

EECS 368

Programming Language Paradigms

David O. Johnson

Fall 2022

Reminders

- Assignment 3 due (today): 11:59 PM, Monday, October 3
- Assignment 4 due: 11:59 PM, Monday, October 17

Any Questions?

In-Class Problem Solution

- 16-(9-30) In-Class Problem Solution.pptx

Any Questions?

Chapter 18 - HTTP and Forms

- ~~The protocol~~
- ~~Browsers and HTTP~~
- ~~Fetch~~
- ~~HTTP sandboxing~~
- ~~Appreciating HTTP~~
- ~~Security and HTTPS~~
- Form fields
- Focus
- Disabled fields
- The form as a whole
- Text fields
- Checkboxes and radio buttons
- Select fields
- File fields
- Storing data client-side

Form Fields

- Forms were originally designed for the pre-JavaScript Web to allow web sites to send user-submitted information in an HTTP request.
 - This design assumes that interaction with the server always happens by navigating to a new page.
- But their elements are part of the DOM like the rest of the page.
 - And the DOM elements that represent form fields support a number of properties and events that are not present on other elements.
 - These make it possible to inspect and control such input fields with JavaScript programs ...
 - and do things such as adding new functionality to a form or ...
 - using forms and fields as building blocks in a JavaScript application.
- Forms can be done entirely with HTML or JavaScript, but using them together is better.

Form Fields

- A web form consists of any number of input fields grouped in a `<form>` tag.
- HTML allows several different styles of fields, ...
 - ranging from simple on/off checkboxes to ...
 - drop-down menus and ...
 - fields for text input.
- We won't try to comprehensively discuss all field types, but we'll start with a rough overview.

- Form fields do not necessarily have to appear in a `<form>` tag.
- You can put them anywhere in a page.
- Such form-less fields cannot be submitted, only a form as a whole can.
- But when responding to input with JavaScript, we often don't want to submit our fields normally anyway.

Whenever the value of a form field changes, it will fire a `"change"` event.

Form Fields

- We will look at 3 types of HTML form field tags:
- `<input>`
 - most used form element
 - can be displayed in many ways, depending on the `type` attribute
- `<textarea>`
 - defines a multi-line text input field
- `<select>`
 - used to create a field that allows the user to select from a number of predefined options.

Any Questions?

Field Types <input>

Here are the different input types you can use in HTML:

<input type="button">

<input type="checkbox">

<input type="color">

<input type="date">

<input type="datetime-local">

<input type="email">

<input type="file">

<input type="hidden">

<input type="image">

<input type="month">

<input type="number">

<input type="password">

<input type="radio">

<input type="range">

<input type="reset">

<input type="search">

<input type="submit">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

<input type="week">

We'll look at 5 commonly used ones.

See https://www.w3schools.com/html/html_form_input_types.asp for the others.

Field Types <input>

- A lot of field types use the `<input>` tag.
- This tag's `type` attribute is used to select the field's style.
- These are some commonly used `<input>` types:
 - `text` A single-line text field
 - `password` Same as `text` but hides the text that is typed
 - `checkbox` An on/off switch
 - `radio` (Part of) a multiple-choice field
 - `file` Allows the user to choose a file from their computer

The fields created with this HTML code look like this:

```
<p><input type="text" value="abc"> (text)</p>
```

```
<p><input type="password" value="abc"> (password)</p>
```

```
<p><input type="checkbox" checked> (checkbox)</p>
```

```
<p><input type="radio" value="A" name="choice">
```

```
<input type="radio" value="B" name="choice" checked>
```

```
<input type="radio" value="C" name="choice"> (radio)</p>
```

```
<p><input type="file"> (file)</p>
```

abc (text)

... (password)

☒ (checkbox)

☐ ☒ ☐ (radio)

Browse... No file selected. (file)

Field Types <input>

- A lot of field types use the `<input>` tag.
- This tag's `type` attribute is used to select the field's style.
- These are some commonly used `<input>` types:
 - `text` A single-line text field
 - `password` Same as `text` but hides the text that is typed
 - `checkbox` An on/off switch
 - `radio` (Part of) a multiple-choice field
 - `file` Allows the user to choose a file from their computer

The fields created with this HTML code look like this:

```
<p><input type="text" value="abc"> (text)</p>
<p><input type="password" value="abc"> (password)</p>
<p><input type="checkbox" checked> (checkbox)</p>
<p><input type="radio" value="A" name="choice">
<input type="radio" value="B" name="choice" checked>
<input type="radio" value="C" name="choice"> (radio)</p>
<p><input type="file"> (file)</p>
```

abc (text)

... (password)

☒ (checkbox)

☐ ☒ ☐ (radio)

Browse... No file selected. (file)

Field Types <input>

- A lot of field types use the `<input>` tag.
- This tag's `type` attribute is used to select the field's style.
- These are some commonly used `<input>` types:
 - `text` A single-line text field
 - `password` Same as `text` but hides the text that is typed
 - `checkbox` An on/off switch **"checked" indicates box is initially checked**
 - `radio` (Part of) a multiple-choice field
 - `file` Allows the user to choose a file from their computer

The fields created with this HTML code look like this:

```
<p><input type="text" value="abc"> (text)</p>
<p><input type="password" value="abc"> (password)</p>
<p><input type="checkbox" checked> (checkbox)</p>
<p><input type="radio" value="A" name="choice">
<input type="radio" value="B" name="choice" checked>
<input type="radio" value="C" name="choice"> (radio)</p>
<p><input type="file"> (file)</p>
```

abc (text)

... (password)

☒ (checkbox)

☐ ☒ ☐ (radio)

Browse... No file selected. (file)

Field Types <input>

- A lot of field types use the `<input>` tag.
- This tag's `type` attribute is used to select the field's style.
- These are some commonly used `<input>` types:
 - `text` A single-line text field
 - `password` Same as `text` but hides the text that is typed
 - `checkbox` An on/off switch
 - `radio` (Part of) a multiple-choice field (A, B, and C are not displayed)
 - `file` Allows the user to choose a file from their computer

The fields created with this HTML code look like this:

```
<p><input type="text" value="abc"> (text)</p>
<p><input type="password" value="abc"> (password)</p>
<p><input type="checkbox" checked> (checkbox)</p>
<p><input type="radio" value="A" name="choice">
<input type="radio" value="B" name="choice" checked>
<input type="radio" value="C" name="choice"> (radio)</p>
<p><input type="file"> (file)</p>
```

abc (text)

... (password)

☒ (checkbox)

☐ ☒ ☐ (radio)

Browse... No file selected. (file)

Field Types <input>

- A lot of field types use the `<input>` tag.
- This tag's `type` attribute is used to select the field's style.
- These are some commonly used `<input>` types:
 - `text` A single-line text field
 - `password` Same as `text` but hides the text that is typed
 - `checkbox` An on/off switch
 - `radio` (Part of) a multiple-choice field
 - `file` Allows the user to choose a file from their computer

The fields created with this HTML code look like this:

```
<p><input type="text" value="abc"> (text)</p>
<p><input type="password" value="abc"> (password)</p>
<p><input type="checkbox" checked> (checkbox)</p>
<p><input type="radio" value="A" name="choice">
<input type="radio" value="B" name="choice" checked>
<input type="radio" value="C" name="choice"> (radio)</p>
<p><input type="file"> (file)</p>
```

abc (text)

... (password)

☒ (checkbox)

☐ ☒ ☐ (radio)

Browse... No file selected. (file)

Any Questions?

Field Types <textarea>

- The <textarea> tag defines a multi-line text input field.
- The <textarea> element is often used in a form, to collect user inputs like comments or reviews.
- A text area can hold an unlimited number of characters, and the text renders in a fixed-width font (usually Courier).
- The name attribute is needed to reference the form data after the form is submitted.
- The size of a text area is specified by the cols and rows attributes.

```
<textarea name="w3review-1" rows="4" cols="50">  
</textarea>
```

```
<textarea name="w3review-2" rows="10" cols="30">  
</textarea>
```

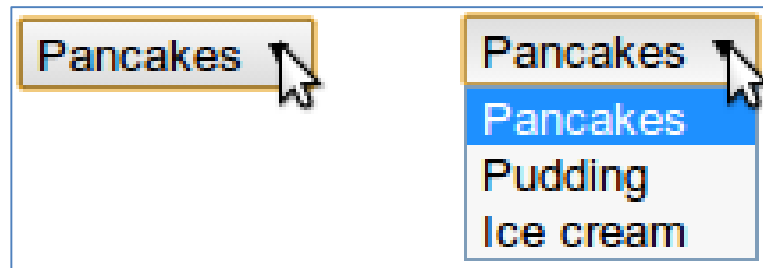


Any Questions?

Field Types <select>

The `<select>` tag is used to create a field that allows the user to select from a number of predefined options.

```
<select>  
  <option>Pancakes</option>  
  <option>Pudding</option>  
  <option>Ice cream</option>  
</select>
```



Any Questions?

Keyboard Focus

- Unlike most elements in HTML documents, form fields can get **keyboard focus**.
- When clicked or activated in some other way, they become the currently active element and the recipient of keyboard input.
- Thus, you can type into a text field only when it is focused.
- Other fields respond differently to **keyboard events**.
- For example, a `<select>` menu tries to move to the option that contains the **text the user typed** and ...
- responds to the **arrow keys** by moving its selection up and down.

focus and blur Methods

- We can control focus from JavaScript with the `focus` and `blur` methods.
- `focus` - moves focus to the DOM element it is called on
- `blur` – removes focus
- The value in the `document.activeElement` corresponds to the currently focused element.

```
<input type="text">
<script>
  document.querySelector("input").focus();
  console.log(document.activeElement.tagName);
  // → INPUT

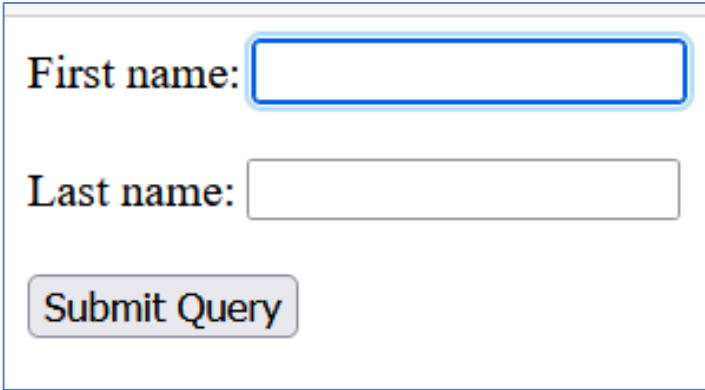
  document.querySelector("input").blur();
  console.log(document.activeElement.tagName);
  // → BODY
</script>
```

Autofocus with JavaScript

- For some pages, the user is expected to want to interact with a form field immediately.
- JavaScript can be used to focus this field when the document is loaded.

Setting autofocus with JavaScript:

```
var x = document.getElementById("firstName").autofocus;
```



First name:

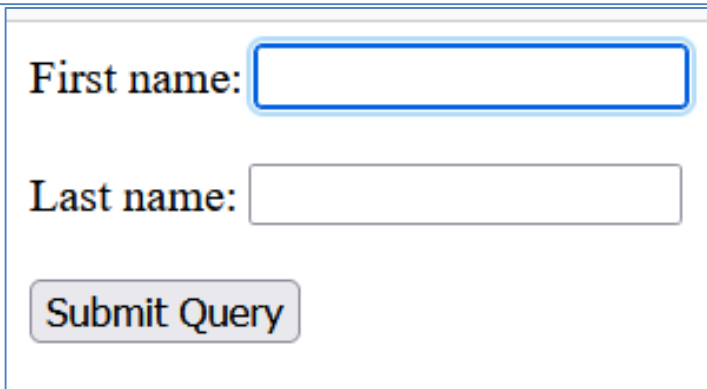
Last name:

Autofocus with HTML

- But HTML also provides the `autofocus` attribute, which produces the same effect while letting the browser know what we are trying to achieve.

Setting autofocus with HTML:

```
<form action="/action_page.php">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname" autofocus><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <input type="submit">  
</form>
```



First name:

Last name:

tabindex Attribute

- Browsers traditionally also allow the user to move the focus through the document by pressing the tab key.
- We can influence the order in which elements receive focus with the `tabindex` attribute.
- The following example document will let the focus jump from the text input to the OK button, rather than going through the help link first:

```
<input type="text" tabindex=1> <a href=".">(help)</a>  
<button onclick="console.log('ok')" tabindex=2>OK</button>
```



- By default, most types of HTML elements cannot be focused.
- But you can add a `tabindex` attribute to any element that will make it focusable.
- A `tabindex` of -1 makes tabbing skip over an element, even if it is normally focusable.

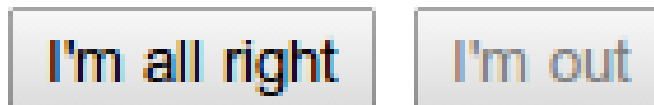
Any Questions?

Disabled Fields

- All form fields can be *disabled* through their `disabled` attribute.
- It is an attribute that can be specified without value.
- The fact that it is present at all disables the element.

```
<button>I'm all right</button>  
<button disabled>I'm out</button>
```

- Disabled fields cannot be focused or changed, and browsers make them look gray and faded.



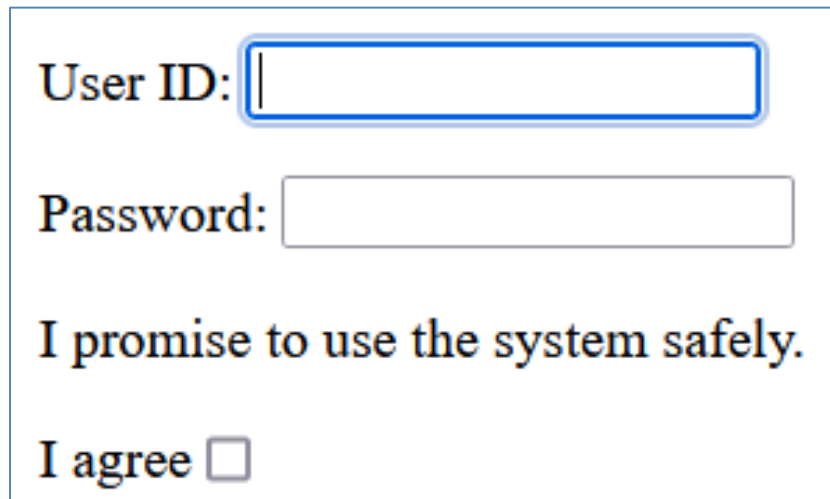
- When a program is in the process of handling an action caused by some button or other control that might require communication with the server and thus take a while, ...
- it can be a good idea to disable the control until the action finishes.
- That way, when the user gets impatient and clicks it again, they don't accidentally repeat their action.

Any Questions?

In-Class Problem

Create an HTML form that displays the following:

- Prompts a user for a user ID with “User ID:”
- Prompts the user for a hidden password with “Password:”
- Displays the message “I promise to use the system safely.”
- Displays a check box that says “I agree”.
- Autofocuses to the “User ID:” field.



User ID:

Password:

I promise to use the system safely.

I agree ☐