BOOTCAMP

v2023

Introduction to HTML

What is HTML?

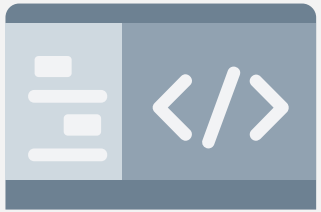
HyperText Markup Language

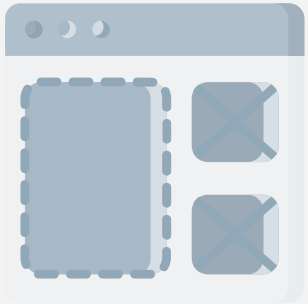
HTML describes the structure of the pages.
It uses elements and tags.

It is the most basic building block of the Web.

Software used to Create Websites

Visual Studio Code





01. Text-Editor

VS Code, Sublime Text, Notepad++, Atom

02. Browser

Google Chrome, Safari, Mozilla Firefox, Internet Explorer

Kredo

HTML

```
<h1>Top level heading: Maybe a page title</h1>

<p>A paragraph of text. Some information we wo
uld like to communicate to the user. This can
be as long or short as we would like.</p>

<ol>
  <li>Number one on the list</li>
  <li>Number two</li>
  <li>A third item</li>
</ol>
```

Web Page

Top level heading: Maybe a page title

A paragraph of text. Some information we would like to communicate to the user. This can be as long or short as we would like.

1. Number one on the list
2. Number two
3. A third item

HTML STRUCTURE

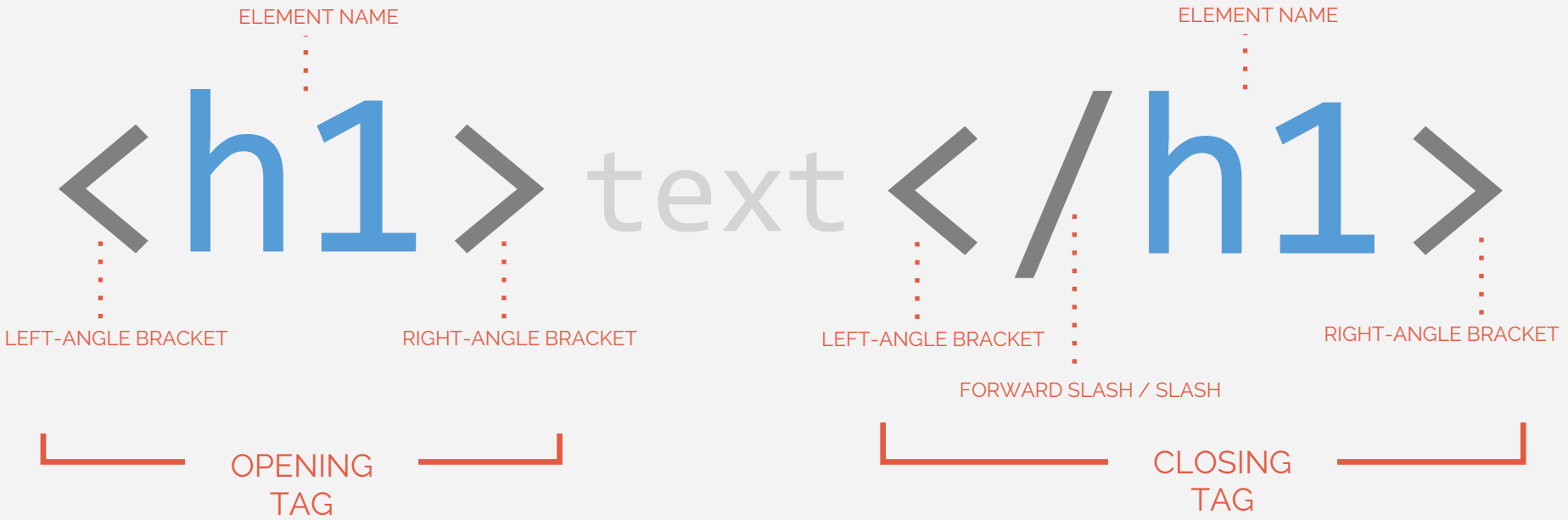
Kredo

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document Title</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <!-- contents here -->
</body>
</html>
```

VS Code shortcut: ! or html:5

HTML Tags, Elements, and Attributes

HTML SYNTAX


ELEMENT NAME
<h1>
text
</h1>
ELEMENT NAME
LEFT-ANGLE BRACKET
RIGHT-ANGLE BRACKET
LEFT-ANGLE BRACKET
FORWARD SLASH / SLASH
RIGHT-ANGLE BRACKET
OPENING TAG
CLOSING TAG

HTML Elements & Tags

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

HTML Syntax:

```
<h1 class="topHeading">Top level heading: Maybe a page title</h1>
```

Case-insensitive

Tags in HTML are case-insensitive, i.e. they can be written in uppercase or lowercase.

```
1 | <title>My HTML Page</title>
```

```
2 | <TITLE>My HTML Page</TITLE>
```

```
3 | <Title>Coffee Bean</Title>
```

Nesting Elements

Elements can be placed within other elements too.

```
1 | <p>My name is Kirby</p>
```

```
2 | <p>My name is <strong>Kirby</strong></p>
```

x

```
3 | <p>My name is <strong>Kirby</p></strong>
```

HTML Comments

HTML comments are not displayed in the browser, but they can help document your HTML source code.

```
<!-- This is an example of a comment -->
```

HTML comments

While Comments are helpful in documenting and reminding you about your code, having unnecessary comments can create confusion to the collaborators of the code, specially if there is more 1 person who is working with the current project.

What to do:

- keep comments short
- make it brief and concise
- omit comments that doesn't make sense or not useful anymore
- don't make it like your notes/journal/diary where it contains unnecessary info

Block Elements vs Inline Elements

Block Elements always start on a new line and takes up the full width of a page, from left to right.

A block-level element can take up one or multiple lines and has a line break before and after the element.

BLOCK ELEMENTS

BLOCK ELEMENTS

Kredo

```
<h1>Top level heading</h1>

<p>My name is Kirby.</p>

<p>This will occupy an entire horizontal space.</p>

<div>This is also a block element.</div>
```

Block Elements vs Inline Elements

Inline Elements do not cause a line break or start on a new line, and does not take up the full width of a page.
Its width only extends as far as it is defined by its tags.
It is usually used within other HTML elements.

INLINE ELEMENTS

INLINE ELEMENTS

Kredo

```
<p>This contains an <em>inline-level</em> element.</p>
```

```
<p><strong>Warning:<strong> Do not touch.</p>
```

```
<p>Span is a generic <span>inline container.<span></p>
```

Div vs Span

Div, also known as ***division tag***,is used in HTML to make divisions of content on the web page like text, images, header, footer, navigation bar, etc. It is also used to group elements together.

The div element is a **block** element.

Div vs Span

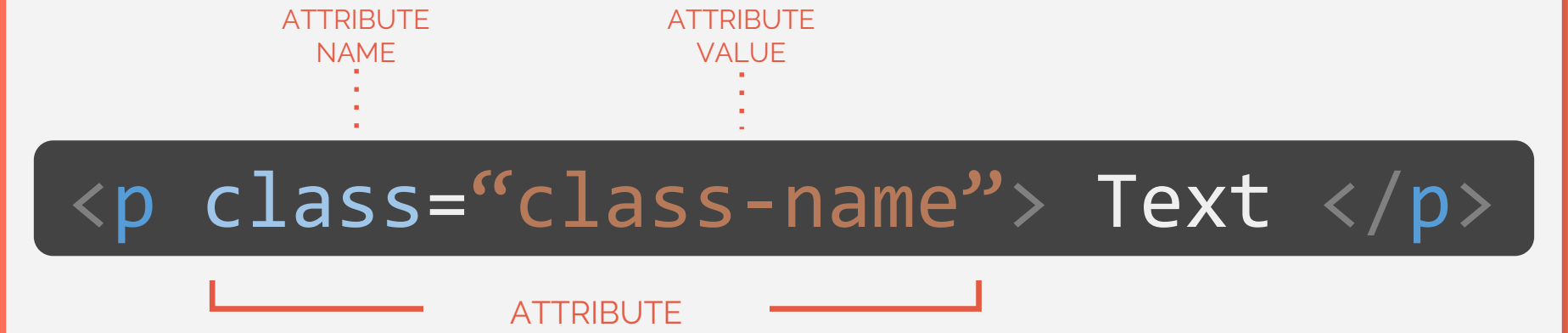
Span is an inline container for inline elements and content. It used to group inline elements for styling purposes.

The span element is an **inline** element.

HTML Attributes

Attributes provide additional information about an element.

HTML Attributes



HTML Attributes

- **class** - specifies a class name to an element; used for styling
- **id** - specifies a unique identifier to an element; used to target an element

HTML Typography

HTML Headings

Headings are used to create headings in descending order of importance where **<h1>** is the most important and **<h6>** is the least.

It is basically the headlines for your web page.

HEADING

Kredo

```
<h1>Most important heading</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Least important heading</h6>
```

h1 to h6 only

HTML Paragraph

Paragraph elements are used to create paragraphs.

To create a paragraph:
** ** Your text here ** **

PARAGRAPH

Kredo

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Beatae id repellat cupiditate eaque. Vitae sit sed dolores totam quod. Molestiae ipsam nam accusamus, quos eligendi at odio obcaecati! Velit, veniam!</p>
```

VS Code shortcut: **p>lorem30**

Line Breaks

Line Breaks are used to add a single line break in a text.

To insert a line break use:
**
**

LINE BREAK

Kredo

```
<p>Hold fast to dreams,<br>for if dreams die,<br>life is a broken-winged
bird<br>that cannot fly.</p>
```

Line Breaks

Take note, line breaks are not used to create huge "space" between rows.

```
<p> First Paragraph … </p>
<br>
<br>
<br>
<p> Second Paragraph … </p>
```

Line Breaks

Line Breaks should not be used excessively, or just because you want the element/content to be on the next line/row.

Make use of the **BLOCK** elements such as ** ** group your contents or take them to another row.

Line Break

```
<span> Apple </span>
<span> Banana </span>

<br>
<br>

<span> Cat </span>
<span> Dog </span>
```

```
<div>
    <span> Apple </span>
    <span> Banana </span>
</div>

<div>
    <span> Cat </span>
    <span> Dog </span>
</div>
```

Horizontal Rule

Horizontal Rule is used to add a thematic break between paragraph-level elements.

To insert a horizontal rule:

<hr>

HORIZONTAL RULE

Kredo

```
<p>First story. Lorem ipsum, dolor sit amet consectetur adipisicing elit. Assumenda fugiat architecto eaque neque adipisci at blanditiis suscipit non sit laboriosam numquam provident officiis quo quibusdam perspiciatis</p>

<hr>

<p>Second story. Lorem ipsum dolor, sit amet consectetur adipisicing elit. Eveniet, omnis corporis? Deserunt natus placeat vero ad omnis perferendis quia veniam!</p>
```

HTML Links

HTML Links

Anchor tags <a> are used to define a hyperlink that links one page to another page, as well as files, location, or any URL.

The **href** attribute is the most important attribute of the HTML **<a>** tag. and which links to destination page or URL.

LINKS

Kredo

```
<a href="www.twitter.com">Twitter</a>
```

to an external page

```
<a href="contact-us.html">Contact Us</a>
```

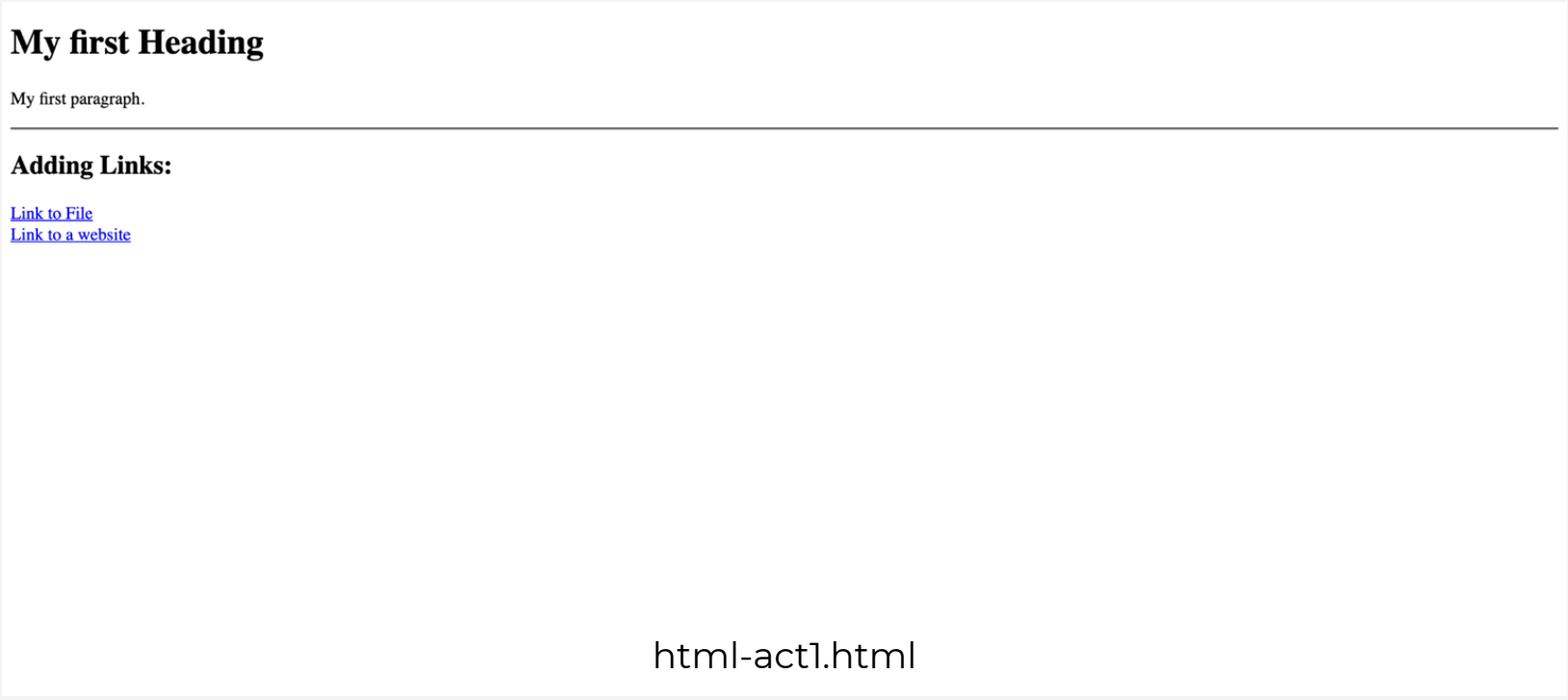
to another page

```
<div id="top">Header</div>

<a href="#top">Back to top</a>
```

to another location on the same page

HTML Activity 1



html-act1.html

HTML Lists

HTML Ordered Lists

Ordered List is a type of list that is numerically or alphabetically ordered.

1. Item 1
2. Item 2
3. Item 3

ORDERED LIST

Kredo

```
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ol>
```

HTML Unordered Lists

Unordered List is a type of list that is not ordered. It is rendered as a bulleted list.

- Item 1
- Item 2
- Item 3

UNORDERED LIST

Kredo

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

HTML Lists

The <ul> and <ol> elements may be **nested** as desired, and may alternate without restrictions.

1. Fruits
    - Banana
    - Apple
2. Desserts
    - Chocolate Bar
    - Cake

NESTED LIST

```
<ol>
    <li>Fruits
        <ul>
            <li>Banana</li>
            <li>Apple</li>
        </ul>
    </li>
    <li>Desserts
        <ul>
            <li>Chocolate Bar</li>
            <li>Cake</li>
        </ul>
    </li>
</ol>
```

Kredo

HTML Images

HTML Images

The **img** element adds an image into the document.

The **src** attribute defines the URL or the path to the image.

The **alt** attribute defines the alternate text for an image for when it is not displayed.

HTML Images

The **height** attribute defines the height of the image. Use **%** (percentage) or **px** (pixel).

The **width** attribute defines the width of the image. Use **%** (percentage) or **px** (pixel).

IMAGE

Kredo

```
<img src="assets/images/dog.jpg" alt="brown dog"
width="500px" height="100%">
```

HTML Activity 2

The Rabbit

- Rabbits are amazing athletes. They can jump as high as 90 centimetres in one leap!
- A baby rabbit is called a kit, a female is called a doe and a male is called a buck.
    - Rabbits are very social creatures that live in groups.

Facts about rabbits:

1. A single bunny is a lonely bunny
2. They can live for up to 12 years
3. They are masters of hearing

html-act2.html

HTML Tables

HTML TABLES

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

To create an HTML table:
<table></table>

TABLE

Kredo

<table>

1	2	3
4	5	6
7	8	9

</table>

Table Row

The **tr** element defines a **row** inside a table. A table row can contain **one** or **more <th>** or **<td>.**

To create a row:
<tr> </tr>

TR

Kredo

<table>

1st row <tr>	1	2	3	</tr>
2nd row <tr>	4	5	6	</tr>
3rd row <tr>	7	8	9	</tr>

</table>

Table Data

The **td** element defines a **data cell** inside the table.

To create a data cell:
<td> </td>

TD

<table>

1st row <tr>	<td>	1	</td>	<td>	2	</td>	<td>	3	</td>	</tr>
2nd row <tr>	<td>	4	</td>	<td>	5	</td>	<td>	6	</td>	</tr>
3rd row <tr>	<td>	7	</td>	<td>	8	</td>	<td>	9	</td>	</tr>

</table>

Table Header

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

To create a table header:
<thead> </thead>

Header Cell

The **header cells** contain the header information or the column names of the table. By default, the font weight is **bold**, and aligned to the **center**.

To create a header cell:
<th> </th>

THEAD + TH

Kredo

<table>

<thead> <tr> <th> Name </th> <th> Age </th> <th> Email </th> </tr> </thead>

<tr> John 29 john@email.com </tr>

<tr> Jane 27 jane@email.com </tr>

</table>

Table Body

tbody is used to group the **body content** of an HTML table.

To create a table body:
<tbody> </tbody>

TBODY

Kredo

<table>

<tr>

Name	Age	Email
John	29	john@email.com
Jane	27	jane@email.com

</tr>

<tbody>

<tr>

</tr>

<tr>

</tr>

</tbody>

</table>

Colspan

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

Syntax:

<td **colspan="2"**>

Colspan

Name	<th colspan="2"> Contact Numbers </th>	
Bill	12345	67890
Steve	13579	24680

Rowspan

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

Syntax:

<td **rowspan="2"**>

TABLES: Rowspan

Kredo

Name	Bill	Steve
<td rowspan="2"> **Contact Numbers** </td>	12345	13579
	67890	24680

HTML Activity 3

My Work Plan for the Week					
	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
Activity	Meeting at Room 202	Business Lunch at 1pm	Project due by 5pm	Web Conference	Early Finish (4pm)
	Club Practice	Meeting at 2pm	NONE	NONE	Submit Report

html-act3.html

HTML Forms

HTML 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.

FORM STRUCTURE

Kredo

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

    <input type="submit" name="btn_submit" value="Submit">
</form>
```

Form Structure

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

The **method** attribute defines **how** the form-data is passed.

Labelling Form Controls

Labelling form controls is 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.

Kinds of Form Controls

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

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="last-
name">
```

Web Page

First Name: John Last Name: Doe

Name

The **name** attribute specifies the name of the form control to use **for form submission**. It is submitted with the value.

Password Input

Kredo

Syntax

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

Web Page

Password: •••••

Number Input

Kredo

Syntax

```
<label for="age">Age</label>
<input type="number" name="age" id="age">
```

Web Page

Age 18

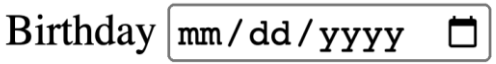
Date Input

Kredo

Syntax

```
<label for="bday">Birthday</label>
<input type="date" name="bday" id="bday">
```

Web Page



Email Input

Kredo

Syntax

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

Web Page

Email: test@email.com

Textarea

Kredo

Syntax

```
<label for="comments">Comments:</label>
<br>
<textarea name="comments" id="comments"
cols="60" rows="10"></textarea>
```

Web Page

Comments:

Radio Button

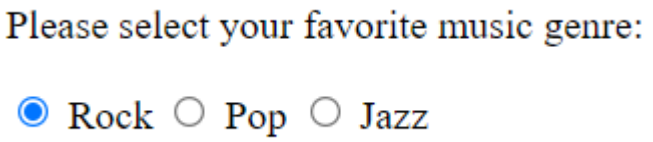
Syntax

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

<input type="radio" name="music" id="pop"
value="pop">
<label for="pop">Pop</label>

<input type="radio" name="music" id="jazz"
value="jazz">
<label for="jazz">Jazz</label>
```

Web Page



Kredo

Checkbox

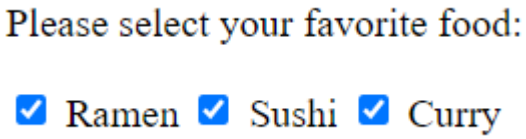
Syntax

```
<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

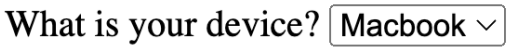


Dropdown List

Syntax

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

Web Page

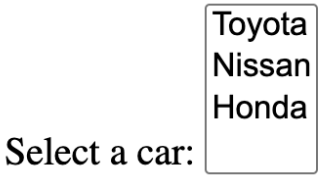


Multiple Select

Syntax

```
<label for="car">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



File Input Box

Syntax

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

Web Page

Kredo

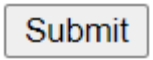
Input:Submit

Kredo

Syntax

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

Web Page



Button:Submit

Syntax

```
<button type="submit" name="submit">Submit</button>
```

Web Page

Submit

Kredo

Form Validation

HTML5 has the ability to validate most user data once submitted. This is done by using **validation attributes** in the form elements.

Form Validation

- **required** attribute checks whether the input field is **filled in or not** before being submitted.
- **min** / **max** attribute limits the **minimum** or **maximum** value of a **numeric type** input fields.

Placeholder

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

Placeholder

Syntax

```
<label for="full-name">Full Name</label>
<input type="text" name="full_name" id="full-name"
placeholder="John Doe">
```

Web Page

Kredo

HTML Activity 4

Please fill this survey form

First Name: MI: Last Name:

City: State: Zip Code:

Choose a Sport:

Basketball Football Hockey

Choose a Year:

Freshman Sophomore Junior Senior

Choose Your Favorite Food:

Chips
Pizza
Nachos

Please let us know your comments

Send it Clear it

html-act4.html

Introduction to Kuregori (クレゴリくん)

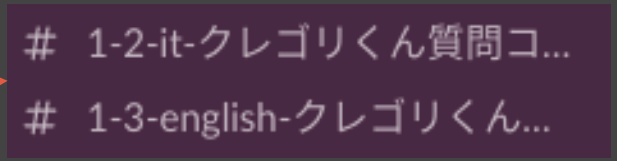
Purpose

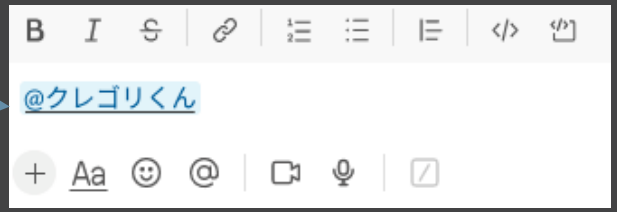
Kuregori can be used as your **virtual** tutor in Slack

You can make good use of it for **self-learning**!

Instructions

1. Go to **Slack (Kredo Online Camp) Workspace**
2. Go to the channels:
   a. #1-2-IT-Kredo Q&A Corner for programming-related questions
   b. #1-3-English-Kredo Q&A Corner for english-related questions
3. Type "**@クレゴリくん**" to mention Kuregori.
4. After mentioning him, input your request or questions for him.
5. **Send!**





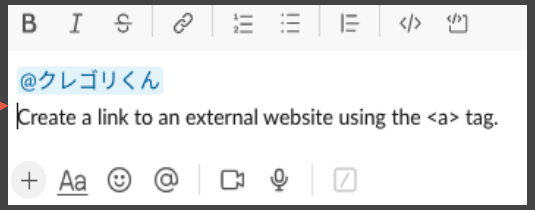
Note:
Mention ***@クレゴリくん*** *again if you have additional messages*

Learn HTML Code

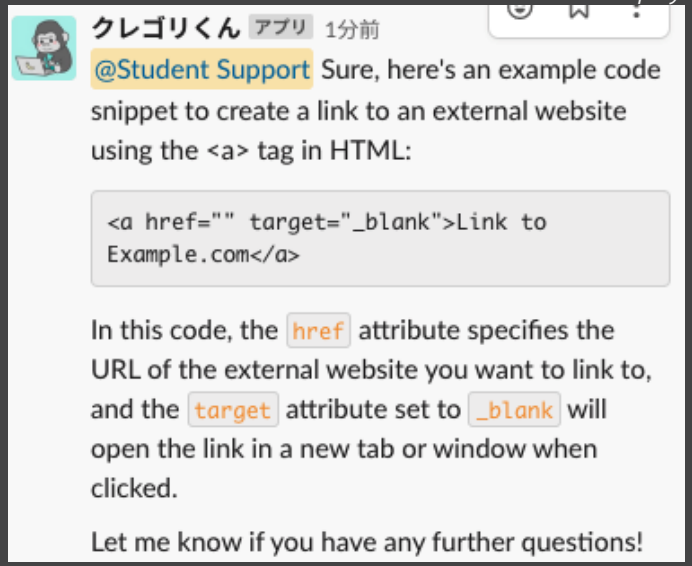
Examples:

- "Add an image to an HTML Page using the **<img> tag**. Make sure to include the alt attribute."
- "Create a link to an external website using the **<a> tag**."
- "Use the **<ul>** and **<li>** tags to create an unordered list on an HTML Page."
- "Use the **<table>** tag to create a simple table on an HTML Page."

** Kuregori will reply using a thread.*



Note:
It might take Kuregori some time to reply.



Reviewing HTML Codes

Kuregori reviews and corrects your code.

Example:

```
<p>Please select your favorite food:</p>
<input typ="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>


Please make this code correct.
```

Please select your favorite food:
<input typ="checkbox" name="food" id="ramen" value="ramen">
<label for="ramen">Ramen</label>

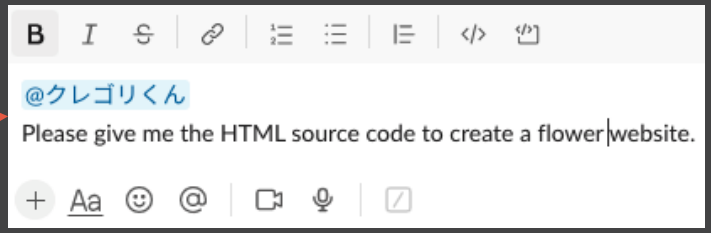
<input type="checkbox" name="food" id="sushi" value="sushi">
<label for="sushi">Sushi</label>

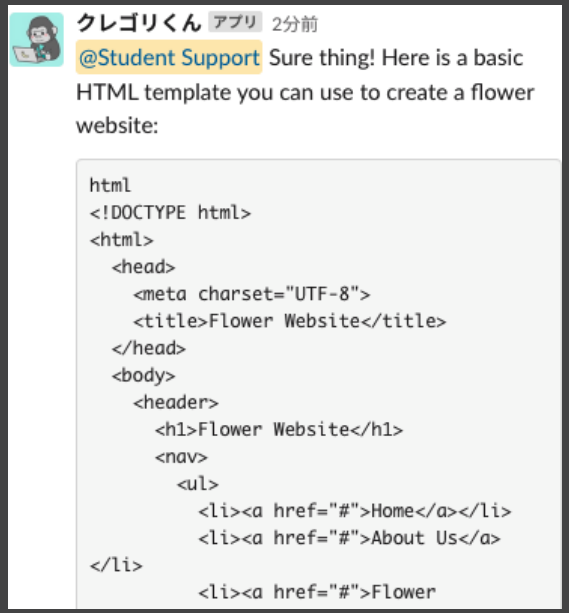
Creating HTML Code Snippets

Kuregori can help create HTML Code Snippets/Templates

Example:

Please give me the HTML source code to create a flower website.





ChatGPT

Having good **prompt skills** is crucial.

Use it more and study efficiently!

Introduction to CSS

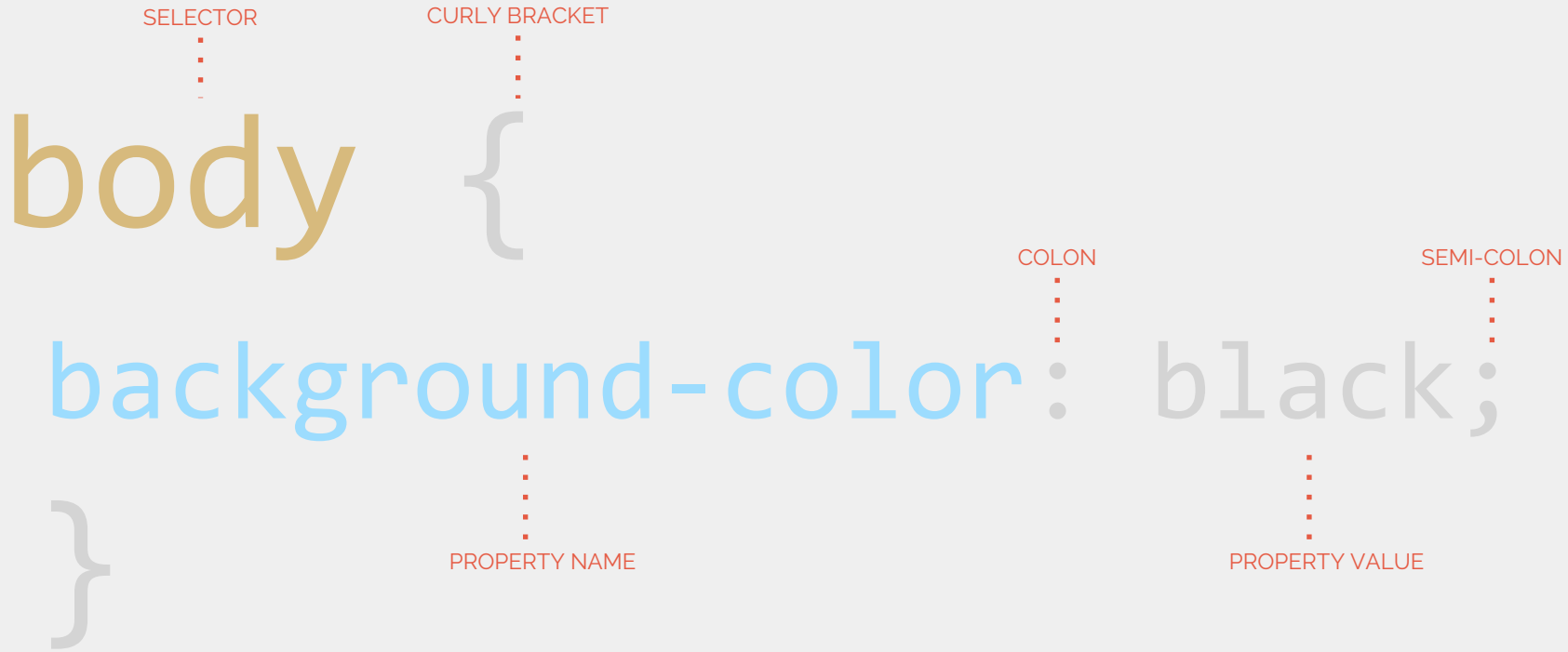
What is CSS?

Cascading Style Sheets

Used to style and layout web pages.

It is a simple design language to simplify the process of designing web pages.

CSS Syntax


SELECTOR
CURLY BRACKET
body {
COLON
SEMI-COLON
background-color: black;
}
PROPERTY NAME
PROPERTY VALUE

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 SELECTORS UNIVERSAL

CSS

```
1  * {
2      color: blue;
3  }
```

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

CSS SELECTORS ELEMENT

Kredo

HTML

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

CSS

```
1   p {
2       color: blue;
3   }
```

CSS SELECTORS CLASS

Kredo

HTML

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

CSS

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

CSS SELECTORS ID

Kredo

HTML

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

CSS

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

Class vs ID

class attribute is used to identify **more than one** element.

id attribute is used to identify **one unique** element.

Kredo
CSS Rules

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

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

CSS

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

OR using a parent class

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

CSS Typography

CSS Color

Kredo

SALMON

rgb(250, 128, 114)

opacity: 1

rgba(250, 128, 114, 1)

#fa8072

CSS Color

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

CSS Color

color name

All modern browsers support around **140 color name values**. However, this is *not* the practical way of applying CSS colors.

rgb

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

CSS Color

opacity

The opacity value specifies the transparency of the object/element.
The property value of opacity should only be in between **0 and 1**. The lower the value, the more the transparency

rgba

RGBA is just like RGB but with the **alpha channel** which determines the **opacity** of the object/element.

CSS Color

hexcode

Hexcode is specified with the pattern **#RRGGBB**: 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**.

Color

```
p {
    color: red;
}

p {
    color: rgb(0, 0, 255);
}

p {
    color: rgba(0, 0, 255, .5);
}

p {
    color: #00ff00;
}
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita quis aperiam quam quaerat amet, sint laudantium nemo? Necessitatibus, ullam quos.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita quis aperiam quam quaerat amet, sint laudantium nemo? Necessitatibus, ullam quos.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita quis aperiam quam quaerat amet, sint laudantium nemo? Necessitatibus, ullam quos.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita quis aperiam quam quaerat amet, sint laudantium nemo? Necessitatibus, ullam quos.

Kredo

Font Family

font-family

font-family property defines the font family name for an element.

FONT-FAMILY

Kredo

SERIF	SANS- SERIF	MONOSPACE
im	im	im

Kinds of Font Family

- **serif** - fonts that have finishing strokes or "tails" at the end
- **sans-serif** - fonts that do not have "tails" or just have have plain ends
- **monospace** - fonts that have the same fixed width

Font Family

```
p {
    font-family: serif;
}
```

```
p {
    font-family: sans-serif;
}
```

```
p {
    font-family: monospace;
}
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita quis aperiam quam quaerat amet, sint laudantium nemo? Necessitatibus, ullam quos.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita quis aperiam quam quaerat amet, sint laudantium nemo? Necessitatibus, ullam quos.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita quis aperiam quam quaerat amet, sint laudantium nemo? Necessitatibus, ullam quos.

Kredo

Font Size

font-size

font-size property defines the size of the font.

Font size may be defined in **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: **bold**, **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

Text Align

text-align

text-align specifies the alignment of the text.

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

TEXT ALIGN

Kredo

LEFT

CENTER

RIGHT

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

Text Shadow

text-shadow

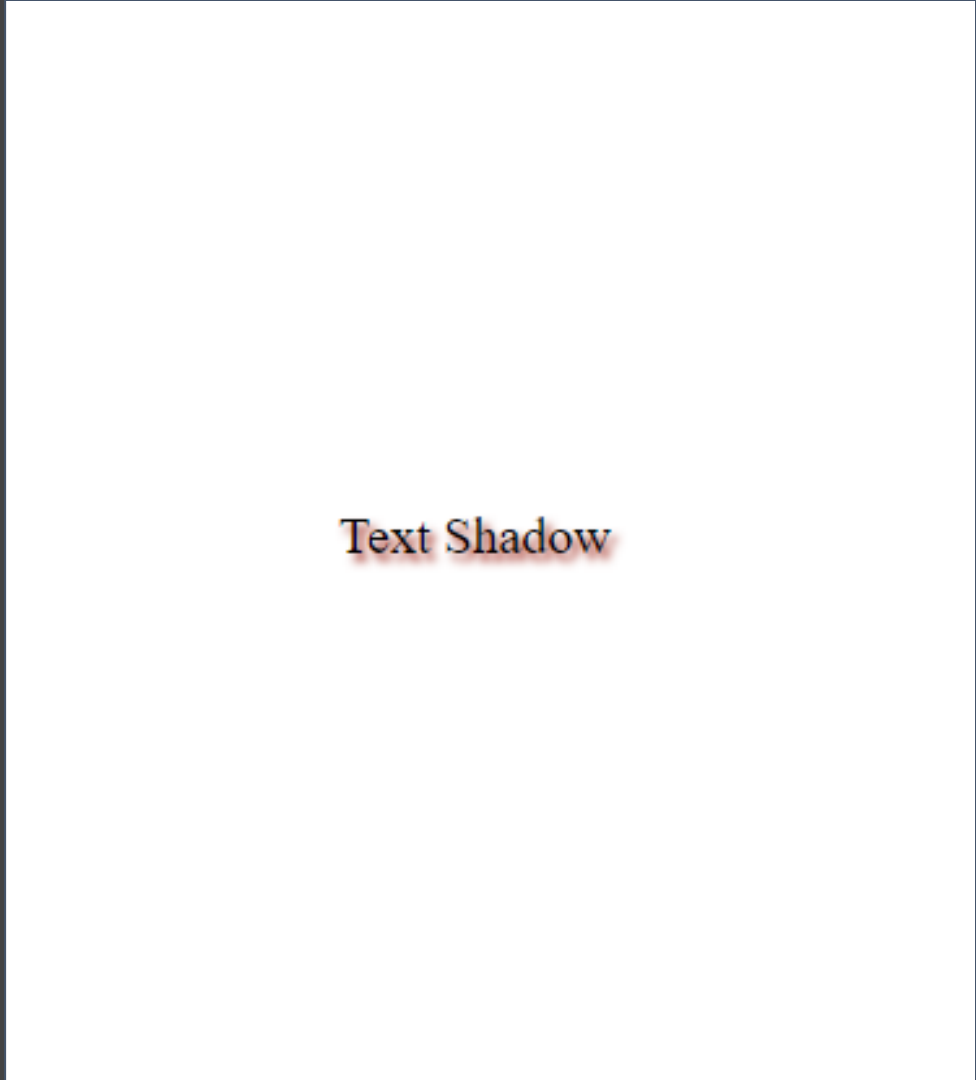
text-shadow adds a shadow to the text.

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 Indent

text-indent

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

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

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

Kredo

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

Kredo

WORD SPACING EXAMPLE

WORD SPACING EXAMPLE `word-spacing: 10px;`

WORD SPACING EXAMPLE `word-spacing: 20px;`

WORD SPACING EXAMPLE `word-spacing: 30px;`

WORD SPACING EXAMPLE `word-spacing: 40px;`

WORD SPACING EXAMPLE `word-spacing: 50px;`

CSS Lists

LIST STYLE TYPE

Kredo

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***;

a. list-style-type: ***lower-alpha***;

A. list-style-type: ***upper-alpha***;

i. list-style-type: ***lower-roman***;

I. list-style-type: ***upper-roman***;

LIST STYLE POSITION

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

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

- Lorem ipsum dolor sit amet consectetur a
  accusamus et doloremque quis ratione ac
  ipsa harum, quos porro dignissimos molli
- Lorem ipsum, dolor sit amet consectetur
  magni necessitatibus, aliquid dolorum. Ve

- Lorem ipsum dolor sit amet consectetu
  facere exercitationem sint cupiditate impe
  mollitia ea nisi vero natus obcaecati quis,
- Lorem ipsum dolor, sit amet consectet
  aut a sapiente fugit, ipsum obcaecati itaqu

list-style shorthand

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

CSS Images

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**

BACKGROUND IMAGE

Kredo

ADDING A BACKGROUND IMAGE

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

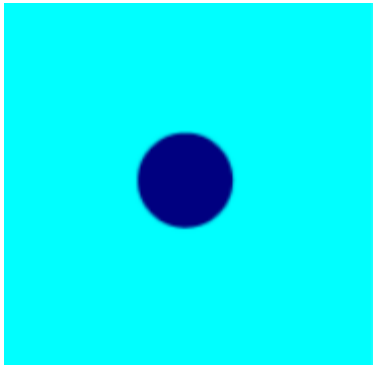
Note: Make sure to define the height and width of the element

BACKGROUND REPEAT

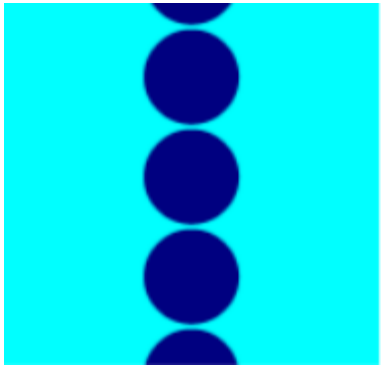
Kredo

background-repeat:

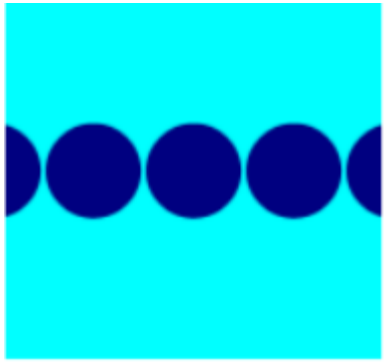
no-repeat;



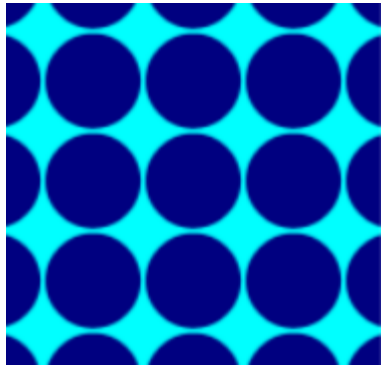
repeat-y;



repeat-x;



repeat;



BACKGROUND SIZE

Kredo

```
background-size: cover;
```

```
background-size: contain;
```

Note: You can also use this

```
background-size: 200px 100px;
```

first value = horizontal size

second value = vertical size

BACKGROUND POSITION

Kredo

```
background-position: center center;
```

BACKGROUND ATTACHMENT

Kredo

```
background-attachment: fixed;
```

The background image stays in the same position on the page.

```
background-attachment: scroll;
```

The background image scrolls with the page.

Background

background shorthand

```
background: bg-color bg-image bg-repeat bg-attachment bg-position;
```

background-size is separated

CSS Tables

Table Properties

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

- width
- height
- color
- font-size
- text-align

Table Border

Kredo

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

ID	Name	Age
1	John Doe	31
2	Peter Parker	26

Use the border property to add borders around the table and the cells.

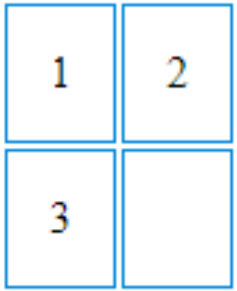
`1px` – width
`solid` – style
`black` – color

Empty Cells

Kredo

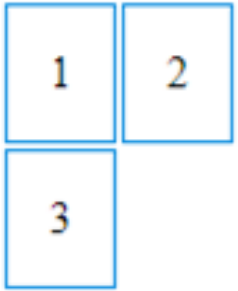
```
empty-cells: show;
```

To show the empty cells in the table.




```
empty-cells: hide;
```

To hide the empty cells in the table.

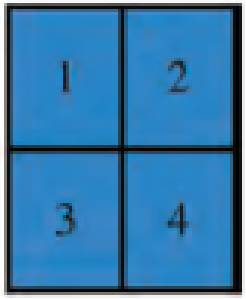


Border Collapse

Kredo

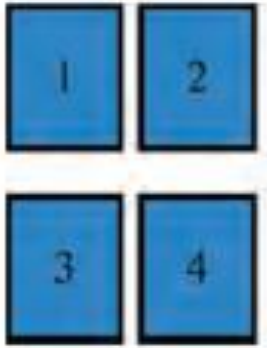
```
border-collapse: collapse;
```

Table borders are collapse into a single border.




```
border-collapse: separate;
```

Table borders are detached from each other.



Border Spacing

Kredo

```
border-spacing: 15px;
```

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

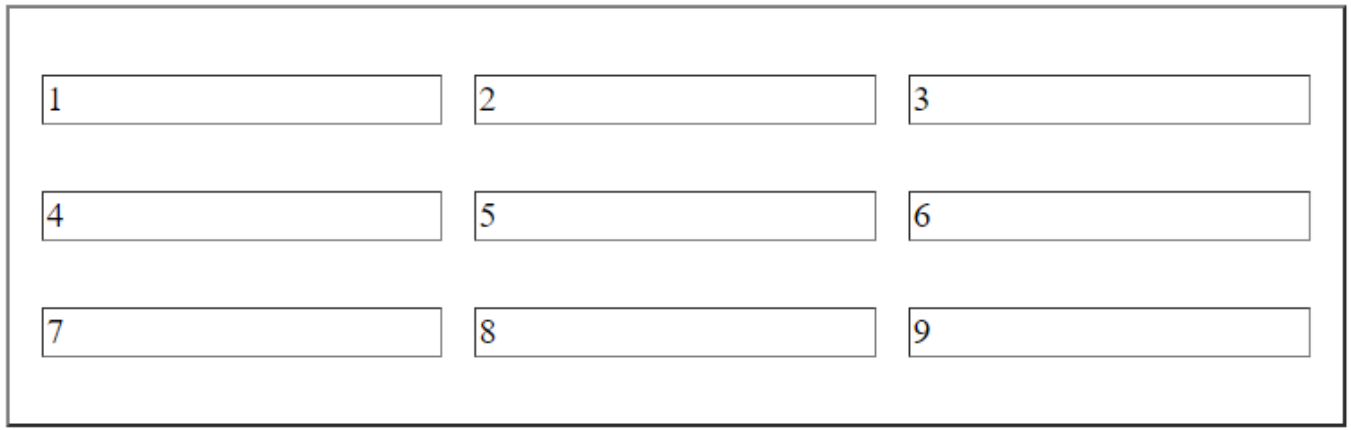
1	2	3
4	5	6
7	8	9

Border Spacing

Kredo

```
border-spacing: 15px 30px;
```

When using two values, the **first value** adjusts the **horizontal spacing** while the **second value** adjusts the **vertical spacing**.



CSS Activity 1

ID NO.	PHOTO	NAME	AGE	EMAIL	ADDRESS
1		Alice	26	alice@email.com	35553 KENAI SPUR HIGHWAY
2		Candice	26	candice@email.com	1616 GADSDEN HWY
3		Troy	26	N/A	35553 9256 PARKWAY E
4		Gab	26	N/A	AVALON AVE.
5		John	26	john@email.com	3006 E HIGHLAND

css-act1.html

use internal CSS

CSS Box Model and Background Color

CSS Background Color

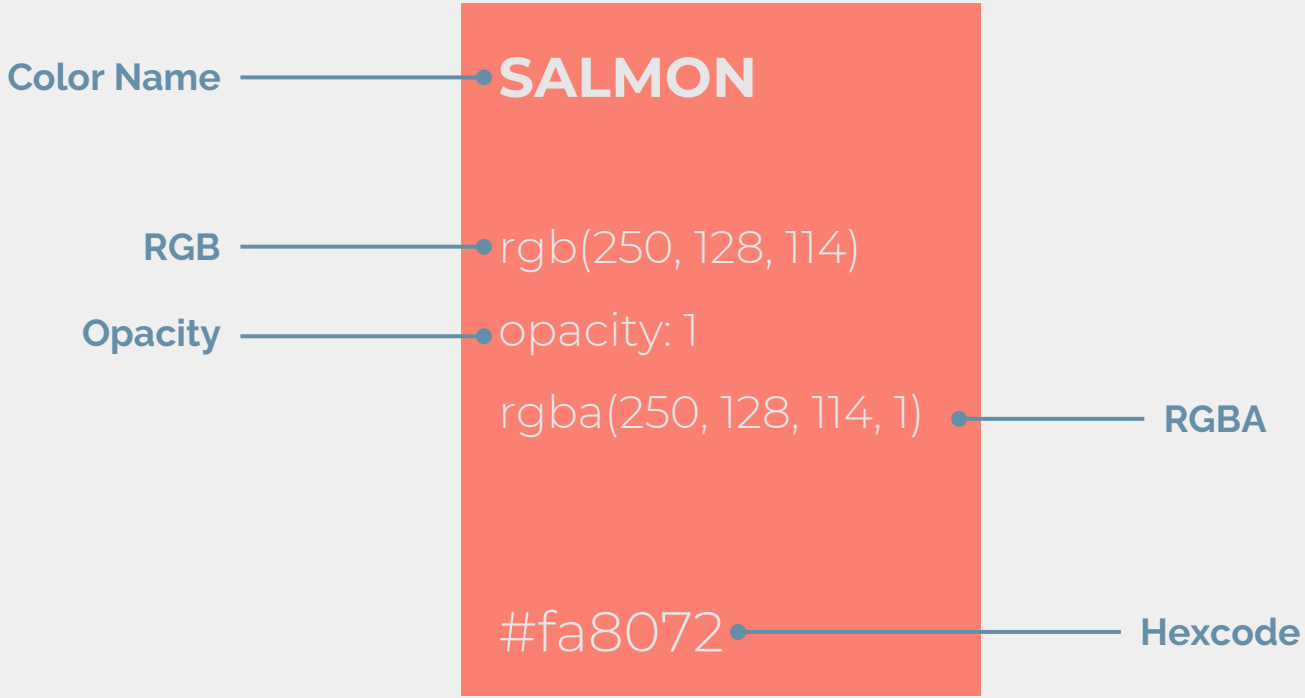
background-color

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

Example:

```
background-color: white;
```

CSS Background Color



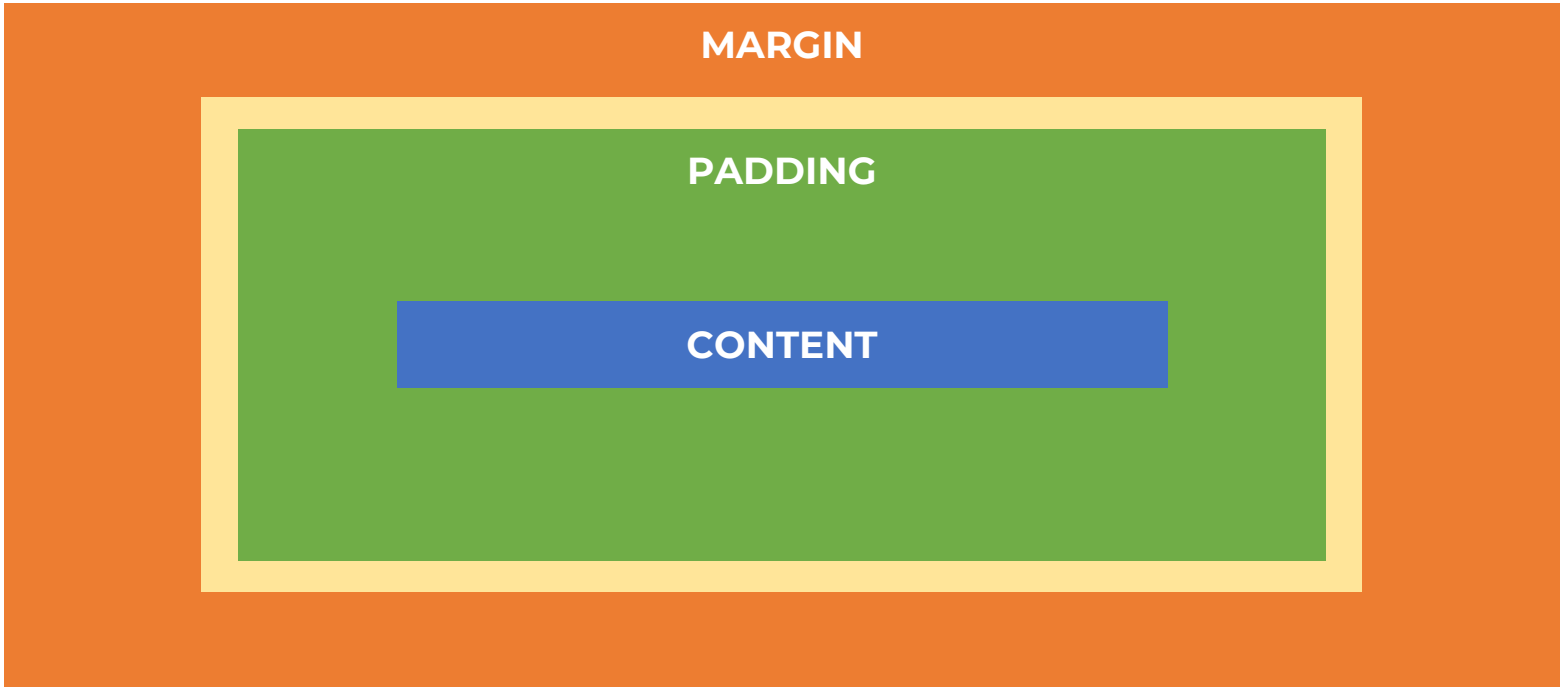
Height & Width

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

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

BOX MODEL

Kredo



BORDER

is the yellow box.

Margin

margin

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

Note:
Margin values are applied clockwise

Margin

Kredo

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

Kredo

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

Kredo

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.

Padding values are applied clockwise

Padding

Kredo

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

Kredo

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

Kredo

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 shorthand

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

Border

Kredo

border-width:
sets the width of the border.

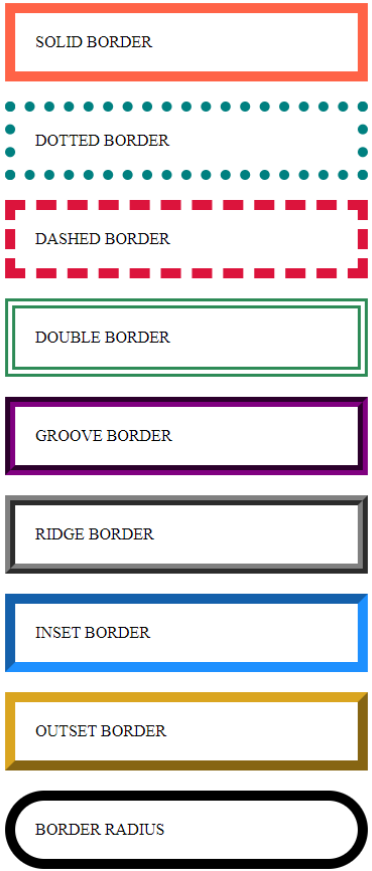
The value of the border-width can be: **px**

border-color:
sets the color of the border.

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

Border Style

```
.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;
}
```



Border

Kredo

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

Kredo

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 Shadow

box-shadow

The **box-shadow** property adds a shadow to an element

box-shadow shorthand

```
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 SIZING

Kredo

By default (*`box-sizing: content-box;`*), 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.

Centering Content

Kredo

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;

CSS Layouting

FLOATING

Kredo

`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

Kredo

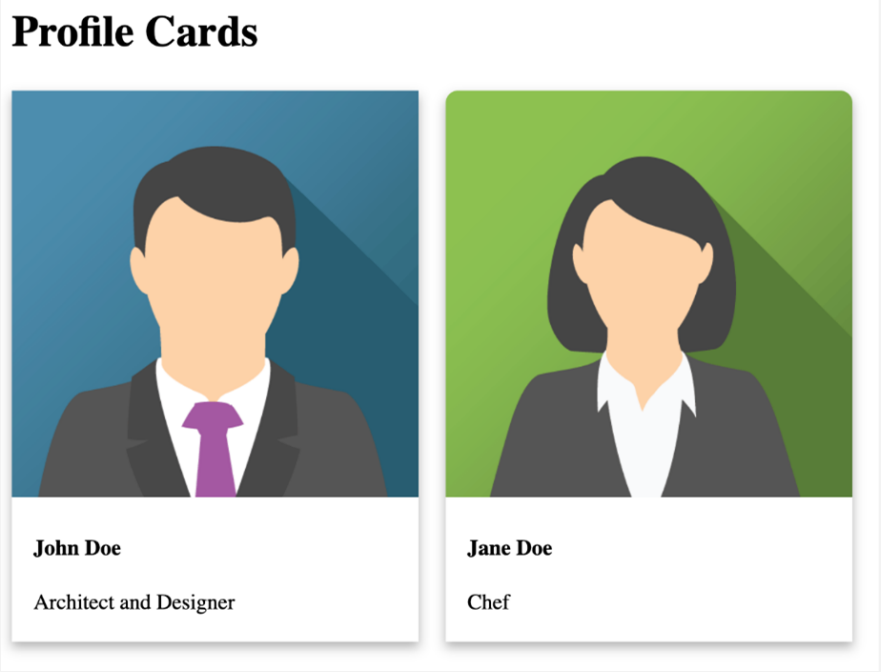
`clear: none;` - the element is not moved down to clear past floats.

`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.

CSS Activity 2



css-act2.html

POSITION

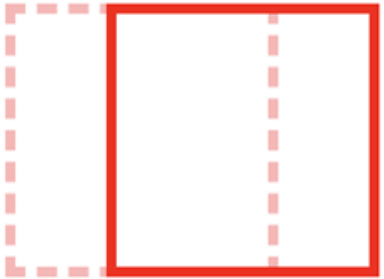
Kredo

`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 RELATIVE

```
.red-box {
    width: 50px;
    height: 50px;
    border: 2px solid red;
    position: relative;
    left: 20px;
}
```



The box will move 20px to the right relative to its original position.

POSITION

Kredo

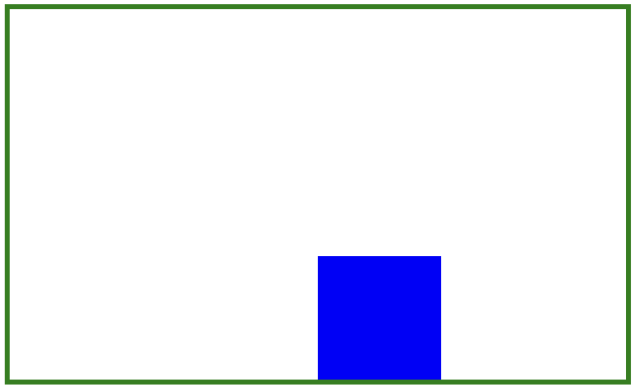
`position: absolute;` - the element is removed from the flow of the page and other elements will behave as if it's not even there. All the other positional properties will work on it.

`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.

POSITION ABSOLUTE

```
.parent {
    width: 250px;
    height: 150px;
    border: 2px solid green;
    position: relative;
}

.child {
    width: 50px;
    height: 50px;
    background-color: blue;
    position: absolute;
    left: 50%;
    bottom: 0;
}
```



The child will move by 50% from the left edge relative to the width of its parent.
There is no space at the bottom of the child box relative to its parent.

POSITION

Kredo

`top | bottom | left | right: % | px;` - are used with `position` to set the placement of an element. They only have an effect on positioned elements, which are elements with the position property set to anything other than static.

The element will move away from the chosen side when its value is positive, and towards it when the value is negative.

DISPLAY

Kredo

`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

Kredo

`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.

Z-INDEX

Kredo

`z-index: value;` - controls the vertical stacking order of elements that overlap. Like, layers in photoshop or in a cake.

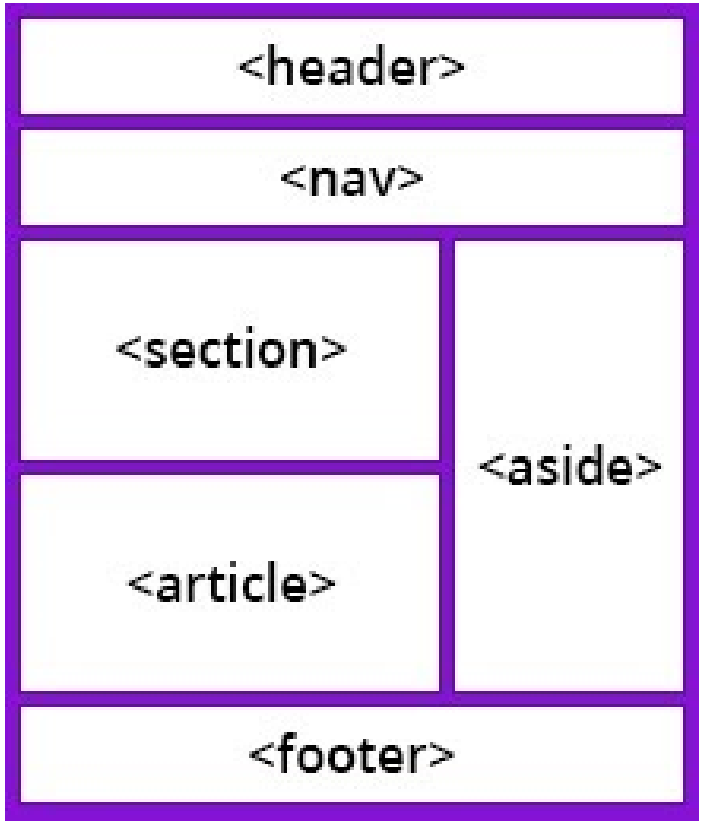
It only affects elements that have a position value other than static (the default).

Semantic Tags

are used to convey the structure of the document in a clear manner. They define the purpose of the element.

TAGS

Kredo



TAGS

Kredo

<header> - defines a header of your document. It is always visible for the users at the top part of the page.

<nav> - defines the space for the navigations links.

<section> - defines a separate section within a webpage. Has its own content; Considered as the "***Parent of articles"*** .

<article> - defines the article content. The sub-section.

<main> - defines the main content of a page.

TAGS

Kredo

<aside> - defines the content which will be set ***to the side.*** It is usually used for creating sidebars and side contents.

<footer> - defines the **footnote** of the web page or content. Can also be used to show the address and give out the reference links

Deployment

DEPLOYMENT

But first...

Register an account in **GitHub (github.com)**

DEPLOYMENT

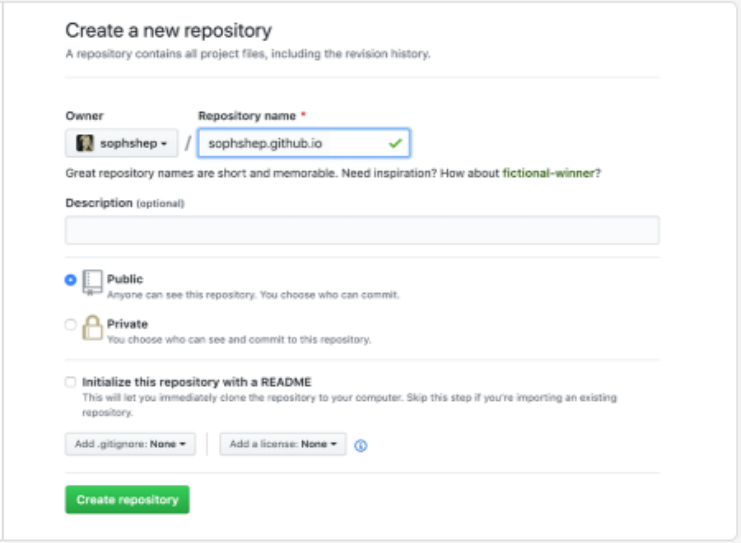
Kredo

1

Create a repository

Head over to GitHub and create a new repository named *username*.github.io, where *username* is your username (or organization name) on GitHub.

If the first part of the repository doesn't exactly match your username, it won't work, so make sure to get it right.



Note:

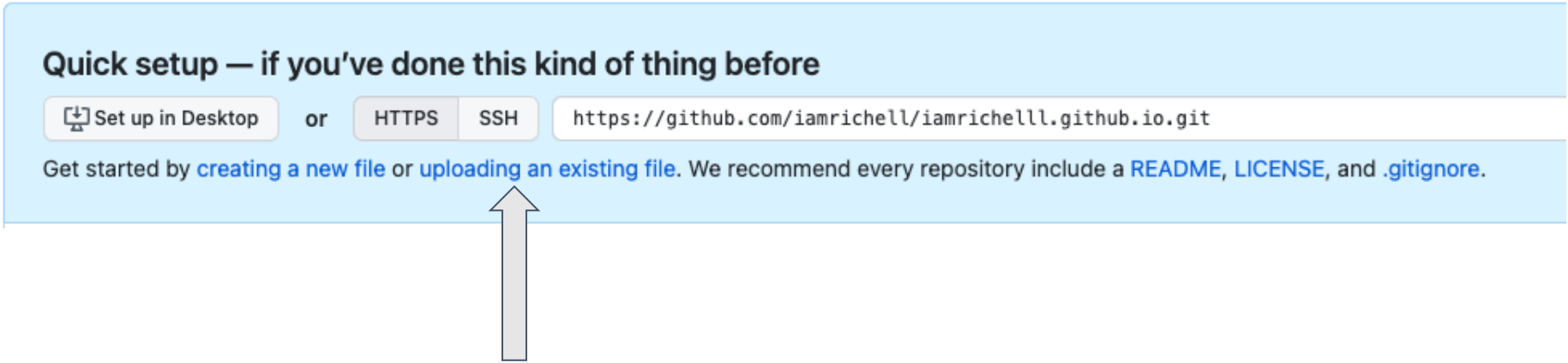
Repository name should be **username.github.io**

this should be the same as the account name

this should always be included in creating the repo else output will not be displayed as it is

DEPLOYMENT

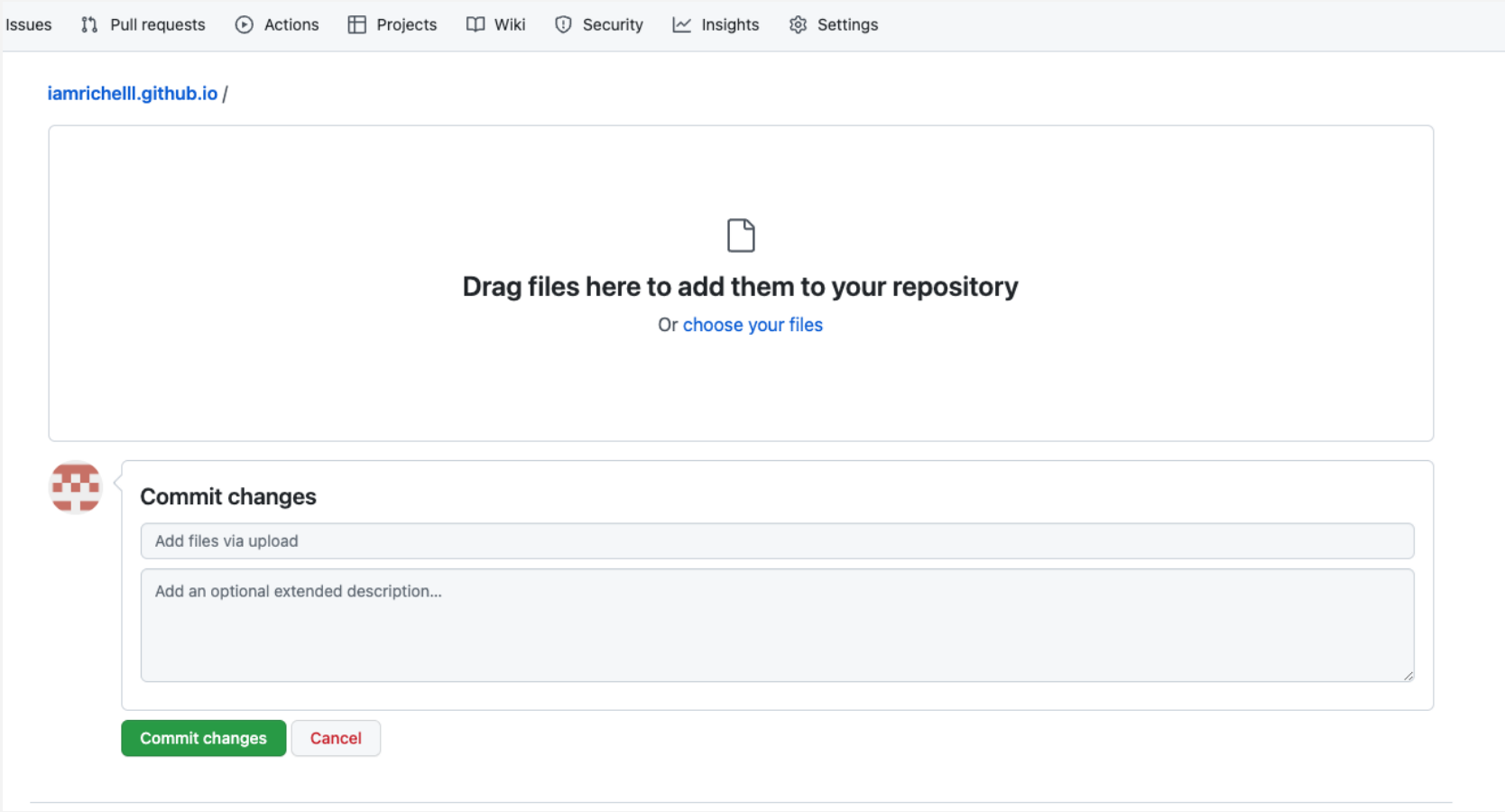
Kredo



In the next page, click **upload an existing file**

DEPLOYMENT

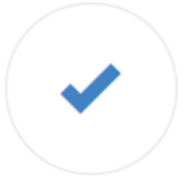
Kredo



DEPLOYMENT

Kredo

5

…and you're done!

Fire up a browser and go to **https://*username*.github.io**.