OIM3690 - Web Technologies

JavaScript - Form Processing

Accessing `input`

- Use `name` attribute of inputs to access their values
- Example:

```
<form id="form-feedback" action="">
  Your Name: <input type="text" name="userName" maxlength="30"/>
</form>
```

In JavaScript:

```
const myForm = document.getElementById("form-feedback");
const name = myForm.userName.value; //get the value typed in this field, or
const name = myForm["userName"].value; //get the same value
console.log(name)

// We can modify the input value as below
myForm["userName"].value = "Michael Scott";
```

Accessing `textarea`

- Example:

```
...
<label>Enter your comments below:<br />
  <textarea name="comments" cols="30" rows="10"></textarea>
</label>
```

In JavaScript:

```
// ...
const comments = myForm["comments"].value;
```

Accessing `select` and `option`

- Example:

```
...
Select the browsers you like:
<select name="browser">
  <option value="Chrome">Chrome</option>
  <option value="Safari">Safari</option>
</select>
```

in JavaScript:

```
// ...
const selectedBrowser = myForm["browser"].value;
// Get the value of the option that the user has selected
```

Exercise: *ex16-calculator.html*

- Create a **form** to implement a simple calculator with the basic functionalities:
  - Addition `+` / subtraction `-` / multiplication `x` / division `-`
- You can design the look of the calculator whatever way you like. However, make sure that the form includes
  - Two `input`'s for the operands (two numbers),
  - A select element (or multiple buttons) to choose the operation,
  - Other elements you think is necessary
- Write JavaScript function(s) that handles events, such as clicking on buttons.
  - You may need `parseFloat()` to convert a string to a floating-point number.
  - No need to validate inputs.
- Update ***sitemap.html*** and **commit/push** to GitHub.

Questions?