

**Using Javascript with**

**Forms**

# Form element, controls, and inputs

- `<form>` (HTMLFormElement)
- `<label>`
- `<input type="...">`
- `<textarea>`
- `<select>` and `<option>`
- `<button>`

```
<form method="post" action="/" id="login-form">
  <div class="input-field">
    <label for="email-input">Email</label>
    <input
      type="email"
      name="email-input"
      id="email-input"
      placeholder="example@domain.com"
      required
    />
  </div>
  <div class="input-field">
    <label for="password-input">Password</label>
    <input
      type="password"
      name="password-input"
      id="password-input"
      required
    />
  </div>
  <div class="input-field">
    <input type="submit" value="Log In" />
  </div>
</form>
```

# input types

- text: the one you use everywhere
- checkbox: a checkbox
- hidden: holds data, doesn't appear
- password: doesn't show the input
- radio: allows a single value to be selected out of multiple choices
- submit: button that submits the form
- reset: button that clears the form
- button: a button with no default behavior (you probably want a `<button>` element instead)

## **input types, bonus round**

- color
- date
- datetime-local
- email
- number
- tel
- time
- url

# Form Events

- `focus`: element gained focus
- `blur`: element lost focus
- `input`: element's value changed
- `change`: "a change to the element's value is committed by the user," usually after losing focus
- `submit`: submit button was pressed
- `reset`: reset button was pressed

# Events

- `event.target`
- `event.preventDefault()`

# DOM Manipulation, revisited

- createElement
- appendChild
- setAttribute
- getAttribute
- removeAttribute



# Form Validation

**Validation** refers to the process of checking to make sure that data entered by a user is present and correct before it is used programmatically or stored somewhere.

We can validate data on the **client-side** AND/OR on the **server-side** (usually both).

If data is **invalid**, we want to know so that we can do something, like generate an error that alerts the user (or not do something! We wouldn't want to save faulty data to a database).