properties:

- border-top
- border-right
- border-bottom
- border-left

And to apply border-radius on an element:

- border-radius

border-radius can use **px or %**

Overflow

The **overflow** property tells the browser what to do if the content contained within a box is larger than the box itself.

overflow: **hidden**;

hides the extra content that does not fit in the box

overflow: **scroll**;

adds a scrollbar to the box so that the user can scroll to see the missing content.

Box Sizing

By default, the width and height of an element is calculated like this:

width + padding + border = actual width of an element

height + padding + border = actual height of an element

The **box-sizing** property allows us to include the **padding** and **border** in an element's total width and height.

Box Shadow

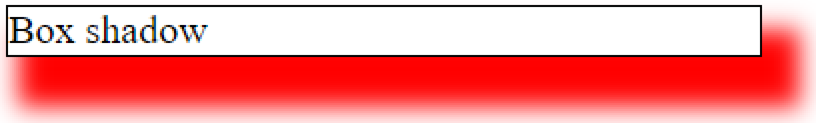
box-shadow

the box-shadow property adds a shadow to an element

```
box-shadow: h-offset v-offset blur spread color;
```

Box Shadow

```
.box-shadow{
    box-shadow: 10px 15px 10px 5px red;
    border: 1px solid black;
}
```



Box shadow

Centering Content

To center a text,
you can simply use **text-align**

To center a block element,
you can simply use:

- **margin-left:** auto;
- **margin-right:** auto;

Font Family

font-family

font-family property defines the font for an element.

```
font-family: Arial, Helvetica, sans-serif;
```

Font Family

SERIF

im

SANS- SERIF

im

MONOSPACE

im

Font Size

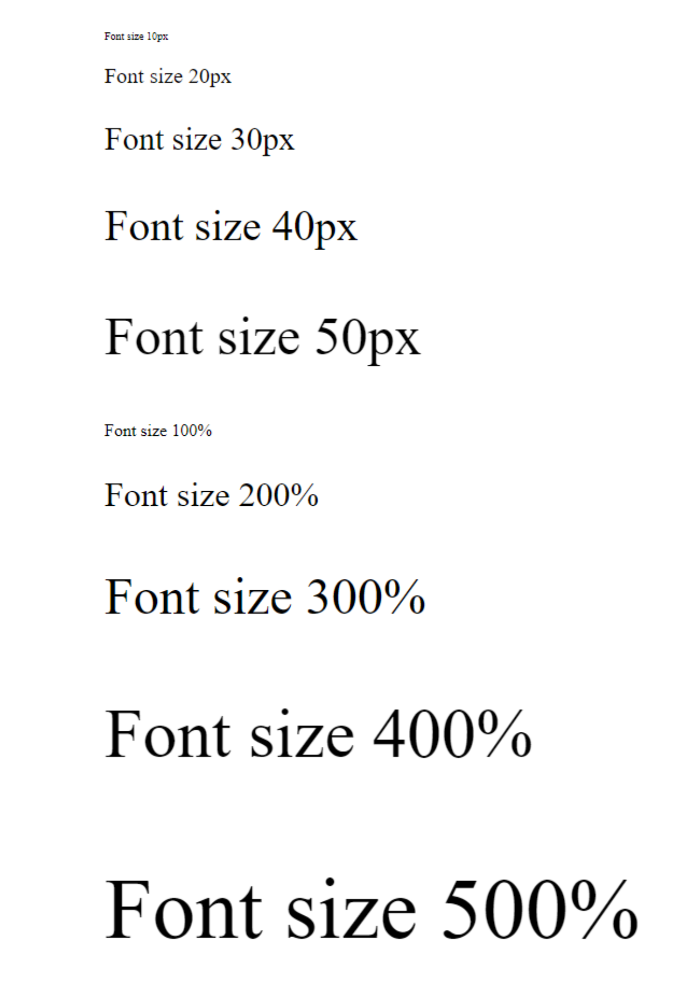
font-size

font-size property defines the size of the font for an element.

The values for font-size are: **px, %, em**

Font Size

```
font-size-10{
      font-size: 10px;
}

font-size-20{
      font-size: 20px;
}

font-size-30{
      font-size: 30px;
}

font-size-40{
      font-size: 40px;
}

font-size-50{
      font-size: 50px;
}
```



Font size 10px

Font size 20px

Font size 30px

Font size 40px

Font size 50px

Font size 100%

Font size 200%

Font size 300%

Font size 400%

Font size 500%

Font Weight

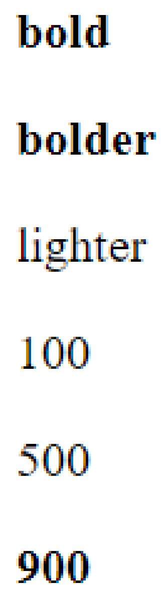
font-weight

font-weight property defines the weight or how thick of thin the font is.

The values for font-weight are: b**old, normal,** **lighter**, or from **100 to 900**.

Font Weight

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

font-weight-bolder{
        font-weight: bolder;
}

font-weight-lighter{
        font-weight: lighter;
}

font-weight-100{
        font-weight: 100;
}

font-weight-500{
        font-weight: 500;
}

font-weight-900{
        font-weight: 900;
}
```



bold

bolder

lighter

100

500

900

Font Style

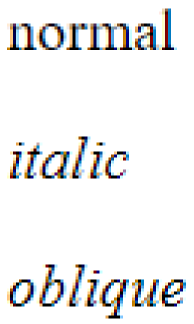
font-style

font-style property specifies the font style of the text.

The values for font-style are: **normal, italic, oblique.**

Font Style

```
font-style-normal{
      font-style: normal;
}

font-style-italic{
      font-style: italic;
}

font-style-oblique{
      font-style: oblique;
}
```



normal

italic

oblique

Text Transform

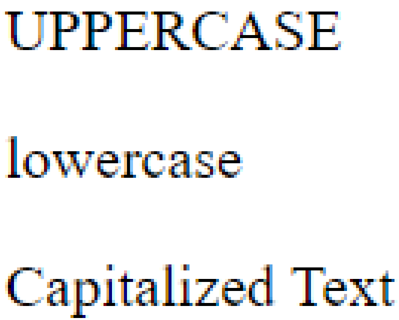
text-transform

text-transform property specifies the capitalization of the text.

The values for text-transform are: **uppercase, lowercase, capitalize.**

Text Transform

```
text-uppercase{
      text-transform: uppercase;
}

text-lowercase{
      text-transform: lowercase;
}

text-capitalize{
      text-transform: capitalize;
}
```



UPPERCASE

lowercase

Capitalized Text

Text Decoration

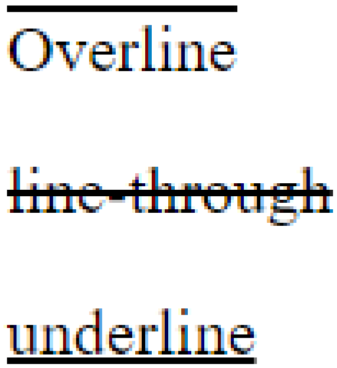
text-decoration

text-decoration property specifies the decoration to the text.

The values for text-decoration are: **overline, underline, line-through.**

Text Decoration

```
text-overline{
	text-decoration: overline;
}

text-line-through{
	text-decoration: line-through;
}

text-underline{
	text-decoration: underline;
}
```



Overline

~~line-through~~

underline

KodeGo

Text Align

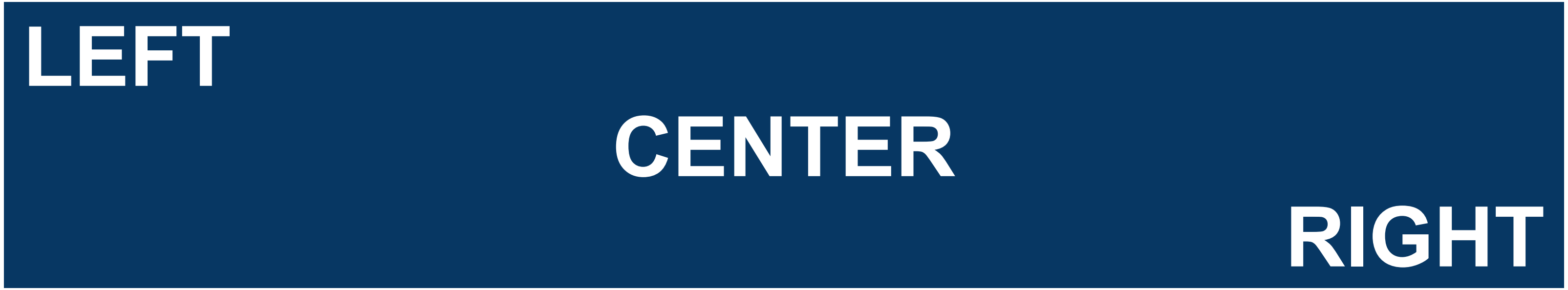
text-align

text-align specifies the alignment of the text.

The values for text-align are: **left, center, right.**

KodeGo

Text Align



LEFT

CENTER

RIGHT

```
selector {
   text-align: center;
}
```

Text Shadow

text-shadow

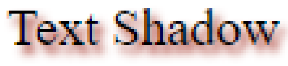
text-shadow adds a shadow to the text.

The syntax of text-shadow is:

```
text-shadow: h-shadow v-shadow blur-radius color;
```

Text Shadow

```
text-shadow{
     text-shadow: 2px 2px 3px #961204;
}
```



Text Shadow

Text Indent

text-indent

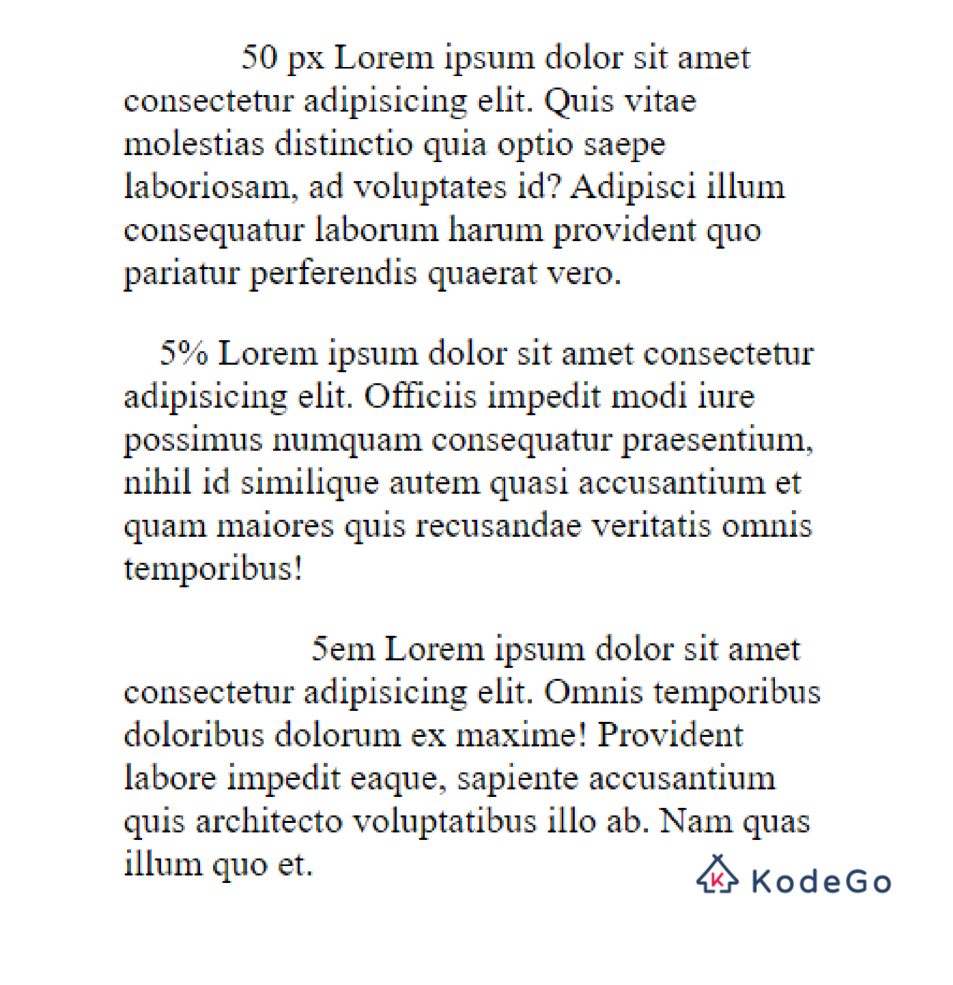
text-indent property specifies the indentation for the first line of a text/paragraph.

The values for text-indent can be: **px, % , em**

KodeGo

Text Indent

```
text-indent-px{
    text-indent: 50px;
}

text-indent-percent{
    text-indent: 5%;
}

text-indent-em{
    text-indent: 5em;
}
```



50 px Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis vitae molestias distinctio quia optio saepe laboriosam, ad voluptates id? Adipisci illum consequatur laborum harum provident quo pariatur perferendis quaerat vero.

5% Lorem ipsum dolor sit amet consectetur adipisicing elit. Officiis impedit modi iure possimus numquam consequatur praesentium, nihil id similique autem quasi accusantium et quam maiores quis recusandae veritatis omnis temporibus!

5em Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis temporibus doloribus dolorum ex maxime! Provident labore impedit eaque, sapiente accusantium quis architecto voluptatibus illo ab. Nam quas illum quo et.

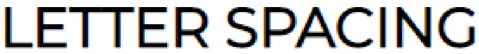
Letter Spacing

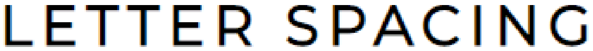
letter-spacing

letter-spacing property **increases** or **decreases** the **spacing** in between the **letters** in a text.

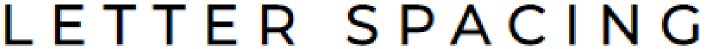
The values for letter-spacing can be in: px

Letter Spacing

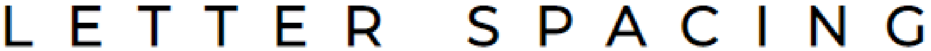

LETTER SPACING


LETTER SPACING

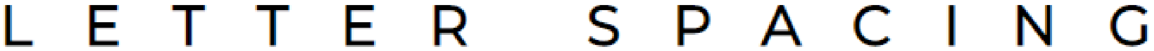
```
letter-spacing: 5px;
```


LETTER SPACING

```
letter-spacing: 10px;
```


LETTER SPACING

```
letter-spacing: 20px;
```


LETTER SPACING

```
letter-spacing: 30px;
```

LETTER SPACING

```
letter-spacing: 50px;
```

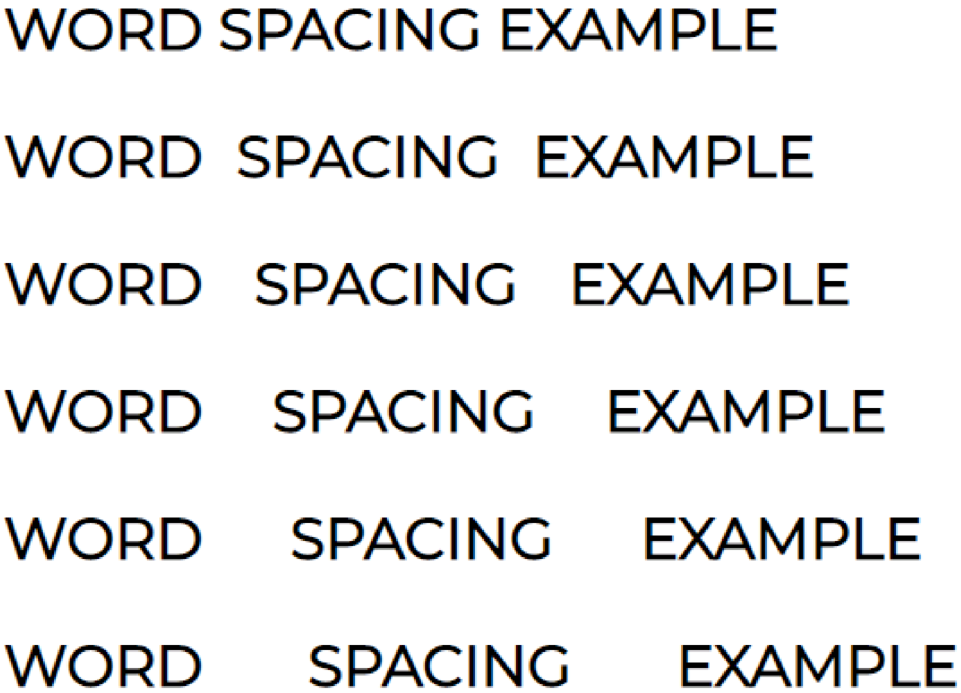
Word Spacing

word-spacing

word-spacing property **increases** or **decreases** the **spacing** in between the **words** in a text.

The values for letter-spacing can be in: **px**

Word Spacing



WORD SPACING EXAMPLE

WORD SPACING EXAMPLE

WORD SPACING EXAMPLE

WORD SPACING EXAMPLE

WORD SPACING EXAMPLE

WORD SPACING EXAMPLE

```
word-spacing: 10px;

word-spacing: 20px;

word-spacing: 30px;

word-spacing: 40px;

word-spacing: 50px;
```

KodeGo

List Style

UNORDERED LIST

list-style-type: none;
● list-style-type: disc;
○ list-style-type: circle;
■ list-style-type: square;

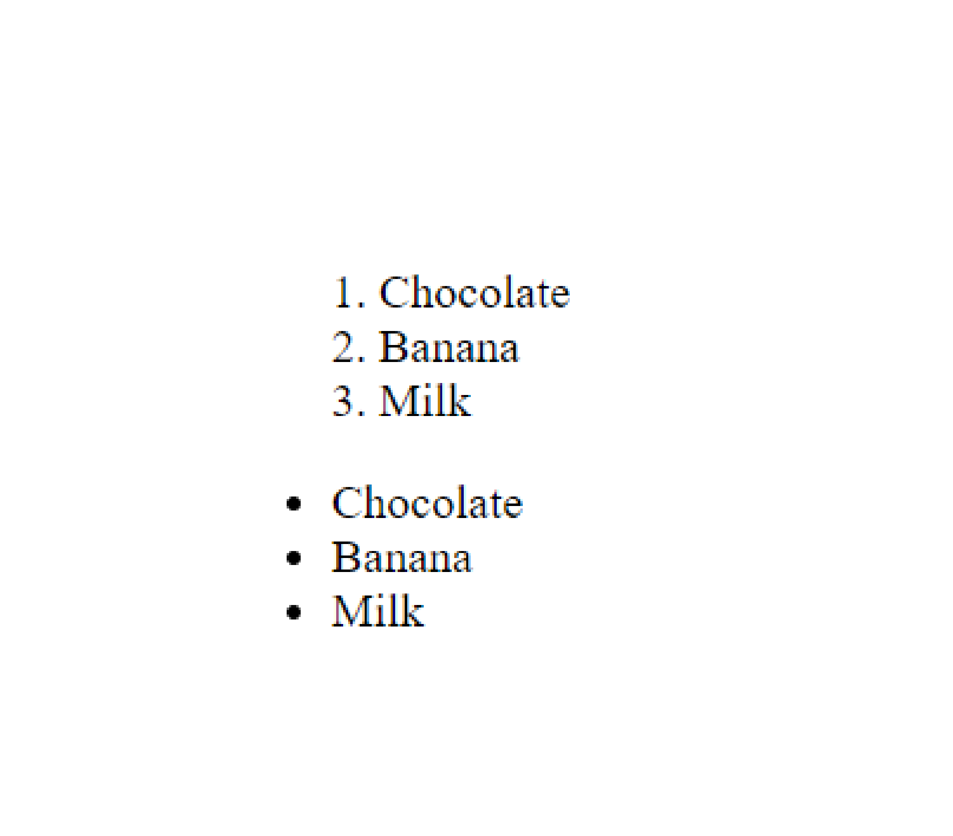
ORDERED LIST

1. list-style-type: decimal-leading-zero;
a. list-style-type: lower-alpha;
i. list-style-type: lower-roman;
I. list-style-type: upper-roman;
A. list-style-type: upper-alpha;

KodeGo

List Style Position

```
ol{
    list-style-position: inside;
}

ul{
    list-style-position: outside;
}
```



1. Chocolate
2. Banana
3. Milk

- Chocolate
- Banana
- Milk

List Style Shorthand

```css
list-style: list-style-type list-style-position;
```

```css
list-style: square outside;
```

CSS Images

The **height** property adjusts the height of the image. You can use % or px

The **width** property adjusts the width of the image. You can use % or px

CSS Tables

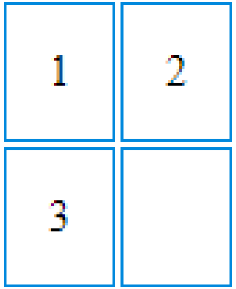
CSS Tables can use some previous css properties that we learned.
Properties such as:

- width
- height
- color
- font-size
- text-align
- and etc.

CSS Tables

empty-cells: show;

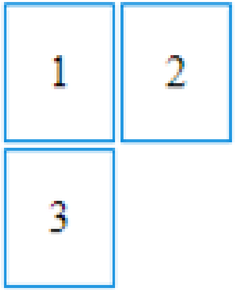
To show the empty cells in the table.



1	2
3	

empty-cells: hide;

To hide the empty cells in the table.

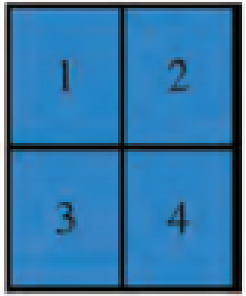


1	2
3	

CSS Tables Border Collapse

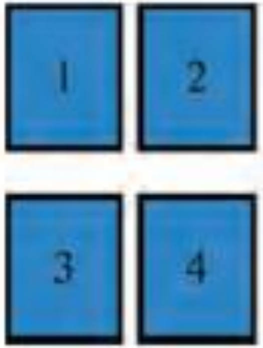
border-collapse: collapse;

Table borders are collapse into a single border.



border-collapse: separate;

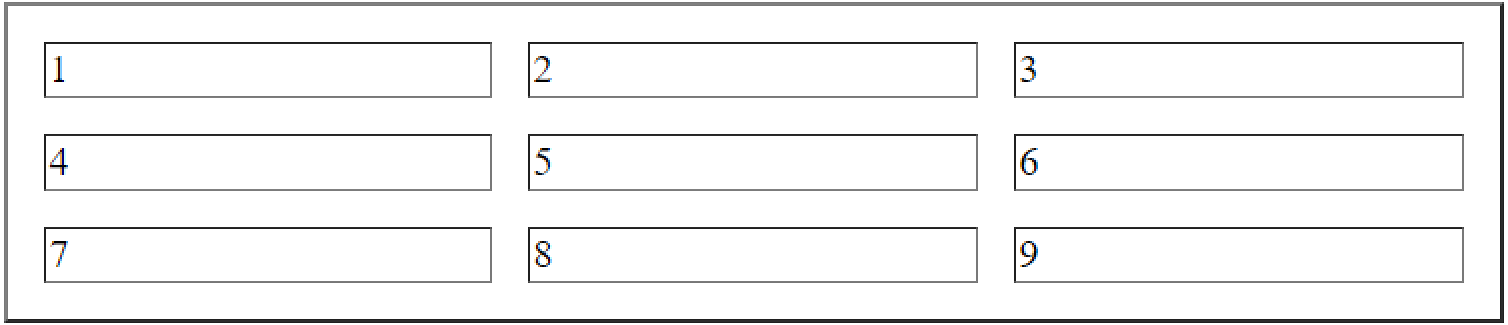
Table borders are detached from each other.



CSS Tables Border Spacing

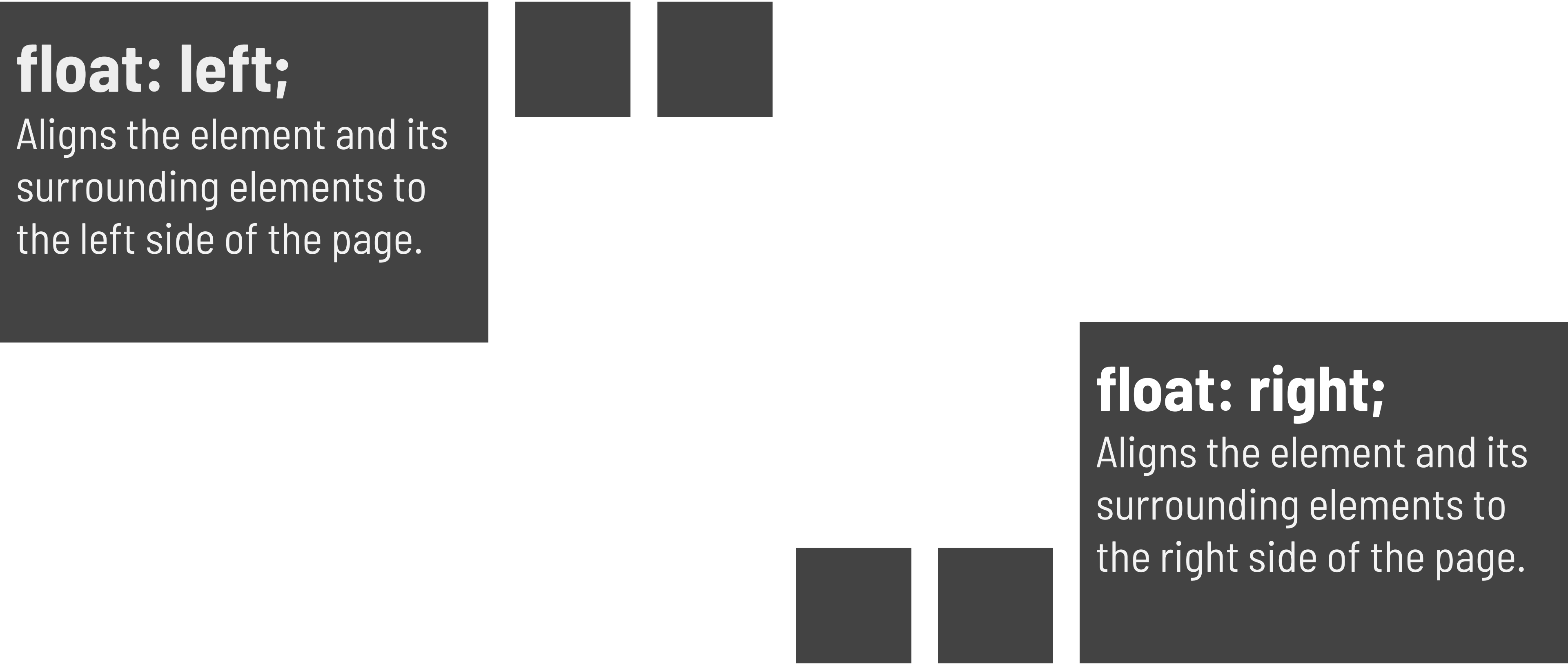
border-spacing: 15px;

The spacing of both the horizontal and vertical spacing will be adjusted.



1	2	3
4	5	6
7	8	9

CSS Float



float: left;

Aligns the element and its surrounding elements to the left side of the page.

float: right;

Aligns the element and its surrounding elements to the right side of the page.

Clear

clear: none;- the element is not moved down to clear past floats.

clear: left; - the element is moved down to clear past left floats.

clear: right;- the element is moved down to clear past right floats.

clear: both; - the element is moved down to clear past both left and right floats.

Position

position: static; - this is the default positioning; the element will stick to the normal page flow. So if there is a **left/right/top/bottom/z-index** set then there will be no effect on that element.

position: relative; -an element's original position remains in the flow of the page, just like the static value. But now **left/right/top/bottom/z-index** will work. The positional properties can move the element from the original position in that direction.

Position

position: absolute; - the element is removed from the flow of the page and other elements will behave as if it’s not even there. All the other positional properties will work on it.

position: fixed; - the element is removed from the flow of the document like absolutely positioned elements. In fact they behave almost the same, only fixed positioned elements are always relative to the document, not any particular parent, and are unaffected by scrolling.

Display

`display: block;`- displays the element as a ***block*** element.
Block level elements do not sit inline instead they will create a new line break. By default (without setting a width) they take up as much horizontal space as they can.

`display: inline;`- displays the element as an ***inline*** element.
An inline element will accept margin and padding, but the element still sits inline as you might expect. Margin and padding will only push other elements horizontally away, not vertically. An inline element will **not** accept **height** and **width**. It will just ignore it.

Display

display: inline-block;- the element will have the characteristic of a *block* element but sits on a *line*. You are now able to set the *width* and *height*, which will be respected.

display: none;- hides the element.

Thank you!