

# ## W10 - Client-side Form Validation ##

Initial check to ensure valid data is supplied by client.

- not an exhaustive security measure
- client-side can be circumvented, so it is primarily used to improve UX.

Form Validation - Immediate feedback when you do not enter correct information

- "This Field Required" (you can't leave blank)
- Format specific pattern - phone numbers, ss#, cc#
- Valid address format
- Password validation

Reasons:

- data correctly formatted
- protect user data
- protect self

Types:

- Built-in Form validation (HTML5)
- JavaScript

HTML5 FORM VALIDATION DOES NOT RELY ON JAVASCRIPT

HTML5 FORM CONTROLS:

required - specifies field is required

- indicates user must specify a value for input.

`<input type="text" required="required">`

- minlength and maxlength specify minimum and maximum string length of textual data

`<input type="text" minlength="5">`

- min and max specify minimum and maximum numerical values

`<input type="range" min="60" step="5" max="100">`

- works for dates, month, week, time, etc

- type - specifies whether input needs to be a number, email, etc

- pattern - specifies a regular expression that defines a pattern *← example in code*

## ## W10 - Client-Side form Validation ##

If data entered in form matches rule specified, it is considered

"valid" - the element matches ":valid" CSS psuedo-class

- the browser will submit the form

"invalid" - the element matches ":invalid" CSS psuedo-class  
others include ":out.of.range"

- browser blocks form submission

### Built-in form validation examples

:invalid

:invalid:required

:valid

Validating against a regular expression

<test area> does not support the pattern attribute

Constraining length

## USING JAVASCRIPT

### CONSTRAINT VALIDATION API

<button> HTMLButtonElement

<output> HTMLOutputElement

<fieldset> HTMLFieldSetElement

<select> HTMLSelectElement

<input> HTMLInputElement

<textarea> HTMLTextAreaElement

Properties available through constraint validation API:

ValidationMessage: localized message describing validation constraints the control doesn't satisfy. If control is not a candidate for constraint validation, willValidate = false  
if valid, returns an empty string

validity: Return ValidityState object containing several properties describing the validity state of the element

[developer.mozilla.org/en-US/docs/Web/API/ValidityState/](https://developer.mozilla.org/en-US/docs/Web/API/ValidityState)

# ##W10 - Client-side Form Validation ##

## Validity State Properties:

`patternMismatch`: true if value does not match the specified pattern, false if it does. If true, matches: `invalid` css  
`tooLong`: true if value is too long, matches: `invalid` (`<maxLength>`)  
`tooShort`: true if value is too short, matches: `invalid` (`<minlength>`)  
`rangeOverflow`: true if value greater than `<max>`: `invalid` :`out-of-range`  
`rangeUnderflow`: true if value less than `<min>`: `invalid` :`out-of-range`  
`typeMismatch`: true if not required syntax (type="email" or "url") : `invalid`  
`valid`: true if meets all validation constraints : `valid`, otherwise: `invalid`  
`valueMissing`: true if required attribute without value : `invalid`  
`willValidate`: true if element will be validated when form submitted

## Methods available through Constraint Validation API

`checkValidity()`: true if element's value has no validity problems, fires an `invalid` event

`setCustomValidity(message)`: Adds a custom error message, element is considered `invalid` and message is displayed.

- no standard way to change look and feel using css
- depend on browser locale, page can be in one language and the error message another

customizing error messages is one of most common use cases

## Validating without built-in APIs

- What kind of validation do I need? string operations, type conversions, regular expression...
- What to do when form doesn't validate? UI behavior
- How to help user correct invalid data?