3_HTML_2

Agenda

- block and inline
  - div and class, id
  - span and styling
- inputs , labels
- forms and validation

Div vs span

span

- span : It is a tag that takes only the required space

```
<span>I am a span</span>
<span>I am a span also</span>
```

I am a span I am a span also

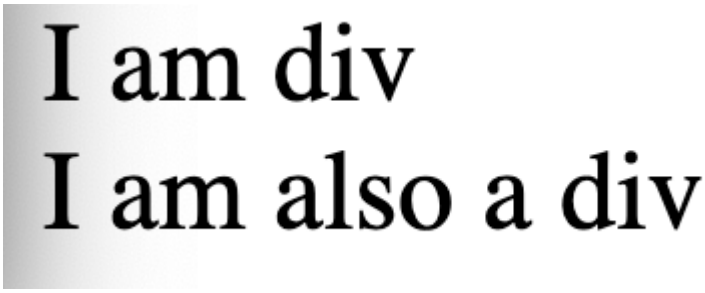
- **use case of span**: The `<span>` element is used for styling parts of inline content.

```
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Officia quod ullam dolores voluptatibus,

<span style="color: blue; font-weight: bold;">soluta tempora</span>

quos maxime quaerat aut totam consequuntur repudiandae ipsa neque sed molestiae fugiat placeat perspiciatis

officiis!

</p>
```

Div

- div individually: It is a tag that takes full width of the and allow next element to appear after it

```
<div>I am div</div>
<div>I am also a div</div>
```



use case of div

- wrapping up text -> for better code readability

```
    <div>

        <h1 >I am heading</h1>
        <h2>I am heading</h2>
        <!-- card level heading -->
        <h3>I am heading</h3>
        <h4>I am heading</h4>
        <h5>I am heading</h5>
        <h6>I am heading</h6>
    </div>

    <!-- paragraph -->
    <div >
        <p>Paragraph :dshjbfjhdsbfjhdbfhj</p>
        <p>Paragraph :dshjbfjhdsbfjhdbfhj</p>
        <p>Paragraph :dshjbfjhdsbfjhdbfhj</p>
        <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. In
eos delectus molestiae corporis vitae dolore.
            Eum
            deleniti labore nostrum? Tenetur quos mollitia natus magni
corporis similique exercitationem? Molestias,
```

```
            iste?
            Quidem?</p>
    </div>

    <!-- button -->
    <div >
        <button>I am a button</button>
        <button>I am a button2</button>
        <button>I am a button3</button>
    </div>

    ```


Problem Statement : let's say you have use case of wrapping similar type of elements -> like group of images.

```html

<div>
        <img src="https://picsum.photos/200/300" alt="random image" height="200px">
        <br>
        <!-- local file -->
        <img src="./car1.png" alt="car image" height="200px">
        <br>
 </div>

 <div>
        <img src="./cycle.png" alt="cycle image" height="100px">
        <br>
        <img src="./image.jpg" alt="flower image" height="300px">

    </div>
```

classes and id

in this case we have two html attributes that comes to rescue

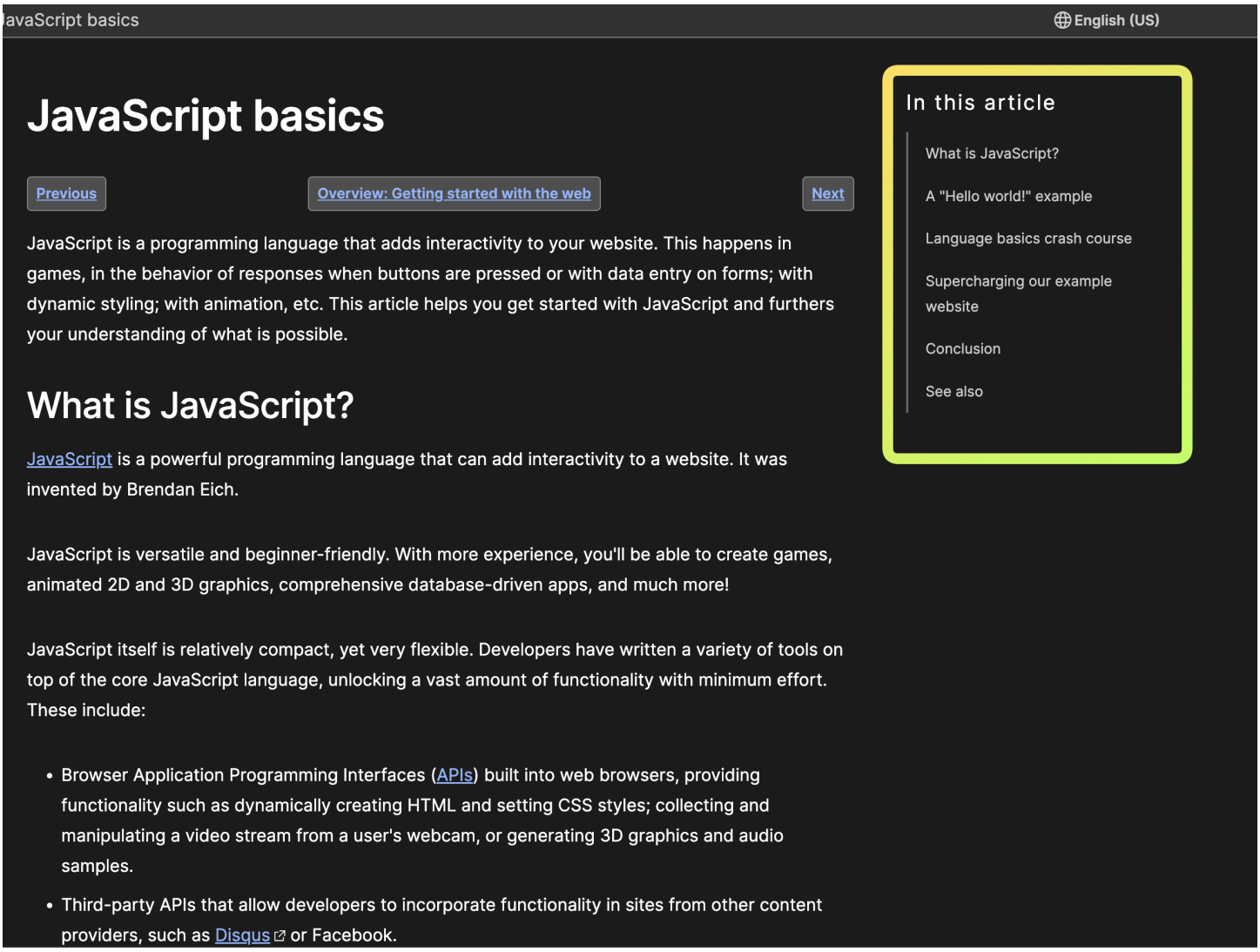
- classes -> to give that common name or group these two
- id -> to provide the unique name to a block of html code

let's use these two wrap it efficiently

```
<div id="first-set" class="images">
        <img src="https://picsum.photos/200/300" alt="random image"
height="200px">
        <br>
        <!-- local file -->
        <img src="./car1.png" alt="car image" height="200px">
        <br>
    </div>
    <div id="second-set" class="images">
        <img src="./cycle.png" alt="cycle image" height="100px">
        <br>
        <img src="./image.jpg" alt="flower image" height="300px">
    </div>
```

Practical use case of id

We have these kinds of navigation bar where once a user clicks on a particular link . they are navigated
to a particular section. let's see how this can be build easily using ids and links



Example

Assuming we have these `sections` and a on top that we have a `navbar` usually navbar `is a list of links so we will be using ul , li and anchor to achieve it` .

```
 <ul>
        <li><a href="">Section 1</a></li>
        <li><a href="">Section 2</a></li>
        <li><a href="">Section 3</a></li>
    </ul>
   <div>
    <h2>I am section-1</h2>
    <p>
        <!-- assume we have a lot of text here  -->
    </p>
</div>
```

```
<div>
    <h2 >I am section-2</h2>
    <p>
        <!-- assume we have a lot of text here  -->
    </p>
</div>


<div>
    <h2 >I am section 3</h2>
    <p>
        <!-- assume we have a lot of text here  -->
    </p>
</div>
```

You can easily achieve it by providing unique ids to each section and adding the similar ids to anchors .

```
 <ul>
        <li><a href="#section-1">Section 1</a></li>
        <li><a href="#section-2">Section 2</a></li>
        <li><a href="#section-3">Section 3</a></li>
    </ul>
   <div id="section-1">
    <h2>I am section-1</h2>
    <p>
        <!-- assume we have a lot of text here  -->
    </p>
</div>
<div id="section-2">
    <h2 >I am section-2</h2>
    <p>
        <!-- assume we have a lot of text here  -->
    </p>
</div>
<div id="section-3">
    <h2 >I am section 3</h2>
    <p>
        <!-- assume we have a lot of text here  -->
    </p>
```

```
</div>
```

Practical use case of classes

Usually there is common styling given to multiple elements -> to reuse it -> you can give them common class

Example

```
<style>

    .section-firstline {
        color: blue;
        font-weight: bold;
    }
</style>

<body>
<h1>Blog Title</h1>
   <div>
    <h2>I am section-1</h2>
    <p>
    <span class="section-firstline">
                Welcome message with paragraphs and inline spans.
            </span>

    </p>
</div>
<div >
    <h2 >I am section-2</h2>
    <p>
      <span class="section-firstline">
                Welcome message with paragraphs and inline spans.
            </span>
    </p>
</div>
<div >
    <h2 >I am section 3</h2>
    <p>
```

```
        <span class="section-firstline">
                Welcome message with paragraphs and inline spans.
            </span>
    </p>
</div>

</body>
```

Block and inline elements

Inline elements :

They only take the required width and allow the next inline element to be placed beside them if space is available

Example

Inline elements -> span , imgs, btns, anchors

Block level elements

They only take the `full width of the parent` and next element will be placed on the next line .

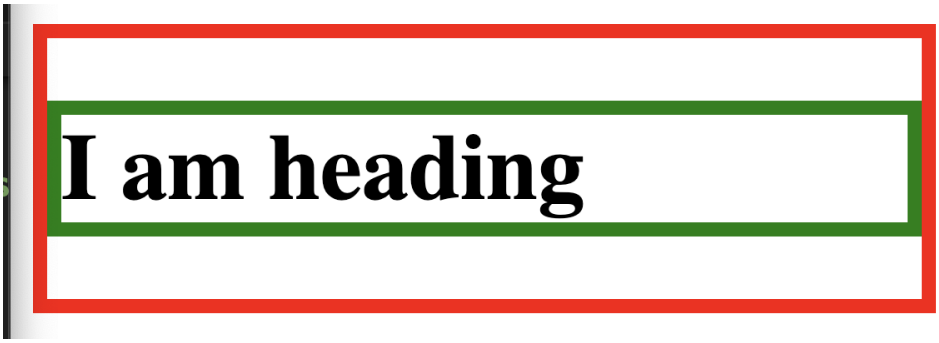
let's understand it in greater depth with help of an example

```
<div class="heading" style="border:5px solid red">
        <!-- page title  -->
        <h1 style="border: 5px solid green;">I am heading</h1>
    </div>
```

You can see the outer div that wraps our h1 -> h1 has width equal to the div shown by the border

now if i will reduce the width of div h1's width will also decrease

```
<div class="heading" style=" width :300px ;border:5px solid red">
        <!-- page title  -->
        <h1 style="border: 5px solid green;">I am heading</h1>
        <!-- section level heading -->
    </div>
```



HTML Form

1. Text Input

The `<input type="text">` element is used to create a single-line text input field.

Example:

```
<input type="text" id="name" placeholder="Enter your name">
```

2. Label

The `<label>` element is used to define a label for an `<input>` element, improving accessibility by associating the label with the input.

Example:

```
<label for="name">Name:</label>
<input type="text" id="name">
```

3. `for` Attribute

The `for` attribute in a `<label>` element binds it to an `<input>` element with a matching `id` attribute.

Example:

```
<label for="nickname">Nick Name:</label>
<input type="text" id="nickname">
```

4. `id` Attribute

The `id` attribute uniquely identifies an `<input>` element on a page, enabling labels and JavaScript to reference it.

Example:

```
<input type="text" id="nickname">
```

5. Placeholder

The `placeholder` attribute provides a hint to the user about what to enter in the input field.

Example:

```
<input type="text" id="name" placeholder="Enter your name">
```

6. Other Input Types

HTML provides various input types to handle different types of data.

Example:

- **Number Input:**

  ```
  <label for="my-num">Number: </label>
  <input type="number" id="my-num" min="10" max="100" step="4"
  required>
  ```

- **Email Input:**

```
<label for="email">Email: </label>
<input type="email" id="email" value="ravi@gmail.com">
```

- **Password Input:**

```
<label for="password">Password: </label>
<input type="password" id="password" minlength="8" maxlength="14">
```

7. Use Case: Form for Email Validation

A form with an email input ensures the user provides a valid email address.

Example:

```
<form action="/submit">
    <label for="email">Email: </label>
    <input type="email" id="email" required>
    <input type="submit">
</form>
```

8. Validation Attributes

- **minlength**: Sets the minimum number of characters for text inputs.
- **maxlength**: Sets the maximum number of characters for text inputs.
- **min**: Sets the minimum value for numeric inputs.
- **max**: Sets the maximum value for numeric inputs.
- **step**: Specifies the increment for numeric inputs.

Example:

```
<input type="password" id="password" minlength="8" maxlength="14">
<input type="number" id="my-num" min="10" max="100" step="4">
```

9. Properties

- **value**: Sets the initial value of the input.
- **readonly**: Makes the input field uneditable while still submitting the value.

Example:

```
<input type="text" id="nickname" readonly value="some value">
```

Complete Form Example

```
<form action="/submit">
    <!-- Text Input -->
    <label for="nickname">Nick Name:</label>
    <input type="text" id="nickname" readonly value="some value">

    <label for="name">Name: </label>
    <input type="text" id="name" placeholder="Enter your name">
    <br><br>

    <!-- Number Input -->
    <label for="my-num">Number: </label>
    <input type="number" id="my-num" min="10" max="100" step="4"
required>
    <br><br>

    <!-- Email Input -->
    <label for="email">Email: </label>
    <input type="email" id="email" value="ravi@gmail.com" required>
    <br><br>

    <!-- Password Input -->
    <label for="password">Password: </label>
    <input type="password" id="password" minlength="8" maxlength="14"
required>
    <br><br>

    <input type="submit">
</form>
```

Additional Input Types

Date Input:

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

- **Purpose**: Allows users to select a date from a date picker.

Time Input:

```
<label for="time">Time:</label>
<input type="time" id="time">
```

- **Purpose**: Allows users to select a time.

Color Input:

```
<label for="color">Color:</label>
<input type="color" id="color">
```

- **Purpose**: Opens a color picker to select a color.

File Input:

```
<label for="file">File:</label>
<input type="file" id="file" name="file">
```

- **Purpose**: Allows users to select a file from their device.

Example HTML for Additional Inputs:

```
<form action="/submit">
    <!-- Date Input -->
    <label for="date">Date:</label>
    <input type="date" id="date">
    <br><br>

    <!-- Time Input -->
    <label for="time">Time:</label>
    <input type="time" id="time">
    <br><br>
```

```
    <!-- Color Input -->
    <label for="color">Color:</label>
    <input type="color" id="color">
    <br><br>

    <!-- File Input -->
    <label for="file">File:</label>
    <input type="file" id="file" name="file">
    <br><br>

    <!-- Search vs Text Input -->
    <label for="search">Search:</label>
    <input type="search" id="search">
    <br><br>

    <label for="name">Name:</label>
    <input type="text" id="name">
    <br><br>
</form>
```

Comparison: Input Type `text` vs `search`

Text Input:

```
<label for="name">Name:</label>
<input type="text" id="name">
```

- **Purpose**: General text input for short data such as names or addresses.
- **Features**: Basic text input.
- **Usage**: Suitable for most single-line text entry needs.

Search Input:

```
<label for="search">Search:</label>
<input type="search" id="search">
```

- **Purpose**: Specifically designed for search queries.
- **Features**: May include additional browser features like a clear button to reset the input, though these features depend on the browser.

- **Usage**: Ideal for search fields where users enter search terms.