**FORMS Validation**

**Requiring an input**

Sometimes we have fields in our <form>s which are not optional, i.e. there must be information provided before we can submit it. To enforce this rule, we can add the required attribute to an <input> element.

<form action="/example.html" method="POST">

<label for="allergies">Do you have any dietary restrictions?</label>

<br>

<input id="allergies" name="allergies" type="text" required>

<br>

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

</form>

**Set min and max**

Another built-in validation we can use is to assign a minimum or maximum value for a number field, e.g. <input type="number"> and <input type="range">. To set a minimum acceptable value, we use the min attribute and assign a value. On the flip side, to set a maximum acceptable value, we assign the max attribute a value. Let’s see this in code:

<form action="/example.html" method="POST">

<label for="guests">Enter # of guests:</label>

<input id="guests" name="guests" type="number" min="1" max="4">

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

</form>

Checking text length

To set a minimum number of characters for a text field, we add the minlength attribute and a value to set a minimum value. Similarly, to set the maximum number of characters for a text field, we use the maxlength attribute and set a maximum value. Let’s take a look at these attributes in code:

<form action="/example.html" method="POST">

<label for="summary">Summarize your feelings in less than 250 characters</label>

<input id="summary" name="summary" type="text" minlength="5" maxlength="250" required>

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

</form>

**Matching a pattern**

In addition to checking the length of a text, we could also add a validation to check how the text was provided. For cases when we want user input to follow specific guidelines, we use the pattern attribute and assign it a regular expression, or regex. Regular expressions are a sequence of characters that make up a search pattern. If the input matches the regex, the form can be submitted.

Let’s say we wanted to check for a valid credit card number (a 14 to 16 digit number). We could use the regex: [0-9]{14,16} which checks that the user provided only numbers and that they entered at least 14 digits and at most 16 digits.

To add this to a form:

<form action="/example.html" method="POST">

<label for="payment">Credit Card Number (no spaces):</label>

<br>

<input id="payment" name="payment" type="text" required pattern="[0-9]{14,16}">

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

</form>