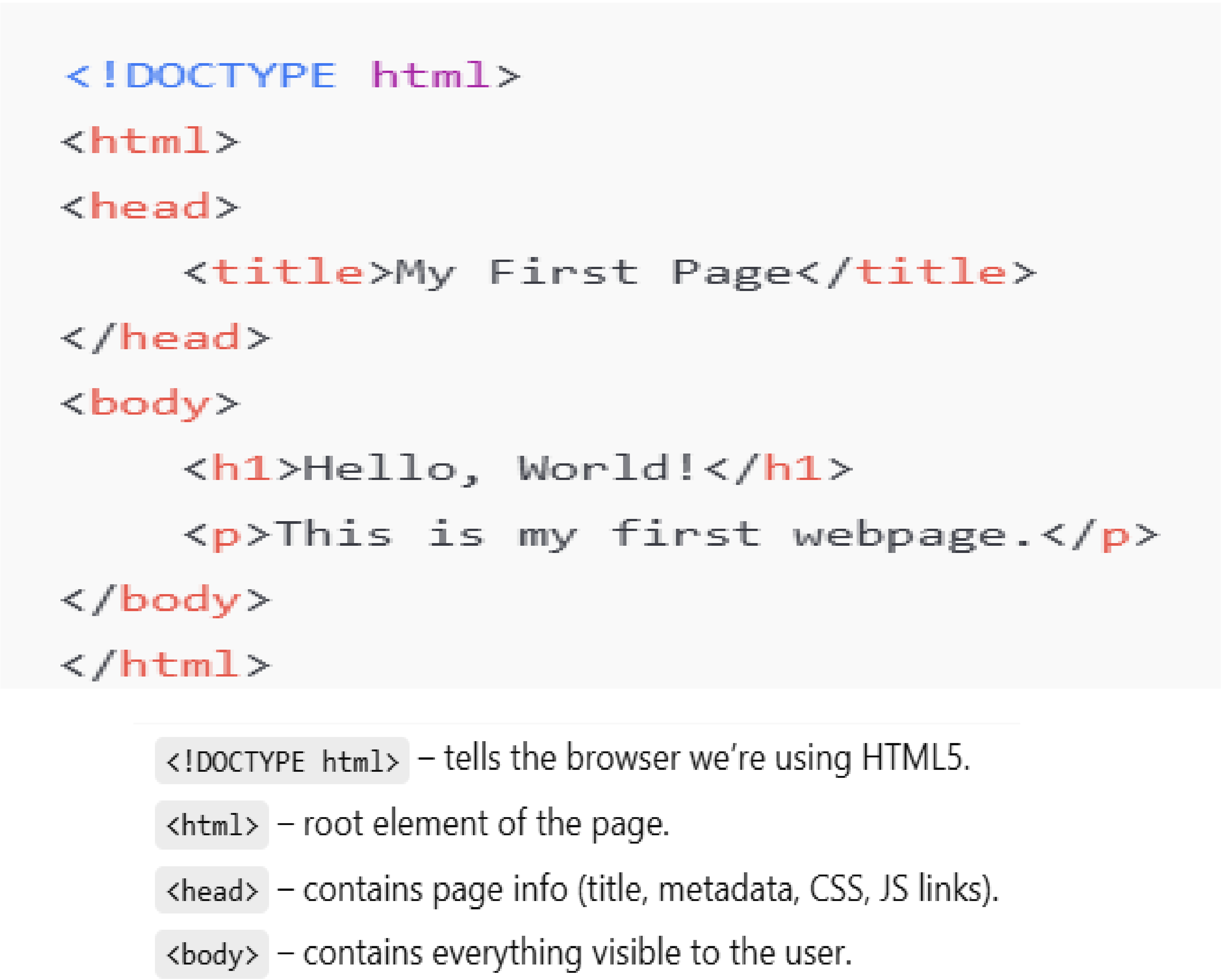
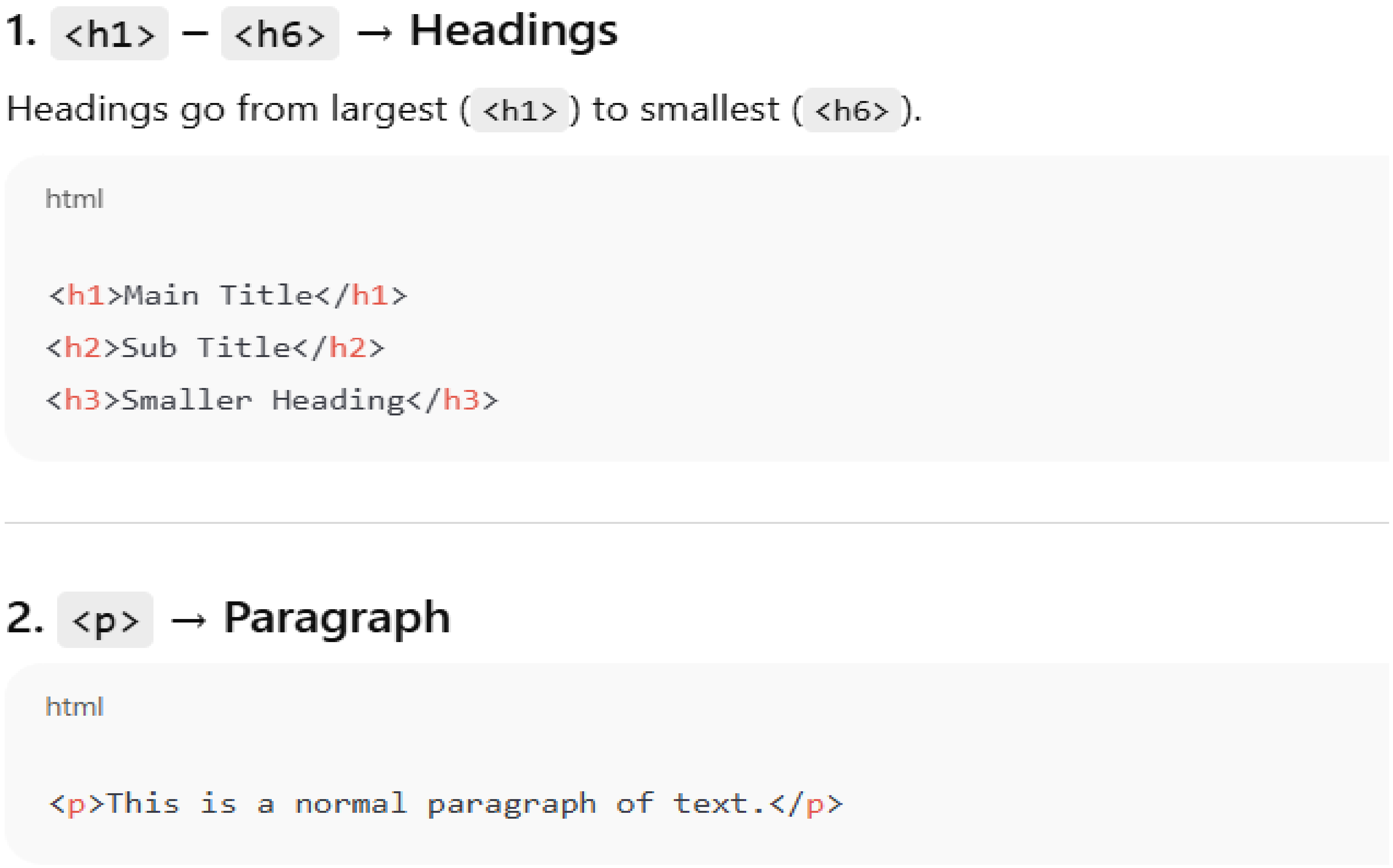


```html
<!DOCTYPE html>
<html>
<head>
    <title>My First Page</title>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>This is my first webpage.</p>
</body>
</html>
```

- `<!DOCTYPE html>` – tells the browser we're using HTML5.
- `<html>` – root element of the page.
- `<head>` – contains page info (title, metadata, CSS, JS links).
- `<body>` – contains everything visible to the user.



## 1. `<h1>` – `<h6>` → Headings

Headings go from largest (`<h1>`) to smallest (`<h6>`).

html

```html
<h1>Main Title</h1>
<h2>Sub Title</h2>
<h3>Smaller Heading</h3>
```

---

## 2. `<p>` → Paragraph

html

```html
<p>This is a normal paragraph of text.</p>
```

### 3. `<a>` → Link

```html
<a href="https://example.com">Visit Example</a>
```

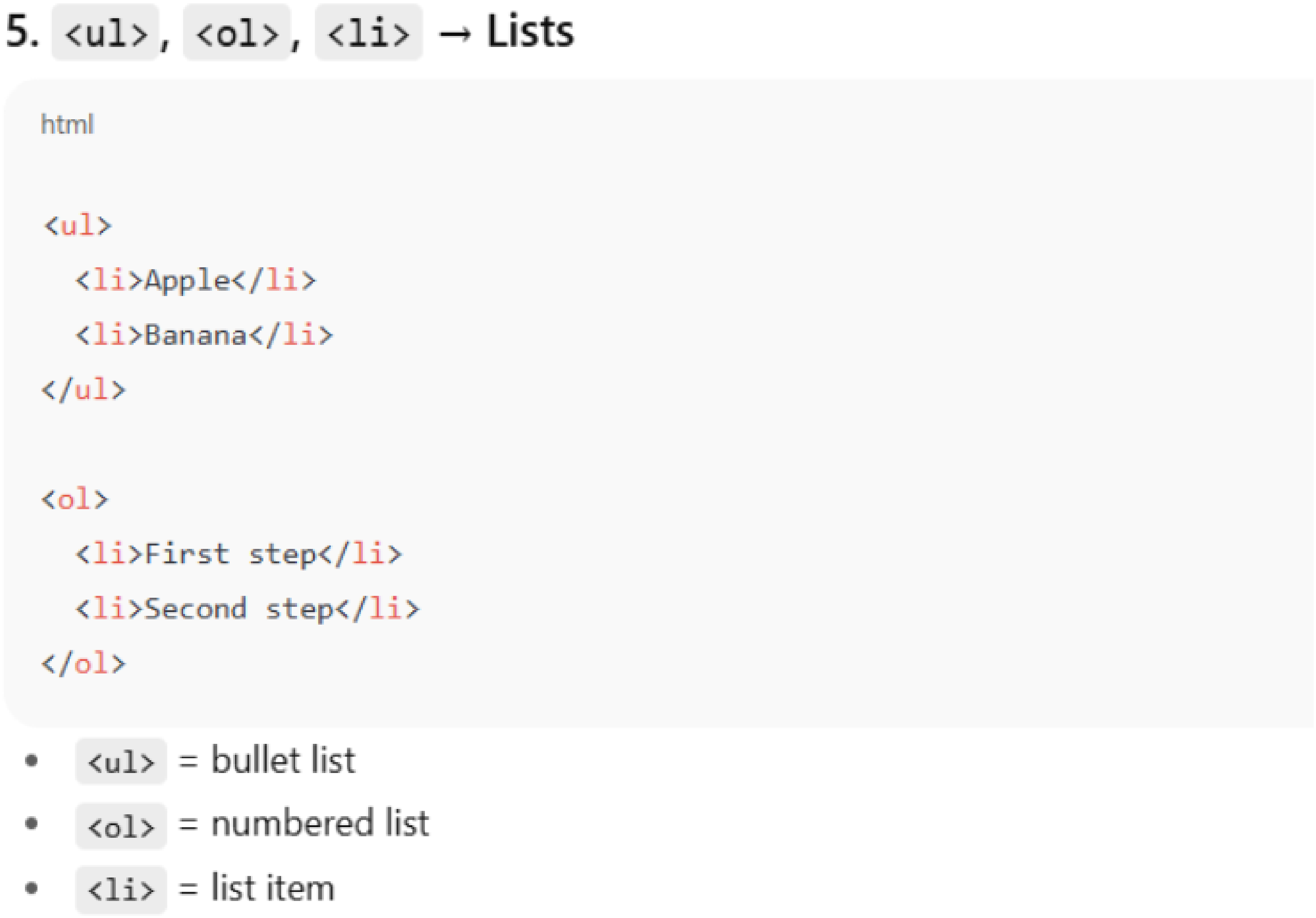
- `href` sets the link address.

---

### 4. `<img>` → Image

```html
<img src="image.jpg" alt="A description" width="150">
```

- `src` = image path
- `alt` = alternative text if image can't load

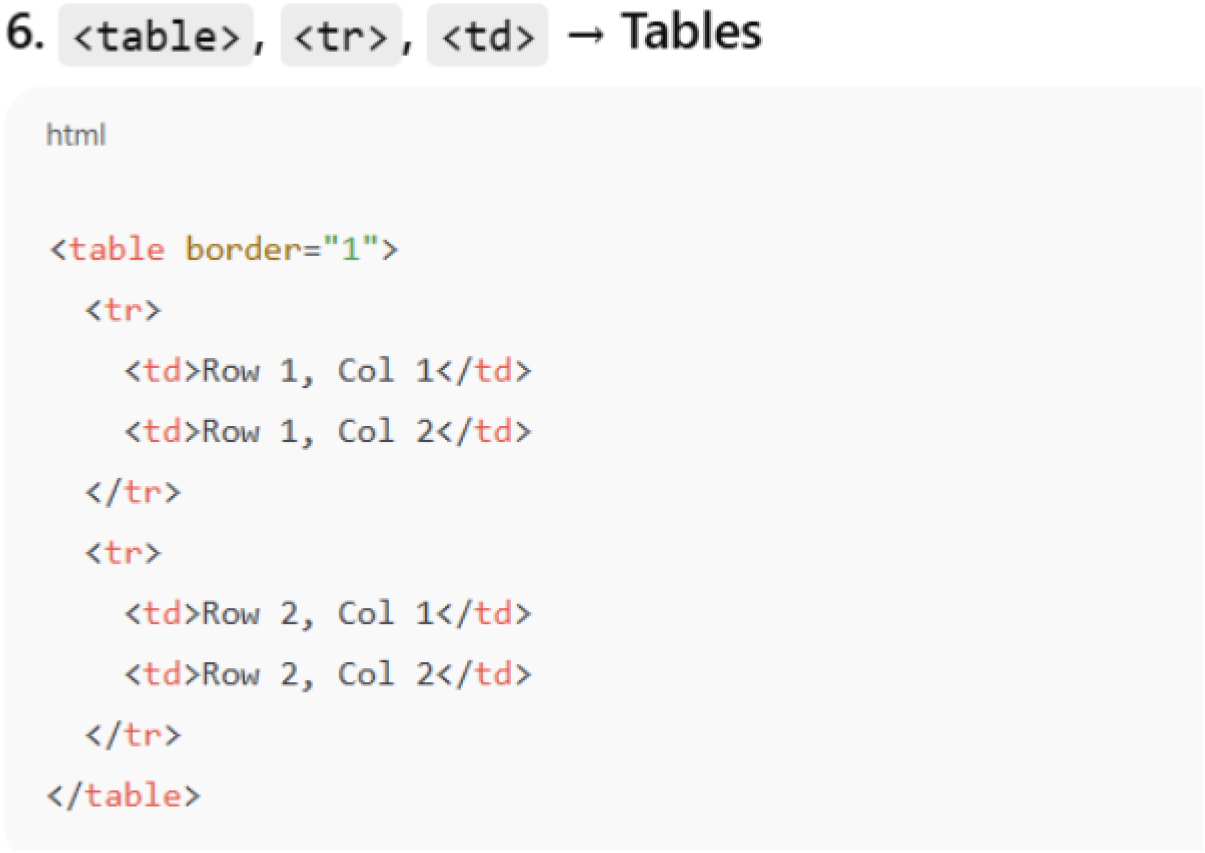


## 5. `<ul>`, `<ol>`, `<li>` → Lists

html

```html
<ul>
  <li>Apple</li>
  <li>Banana</li>
</ul>

<ol>
  <li>First step</li>
  <li>Second step</li>
</ol>
```

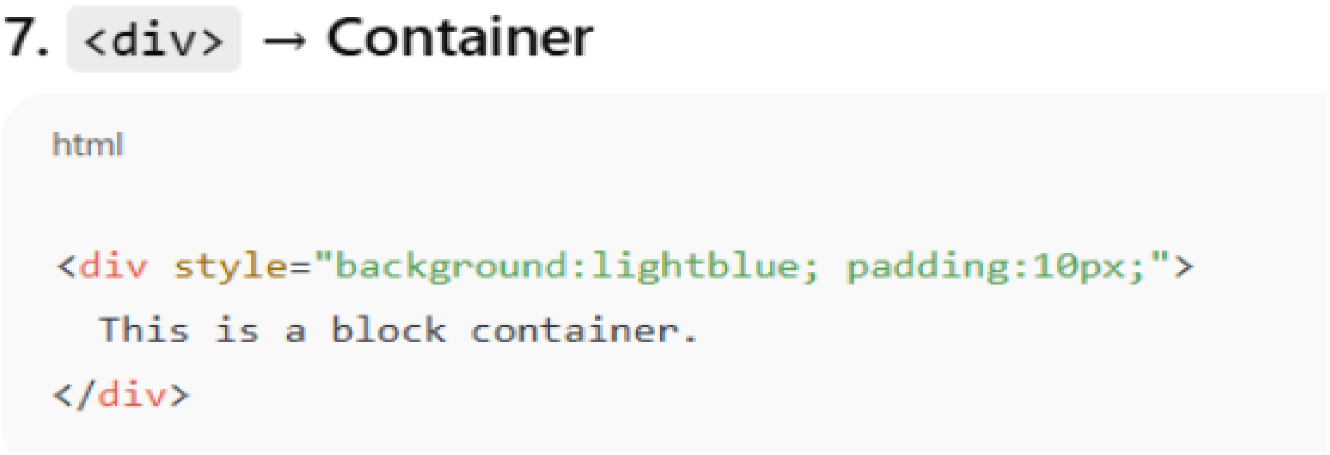
- `<ul>` = bullet list
- `<ol>` = numbered list
- `<li>` = list item



## 6. `<table>`, `<tr>`, `<td>` → Tables

html

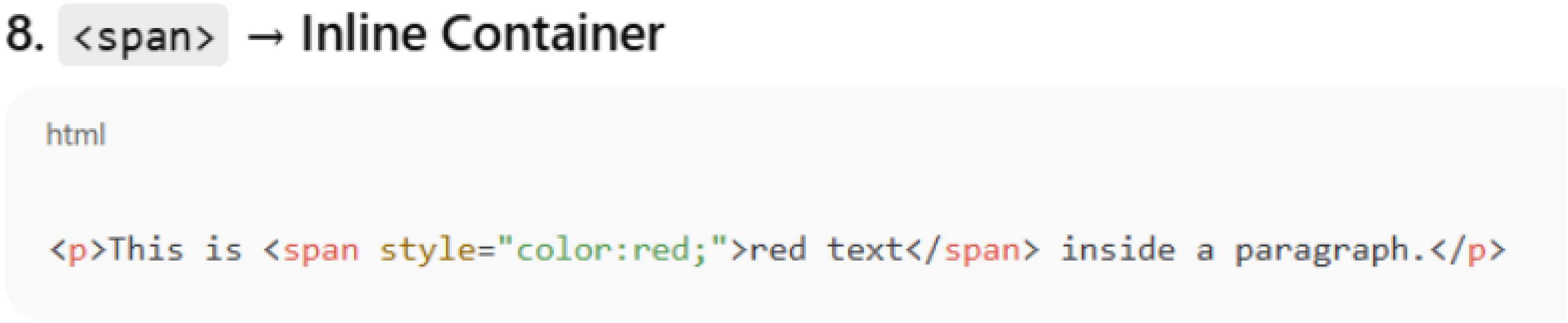
```html
<table border="1">
  <tr>
    <td>Row 1, Col 1</td>
    <td>Row 1, Col 2</td>
  </tr>
  <tr>
    <td>Row 2, Col 1</td>
    <td>Row 2, Col 2</td>
  </tr>
</table>
```



### 7. `<div>` → Container

html

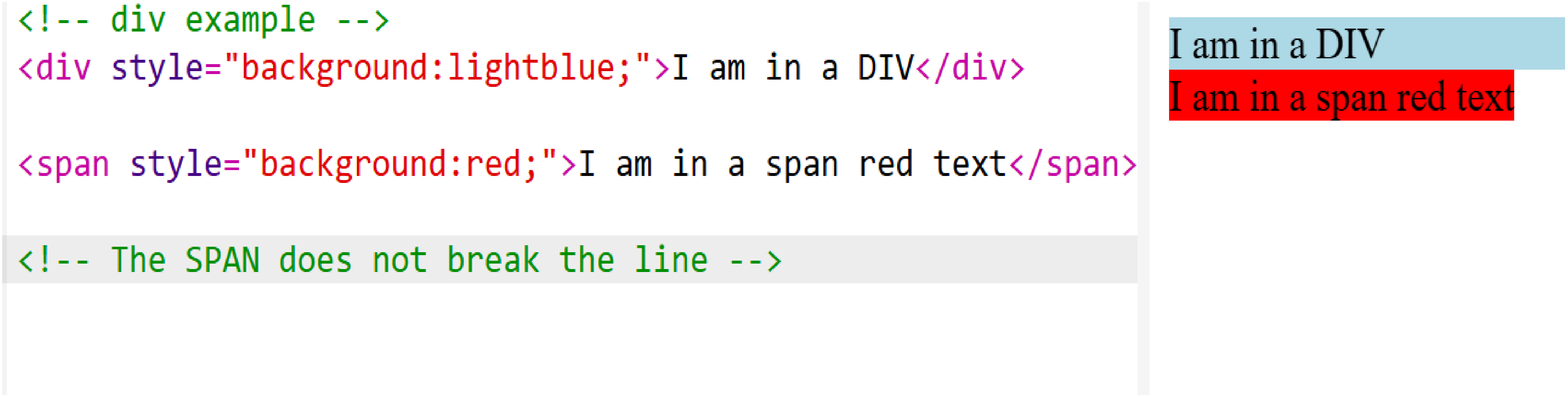
```html
<div style="background:lightblue; padding:10px;">
  This is a block container.
</div>
```



### 8. `<span>` → Inline Container

html

```html
<p>This is <span style="color:red;">red text</span> inside a paragraph.</p>
```



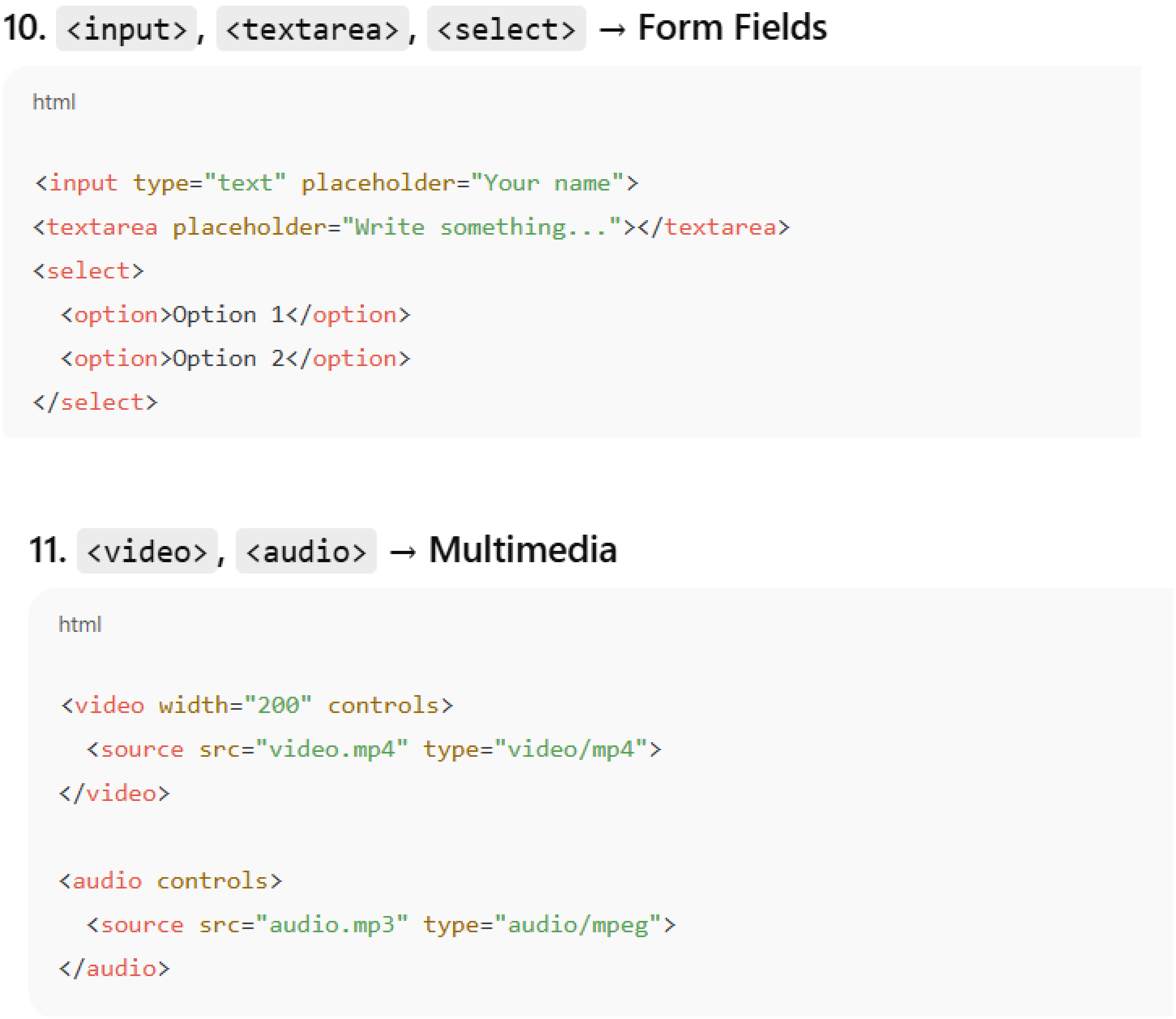
```
<!-- div example -->
<div style="background:lightblue;">I am in a DIV</div>

<span style="background:red;">I am in a span red text</span>

<!-- The SPAN does not break the line -->
```

I am in a DIV

I am in a span red text

9. `<form>` → Collect User Input

```html
<form>
  <!-- Name -->
  Name: <input type="text" name="name" placeholder="Enter your name"><br><br>

  <!-- Email -->
  Email: <input type="email" name="email" placeholder="Enter your email"><br><br>

  <!-- Age -->
  Age: <input type="number" name="age" placeholder="Enter your age"><br><br>

  <!-- Gender -->
  Gender:
  <select name="gender">
    <option value="">Select Gender</option>
    <option value="male">Male</option>
    <option value="female">Female</option>
    <option value="other">Other</option>
  </select><br><br>

  <!-- Submit -->
  <input type="submit" value="Send">
</form>
```

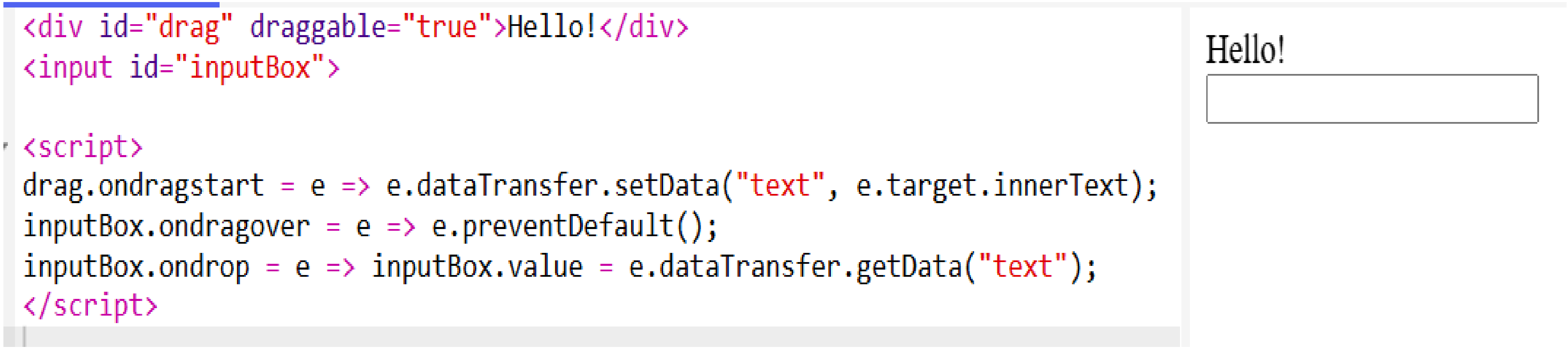


## 10. `<input>`, `<textarea>`, `<select>` → Form Fields

```html
<input type="text" placeholder="Your name">
<textarea placeholder="Write something..."></textarea>
<select>
  <option>Option 1</option>
  <option>Option 2</option>
</select>
```

## 11. `<video>`, `<audio>` → Multimedia

```html
<video width="200" controls>
  <source src="video.mp4" type="video/mp4">
</video>

<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
</audio>
```



```html
<div id="drag" draggable="true">Hello!</div>
<input id="inputBox">

<script>
drag.ondragstart = e => e.dataTransfer.setData("text", e.target.innerText);
inputBox.ondragover = e => e.preventDefault();
inputBox.ondrop = e => inputBox.value = e.dataTransfer.getData("text");
</script>
```

Hello!

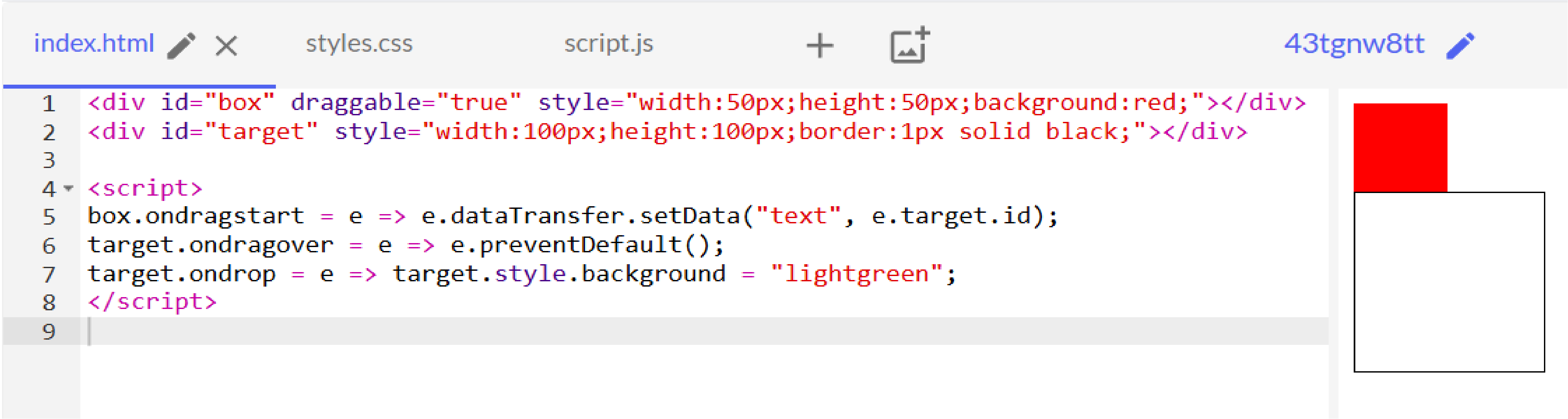
“When dragging starts, take the text from the thing the user is dragging and store it in a storage slot (label) called `text`.”

“When something is dragged over the input box, don’t block it — allow dropping here.”

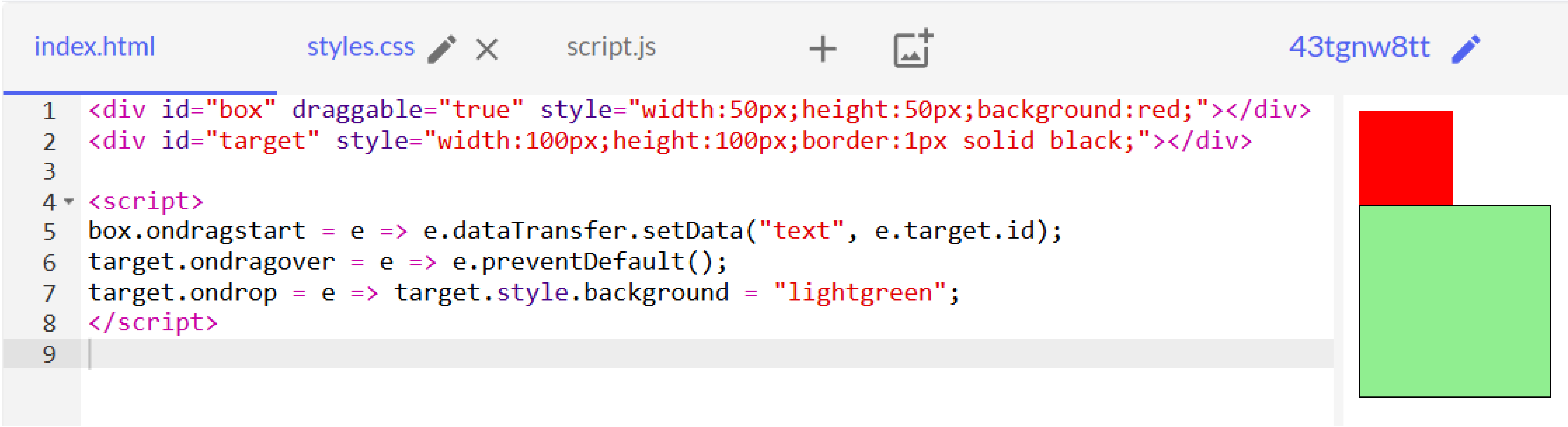
“When something is dropped onto the input box, take the text from the drag bag (labelled `text`) and put it into the input box.”

**Drag start** = put `"Hello!"` in a box labeled `"text"`.

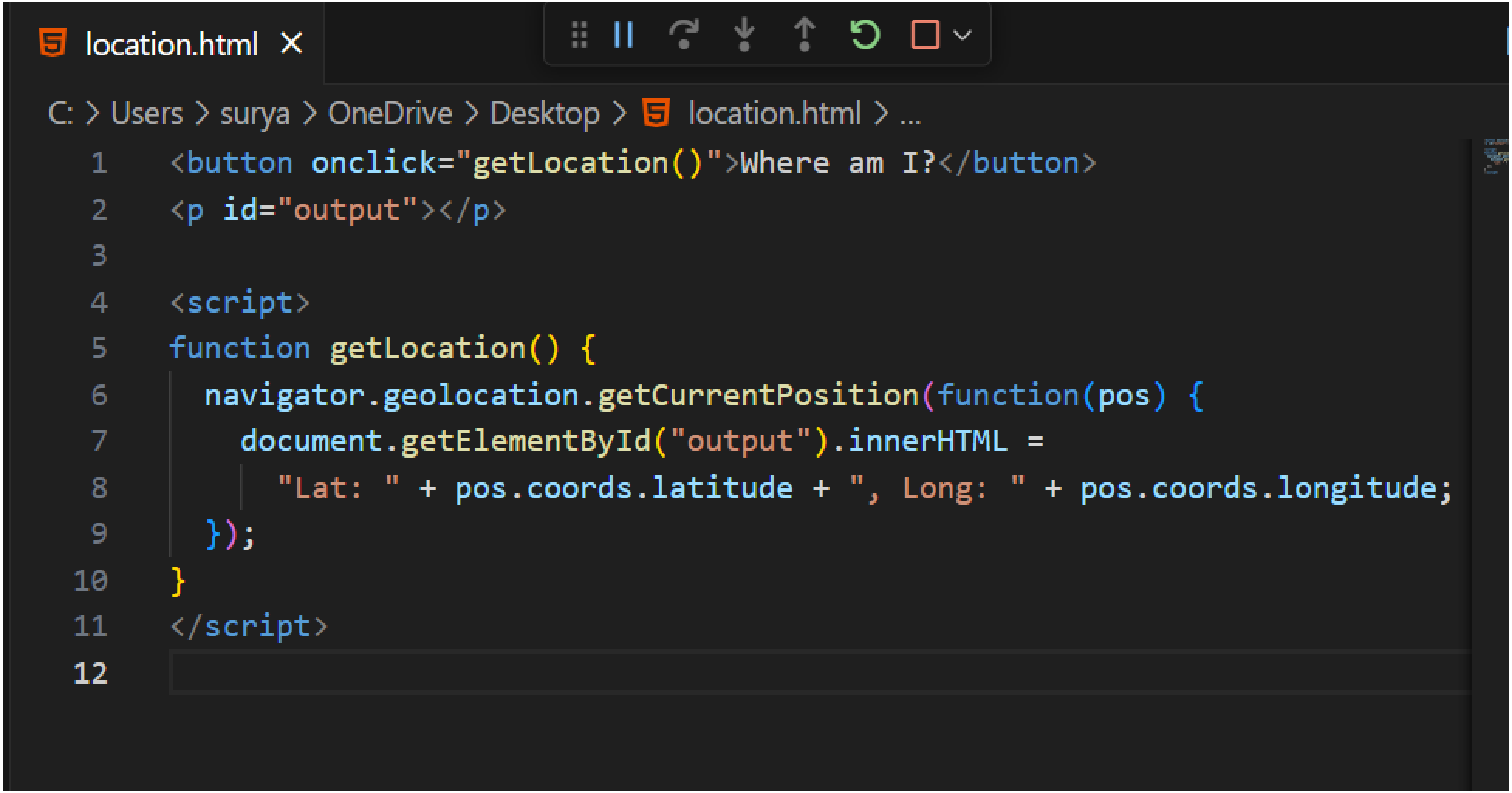
**Drop** = open the box labeled `"text"` and place `"Hello!"` into the input box.



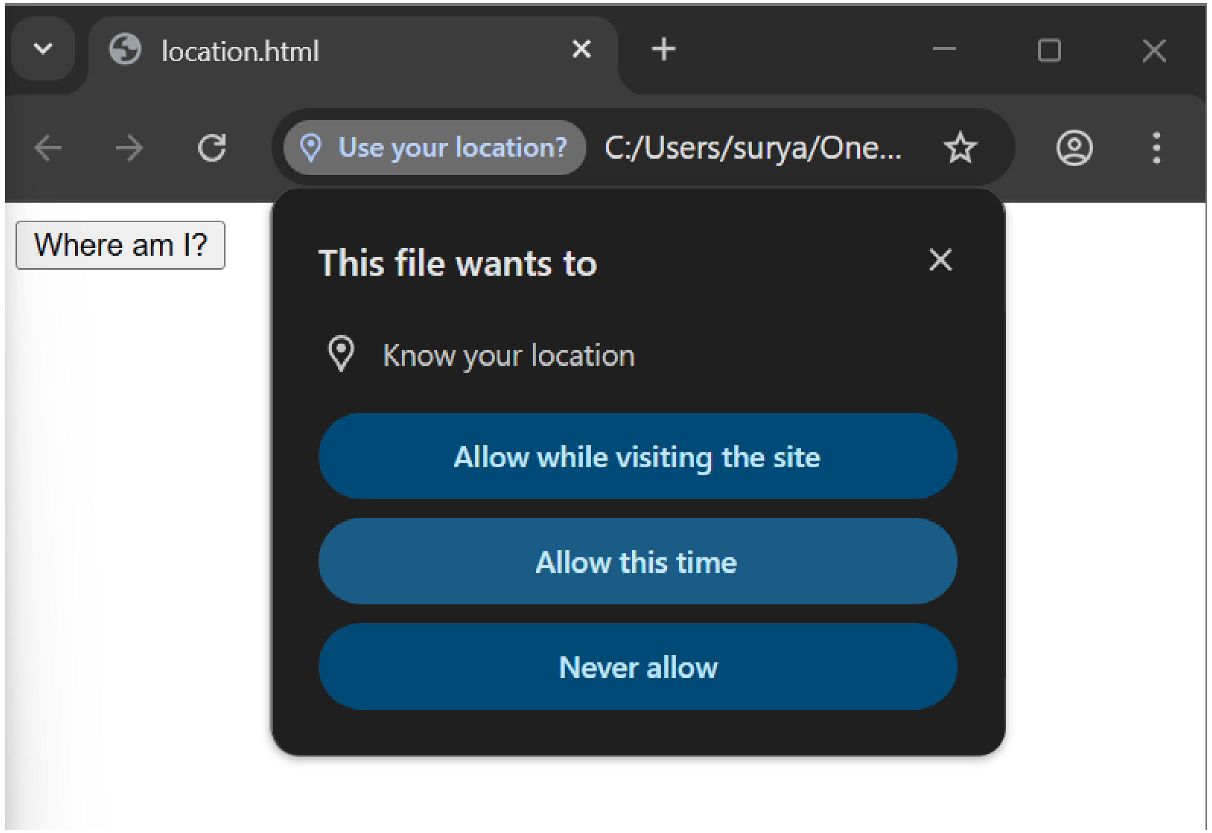
```
<div id="box" draggable="true" style="width:50px;height:50px;background:red;"></div>
<div id="target" style="width:100px;height:100px;border:1px solid black;"></div>

<script>
box.ondragstart = e => e.dataTransfer.setData("text", e.target.id);
target.ondragover = e => e.preventDefault();
target.ondrop = e => target.style.background = "lightgreen";
</script>
```



```
<div id="box" draggable="true" style="width:50px;height:50px;background:red;"></div>
<div id="target" style="width:100px;height:100px;border:1px solid black;"></div>

<script>
box.ondragstart = e => e.dataTransfer.setData("text", e.target.id);
target.ondragover = e => e.preventDefault();
target.ondrop = e => target.style.background = "lightgreen";
</script>
```

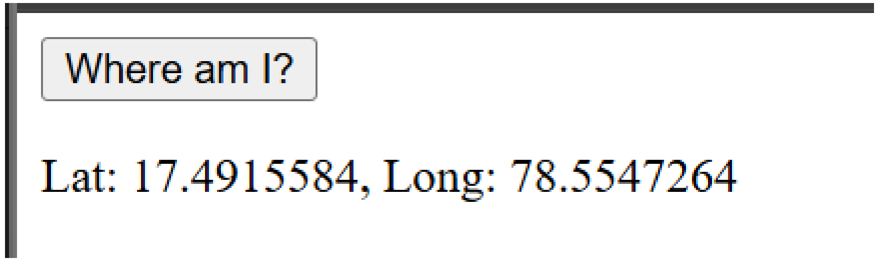

location.html
C: > Users > surya > OneDrive > Desktop > location.html > ...

```html
<button onclick="getLocation()">Where am I?</button>
<p id="output"></p>

<script>
function getLocation() {
  navigator.geolocation.getCurrentPosition(function(pos) {
    document.getElementById("output").innerHTML =
      "Lat: " + pos.coords.latitude + ", Long: " + pos.coords.longitude;
  });
}
</script>

```

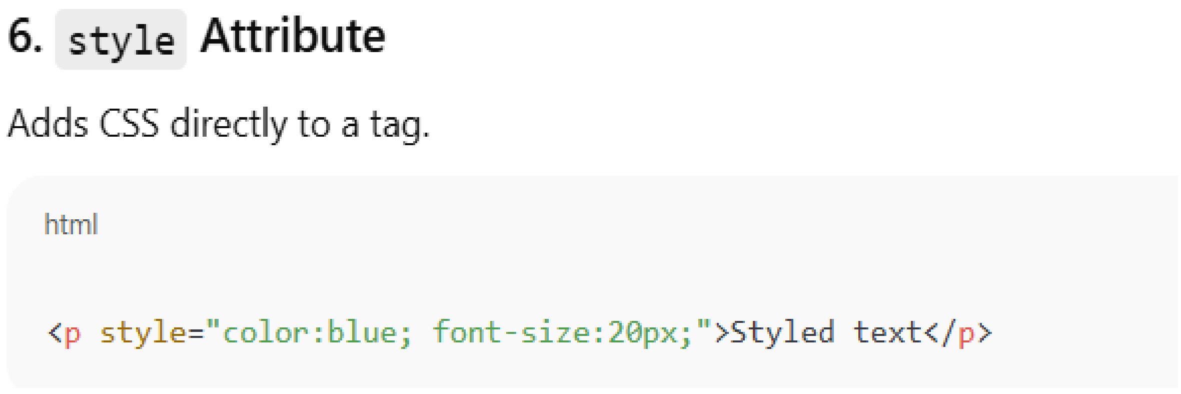


This file wants to

Know your location



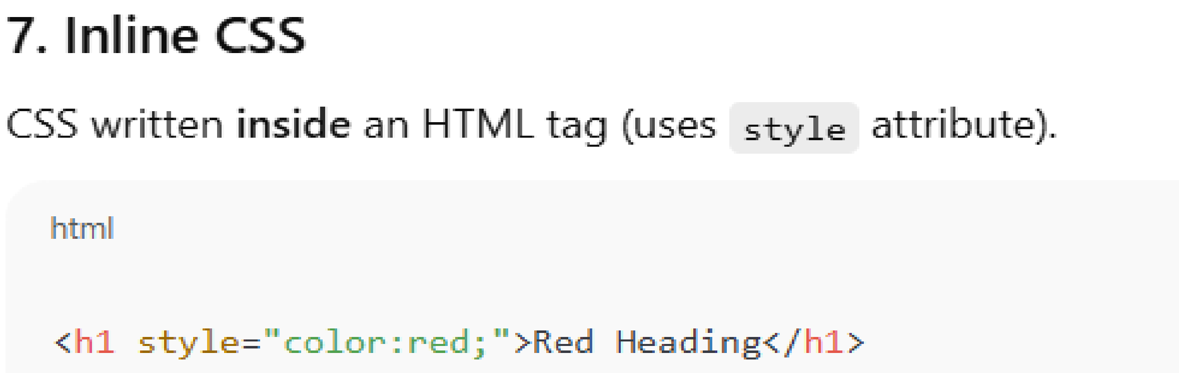
```
Lat: 17.4915584, Long: 78.5547264
```



## 6. `style` Attribute

Adds CSS directly to a tag.

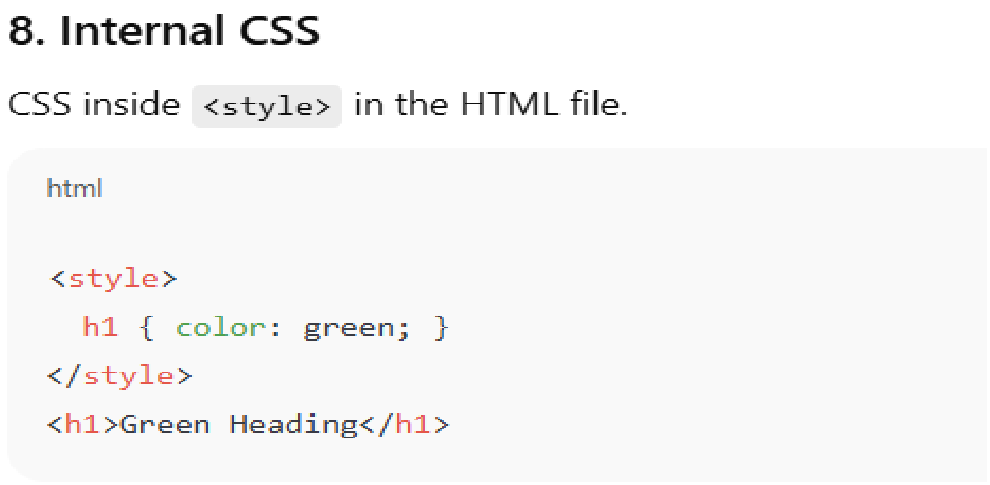
```html
<p style="color:blue; font-size:20px;">Styled text</p>
```



## 7. Inline CSS

CSS written **inside** an HTML tag (uses `style` attribute).

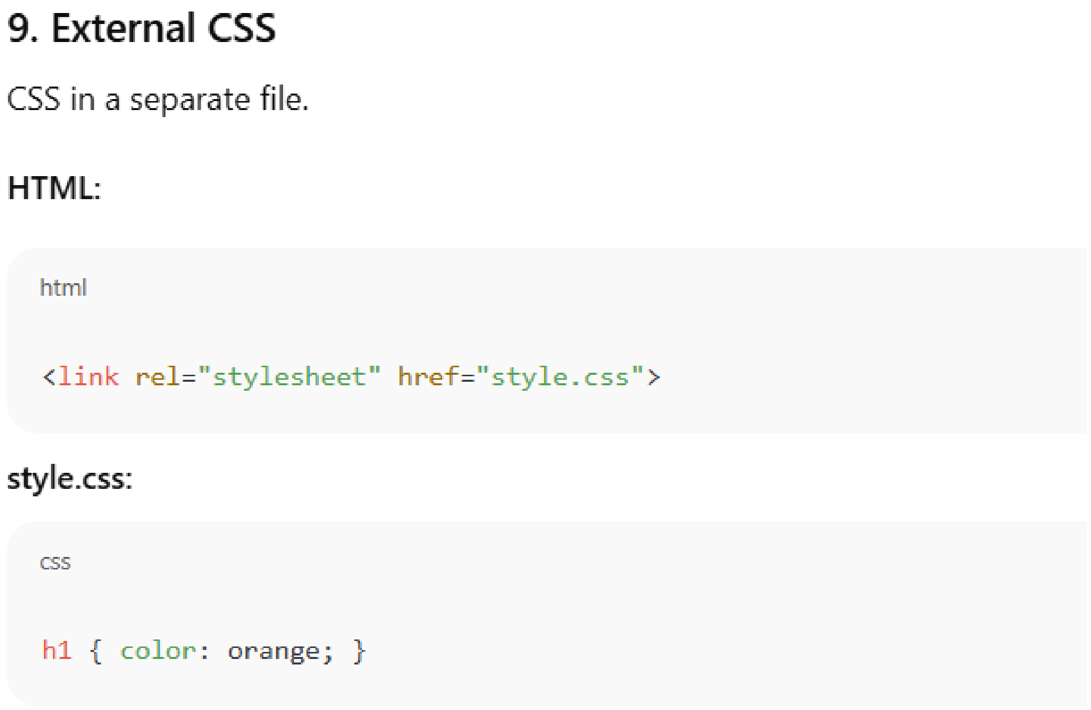
```html
<h1 style="color:red;">Red Heading</h1>
```



## 8. Internal CSS

CSS inside `<style>` in the HTML file.

```html
<style>
  h1 { color: green; }
</style>
<h1>Green Heading</h1>
```



## 9. External CSS

CSS in a separate file.

**HTML:**

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

**style.css:**

```css
h1 { color: orange; }
```

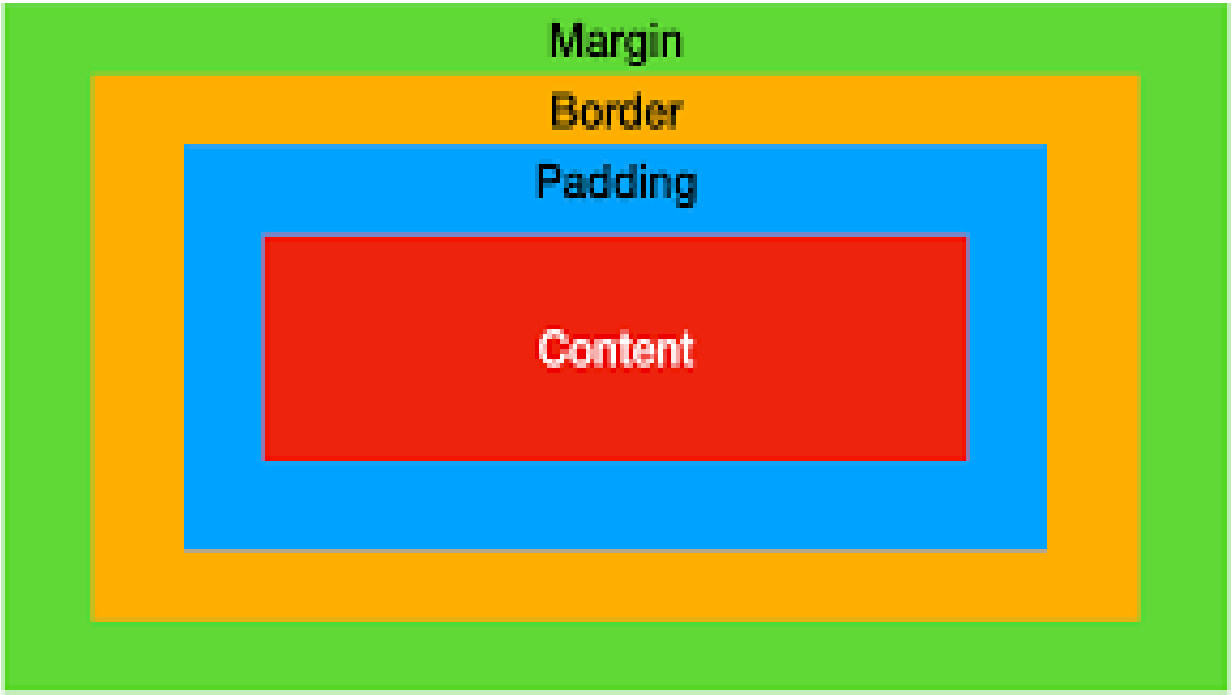
## 10. Box Model

Every HTML element is a box:

**Content → Padding → Border → Margin**

html

```html
<style>
.box {
  width: 100px;
  padding: 10px;
  border: 5px solid black;
  margin: 20px;
}
</style>
<div class="box">Hello</div>
```


Margin
Border
Padding
Content