4_HTML_3

Agenda

- Advanced forms
  - pattern matching, invalid cue
  - other tags
- Multimedia Elements -> video and audio tags
- iframes
  - Embedded content -> Youtube ,linkedin, Google Maps

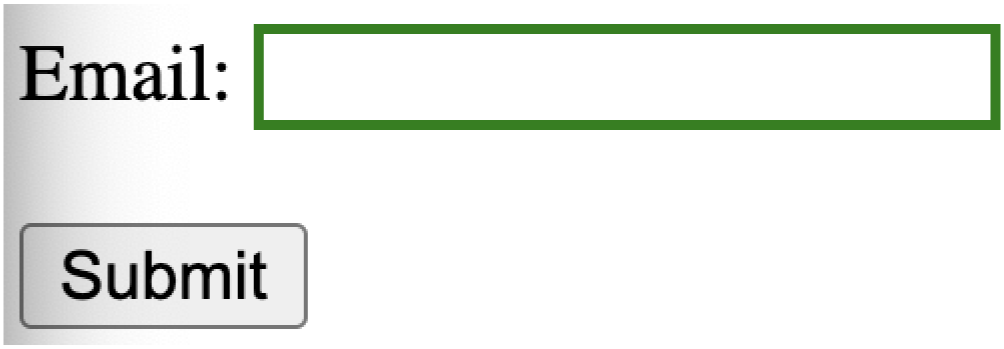
invalid and valid cue

let's say we have a form like this where we have input type email . Input type email checks for whether you have @ and .com/edu(top level domain) in the email .

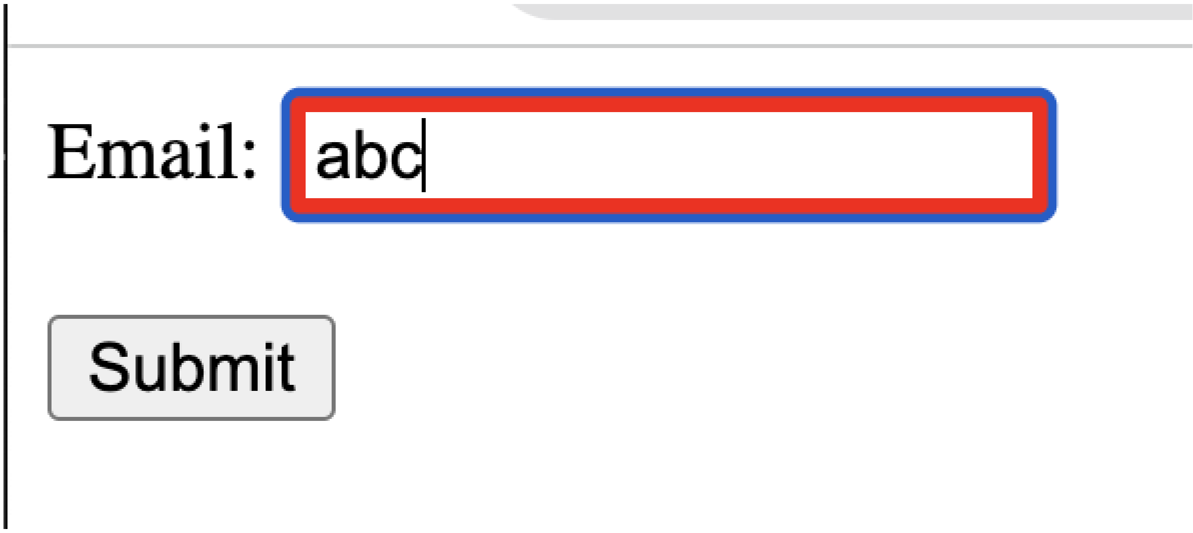
~~~
   <form>
        <label for="email">Email: </label>
        <input type="email" id="email">
        <br>
        <br>
        <button type="submit">Submit</button>
    </form>
    ```

Let's see how we can add visual cues to our input . We can use
`input:valid` and `input:invalid` selectors for it .
```html
<style>
    input:valid {
        border: 2px solid green;
    }

    input:invalid {
        border: 5px solid red;
    }
</style>
~~~

It makes the border of input green when valid and red when invalid

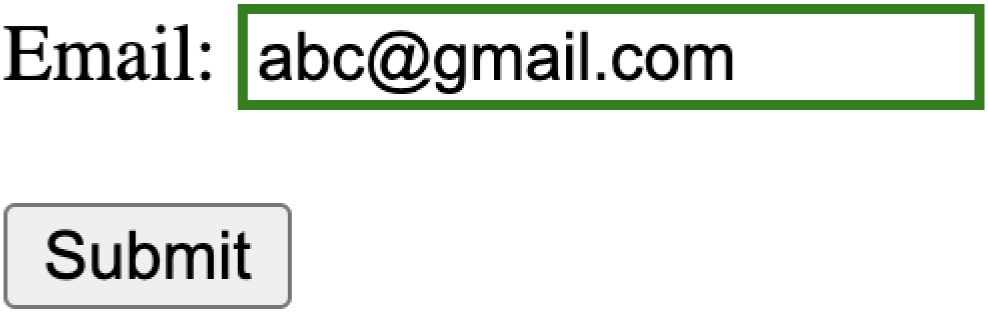
valid state


Email:

invalid state


Email: abc

valid state


Email: abc@gmail.com

Pattern matching

In inputs there are inbuilt validators -> min, max , min length , maxlength .
Let's say if you want to add your custom validation . patterns are there to help .
let's understand how pattern works with an example

1. Credit Card Number

Rule: Credit card number should be 16 digits long.

```html
<label for="creditcard">Credit Card Number:</label>

<input type="text" id="creditcard"  pattern="\d{16}" title="Please enter a valid 16-digit credit card number.">
```

- `<input type="text" id="creditcard" >`: This line creates a text input field with the id and name set to "creditcard".

Now, let's dive into the `pattern` and `title` attributes:

- `pattern="\d{16}"`: This attribute uses a regular expression (`\d{16}`) to enforce that the input must be a sequence of exactly 16 digits. This is a basic validation rule for credit card numbers, which are typically 16 digits long.
- `title="Please enter a valid 16-digit credit card number."`: This attribute sets a tooltip that appears when the user hovers over the input field. If the user's input doesn't match the pattern, this tooltip text will be displayed to guide the user to correct their input.

These two attributes work together to enforce and guide the user to input a valid 16-digit credit card number

### rules for patterns

Certainly! The `pattern` attribute in HTML is a powerful tool for form validation, using regular expressions (regex) to define the format of acceptable input values. Here are all the useful rules and details for using the `pattern` attribute effectively:

Basic Syntax and Usage

```html
<input type="text" pattern="regular_expression">
```

- **Type of Input**: The `pattern` attribute can be used with various input types, including `text`, `search`, `url`, `tel`, and `email`.

Regular Expression Basics

- **Character Classes**: Define a set of characters.
    - `[a]`: Matches any one of the characters a.
    - `[a-z]`: Matches any one lowercase letter from a to z.
    - `[A-Z]`: Matches any one uppercase letter from A to Z.
    - `[0-9]`: Matches any one digit from 0 to 9.
    - `[a-zA-Z0-9]`: Matches any one alphanumeric character.
- **Quantifiers**: Specify the number of occurrences.
    - `*`: Matches 0 or more occurrences.
    - `+`: Matches 1 or more occurrences.
    - `?`: Matches 0 or 1 occurrence.
    - `{n}`: Matches exactly n occurrences.
    - `{n,}`: Matches n or more occurrences.
    - `{n,m}`: Matches between n and m occurrences.
    - `\d`: Matches any digit (equivalent to `[0-9]`).

Custom Validation Messages

Using the `title` attribute, you can provide custom validation messages that will be displayed if the input does not match the specified pattern.

```html
<input type="text" pattern="[A-Za-z]{3}" title="Please enter exactly three letters.">
```

Real world use case of pattern matching

1. Date (YYYY-MM-DD)

Rule: The date must be in the format `YYYY-MM-DD`.

Explanation:

- `\d{4}`: Matches exactly 4 digits (year).
- `-`: Matches a hyphen.
- `\d{2}`: Matches exactly 2 digits (month).
- `-`: Matches another hyphen.

- `\d{2}`: Matches exactly 2 digits (day).

Code Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        input:invalid {
            border: 2px solid red;
        }
    </style>
</head>
<body>
    <form>
        <label for="date">Date:</label>
        <input type="text" id="date" pattern="\d{4}-\d{2}-\d{2}"
title="Please enter a date in the format YYYY-MM-DD." required>
        <br><br>

        <!-- More fields can be added here -->
    </form>
</body>
</html>
```

2. Credit Card Number

Rule: Credit card number should be 16 digits long.

```

<label for="creditcard">Credit Card Number:</label>

<input type="text" id="creditcard" name="creditcard" pattern="\d{16}"
title="Please enter a valid 16-digit credit card number.">
```

3. Indian Phone Number

Rule: The phone number should be a 10-digit number starting with a digit from 6 to 9.

Explanation:

- `[6-9]`: The first digit must be 6, 7, 8, or 9.
- `\d{9}`: Matches exactly 9 more digits, making a total of 10 digits.

Code Example:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        input:invalid {
            border: 2px solid red;
        }
    </style>
<body>
    <form>
        <label for="phone">Phone Number:</label>
        <input type="text" id="phone"  pattern="[6-9]\d{9}" title="Please enter a valid 10-digit Indian phone number starting with 6, 7, 8, or 9." required>

        <br><br>

    </form>
</body>
</html>
```

Absolutely! Here are examples for validating IP addresses, dates, and URLs using the `pattern` attribute, including CSS to give feedback using a red border for invalid inputs.

4. URL

Rule: The URL must start with `http://` or `https://`.

Explanation:

- `https?://`: Matches `http://` or `https://` (the `?` makes the `s` optional).
- `.+`: Matches one or more of any character after the initial URL scheme.

Code Example:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        input:invalid {
            border: 2px solid red;
        }
    </style>
</head>
<body>
    <form>
        <label for="url">Website:</label>
        <input type="text" id="url"  pattern="https?://.+" title="Please enter a valid URL starting with http:// or https://" required>
        <br><br>

    </form>
</body>
</html>
```

5. Alphanumeric Username

Rule: The username should be 3 to 15 characters long and contain only letters and numbers.

Explanation:

- `[A-Za-z0-9]`: Matches any letter (uppercase or lowercase) or digit.
- `{3,15}`: Specifies that the preceding character class must occur between 3 and 15 times.

Code Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        input:invalid {
            border: 2px solid red;
        }
    </style>
</head>
<body>
    <form>
        <label for="username">Username:</label>
        <input type="text" id="username" pattern="[A-Za-z0-9]{3,15}"
title="Username should be 3 to 15 characters long and contain only
letters and numbers." required>
        <br><br>

    </form>
</body>
</html>
```

Note : `You don't need to cram these patterns . Knowing that regular expression exists is more than enough . You can just ask chat-gpt for regualr expression that satisyies your needs`

Form and input tags part-3

1. Text Inputs (Including Textarea)

Explanation: Start with basic text inputs, which are fundamental for collecting user input.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>HTML Form Tutorial</title>
</head>
```

```
<body>
    <form>
        <!-- Text Input -->
        <label for="username">Username:</label>
        <input type="text" id="username" name="username"
placeholder="Enter your username">
        <br><br>

        <!-- Textarea -->
        <label for="comments">Comments:</label>
        <textarea id="comments" name="comments" rows="4" cols="50"
placeholder="Enter your comments here..."></textarea>
        <br><br>
    </form>
</body>
</html>
```

2. Range Input

Explanation: Cover the `<input type="range">` element, which allows users to select a numeric value from a range using a slider.

```
<!-- Continuing from previous code -->
<!-- Range Input -->
<label for="volume">Volume:</label>
<input type="range" id="volume" name="volume" min="0" max="100"
value="50">
<br><br>
```

3. Select Options (Dropdowns)

Explanation: Move on to `<select>` elements to provide a dropdown selection interface for users when there are multiple options.

```
<!-- Continuing from previous code -->
<!-- Select Options (Dropdown) -->
<label for="country">Country:</label>
<select id="country" name="country">
    <option value="india">India</option>
    <option value="usa">USA</option>
    <option value="uk">UK</option>
</select>
<br><br>
```

2. Radio Buttons and Checkboxes

Explanation: Next, introduce radio buttons and checkboxes for options that require a binary or multiple-choice selection.

```
<!-- Continuing from previous code -->
<!-- Radio Buttons -->
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label><br>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label><br>

<!-- Checkboxes -->
<input type="checkbox" id="option1" name="option1" value="option1">
<label for="option1">Option 1</label><br>
<input type="checkbox" id="option2" name="option2" value="option2">
<label for="option2">Option 2</label><br>
```

4. Fieldset and Legend

Explanation: Introduce `<fieldset>` and `<legend>` to group related form elements together and provide a title or caption for the group.

```
<!-- Continuing from previous code -->
<!-- Fieldset and Legend -->
<fieldset>
    <legend>Subscription Options</legend>
    <input type="checkbox" id="newsletter" name="newsletter"
value="newsletter">
    <label for="newsletter">Subscribe to Newsletter</label><br>
    <input type="checkbox" id="updates" name="updates" value="updates">
    <label for="updates">Receive Updates</label><br>
</fieldset>
<br><br>
```

6. Selected and Checked Attributes

Explanation: Explain the `selected` attribute for preselecting options in dropdowns and the `checked` attribute for preselecting checkboxes and radio buttons.

```html
<!-- Continuing from previous code -->
<!-- Selected and Checked Attributes -->
<label for="defaultOption">Default Option:</label>
<select id="defaultOption" name="defaultOption">
    <option value="option1">Option 1</option>
    <option value="option2" selected>Option 2 (Selected by Default)
</option>
</select>
<br><br>

<input type="checkbox" id="preselectedCheckbox"
name="preselectedCheckbox" value="preselectedCheckbox" checked>
<label for="preselectedCheckbox">Preselected Checkbox (Checked by
Default)</label><br>
```

7. Reset Button

Explanation: Finally, cover the `<input type="reset">` element, which allows users to reset all form fields to their default values.

```html
<!-- Continuing from previous code -->
<!-- Reset Button -->
<input type="reset" value="Reset Form">
```

Multimedia tags -> audio & videos

Sure! Let's go step-by-step to build the `<video>` and `<audio>` elements with their attributes, explaining each part as we go.

Embedding a Video with Attributes

1. **Basic Video Embedding**

Start with the basic `<video>` element and source.

```
<h2>Video and audio</h2>
<figure>
    <video>
    // path -> where video is stored
        <source src="../video.mp4">
    </video>
    <figcaption>Water fall video</figcaption>
</figure>
```

Explanation:

- `<video>`: The container for the video.
- `<source src="../video.mp4">`: Specifies the video file.

2. Adding Controls

Add the `controls` attribute to provide video controls.

```
<figure>
    <video controls>
        <source src="../video.mp4">
    </video>
    <figcaption>Water fall video</figcaption>
</figure>
```

Explanation:

- `controls`: Adds playback controls (play, pause, volume, etc.) to the video.

3. Setting Height

Set the height of the video.

```
<figure>
    <video height="200" controls>
        <source src="../video.mp4">
    </video>
    <figcaption>Water fall video</figcaption>
</figure>
```

Explanation:

- `height="200"`: Sets the height of the video to 200 pixels.

4. Adding Muted and Autoplay

Add `muted` and `autoplay` attributes.

```
<figure>
    <video height="200" controls muted autoplay loop>
        <source src="../video.mp4">
    </video>
    <figcaption>Water fall video</figcaption>
</figure>
```

Explanation:

- `muted`: Mutes the video by default.
- `autoplay`: Starts playing the video as soon as it is loaded. autoplay in case of video only works when video is muted by default
- `loop`: it keep playing video on loop

Embedding Audio with Attributes

1. Basic Audio Embedding

Start with the basic `<audio>` element and source.

```
<h2>Music controls</h2>
<figure>
    <audio>
    // path where audio is stored
        <source src="./audio.mp3">
    </audio>
    <figcaption>Drum music</figcaption>
</figure>
```

Explanation:

- `<audio>`: The container for the audio.
- `<source src="./audio.mp3">`: Specifies the audio file.

2. ### Adding Controls

Add the `controls` attribute to provide audio controls.

```
<figure>
    <audio controls>
        <source src="./audio.mp3">
    </audio>
    <figcaption>Drum music</figcaption>
</figure>
```

Explanation:

- `controls`: Adds playback controls (play, pause, volume, etc.) to the audio.

3. ### Adding Autoplay and Muted

Add `autoplay` and `muted` attributes.

```
<figure>
    <audio controls autoplay muted>
        <source src="./audio.mp3">
    </audio>
    <figcaption>Drum music</figcaption>
</figure>
```

Explanation:

- `autoplay`: Starts playing the audio as soon as it is loaded.
- `muted`: Mutes the audio by default.

Full Example

Here's the final example incorporating all elements and attributes:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Multimedia Example</title>
</head>
<body>
    <h2>Video and audio</h2>
    <figure>
        <video height="200" controls muted autoplay>
            <source src="../video.mp4">
        </video>
        <figcaption>Water fall video</figcaption>
    </figure>

    <h2>Music controls</h2>
    <figure>
        <audio controls autoplay muted>
            <source src="./audio.mp3">
        </audio>
        <figcaption>Drum music</figcaption>
    </figure>
</body>
</html>
```

What is an `<iframe>`?

An `<iframe>` (short for inline frame) is an HTML element that allows you to embed another HTML document within the current document. This can be useful for embedding content like videos, maps, or other web pages.

Basic Structure of an `<iframe>`

Here's the basic structure of an `<iframe>` element:

```
<iframe src="URL"></iframe>
```

Most Used Attributes

1. `src`

- **Description**: Specifies the URL of the page to be embedded.
- **Example**:

```html
<iframe src="https://www.example.com"></iframe>
```

2. `width` and `height`

- **Description**: Define the size of the iframe.
- **Example**:

```html
<iframe src="https://www.example.com" width="600" height="400">
</iframe>
```

3. `allow`

- **Description**: Specifies a feature policy for the iframe.
- **Example**:

```html
<iframe src="https://www.youtube.com" allow="camera; microphone">
</iframe>
```

4. `allowfullscreen`

- **Description**: Allows the iframe to be displayed in fullscreen mode.
- **Example**:

```html
<iframe src="https://www.youtube.com" allowfullscreen></iframe>
```

Example with Attributes

Here's an example that includes these attributes:

```html
<!DOCTYPE html>
<html lang="en">
<head>
```

```html
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Iframe Example</title>
</head>
<body>
    <h1>Iframe Example</h1>
    <iframe src="https://www.youtube.com"
            width="600"
            height="400"
            allow="camera; microphone"
            allowfullscreen>
    </iframe>
</body>
</html>
```

Explanation

- **src**: Embeds the content from "*https://www.youtube.com*".
- **width and height**: Set the dimensions of the iframe to 600x400 pixels.
- **allow**: Permits the iframe to use the camera and microphone.
- **allowfullscreen**: Enables fullscreen mode for the iframe.

Usage Scenarios

- **Embedding Youtube Videos**:

  ```html
  <iframe src="https://www.youtube.com/embed/dQw4w9WgXcQ" width="560" height="315" allowfullscreen></iframe>
  ```

- **Embedding Maps**:

  ```html
  <iframe src="https://www.google.com/maps/embed?pb=YOUR_MAP_EMBED_CODE" width="600" height="450" style="border:0;" allowfullscreen></iframe>
  ```

- **Embedding Other Webpages**:

```
<iframe src="https://www.wikipedia.org" width="100%" height="500">
</iframe>
```