Comp 125 - Visual Information Processing

Spring Semester 2018 - week 7 - friday

Dr Nick Hayward

HTML & JS - Random Greeting Generator - variant 1

example solution - project structure

- sample project structure
    - *project specific directory, e.g.* `myproject`
    - *project subdirectory for* `assets`
    - *assets include JS -* `greeting.js`

```
./
|-- assets
|   |-- js
|       |__ greeting.js
|-- index.html
```

HTML & JS - Random Greeting Generator - variant 1

example solution - HTML

- start with basic HTML template
  - *including metadata in the <head> element*
    - reference to JS script file at foot of <body> element

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- title -->
    <title>Random Greeting Generator</title>
  </head>
  <body>

    <!-- script files -->
    <script src="./assets/js/greeting.js"></script>
  </body>
</html>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - HTML

- add application header and heading to top of `<body>`

```
<header>
  <h2>Create a random greeting...</h2>
</header>
```

- add `<main>` element to `<body>`

```
<main>
  ...
</main>
```

- add `<footer>` to end of `<body>`

```
<footer>
  ...
</footer>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - HTML

- add first `<section>` with form, input, and button to `<main>`

```html
<!-- elements for getting user input -->
<section id="generator">
  <header>
    <h3>Enter a name for the greeting</h3>
  </header>
  <form>
    <!-- player input for guessing a letter -->
    <input name="customName" placeholder="enter a name" type="text" autofocus id="name" class="txtInput"/>
    <!-- send guess letter -->
    <button type="button" id="greetingBtn">create greeting</button>
  </form>
</section>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - HTML

- add second `<section>` with `<p>` for output content to `<main>`

```html
<!-- elements for outputting generated random greeting -->
<section id="output">
    <header>
        <h3>Greeting...</h3>
    </header>
    <p id="greeting"></p>
</section>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - full HTML

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- title -->
    <title>Random Greeting Generator</title>
  </head>
  <body>
    <header>
      <h2>Create a random greeting...</h2>
    </header>
    <main>
      <!-- elements for getting user input -->
      <section id="generator">
        <header>
          <h3>Enter a name for the greeting</h3>
        </header>
        <form>
          <!-- player input for guessing a letter -->
          <input name="customName" placeholder="enter a name" type="text" autofocus id="name" class="txtInpu
          <!-- send guess letter -->
          <button type="button" id="greetingBtn">create greeting</button>
        </form>
      </section>
      <!-- elements for outputting generated random greeting -->
      <section id="output">
        <header>
          <h3>Greeting...</h3>
        </header>
        <p id="greeting"></p>
      </section>
    </main>
    <footer>
      <p>developed by ancientlives</p>
    </footer>
    <!-- script files -->
    <script src="./assets/js/greeting.js"></script>
  </body>
</html>
```

HTML & JS - Random Greeting Generator - variant 1

example solution - JS logic

- start by defining an array for the random greeting

```
// define random greetings - initial fixed examples...
var greetings = [
  "Hello ",
  "Bonjour ",
  "Guten Tag ",
  "Χαίϱετε ",
  "Salve ",
  "Ciao ",
  "こんにちは "
];
```

- we'll need to get a random value from this array
  - *called in an event listener for the form button*

```
// pick a random greeting message
var greeting = greetings[Math.floor(Math.random() * greetings.length)];
```

HTML & JS - Random Greeting Generator - variant 1

example solution - JS logic

- add event listener for `click` on form button

```
// select guess button in document
var greetingBtn = document.getElementById('greetingBtn');

// listen for user click on `greeting` button
greetingBtn.addEventListener('click', function() {
  ...
}, false);
```

HTML & JS - Random Greeting Generator - variant 1

example solution - JS logic

- add `greeting` variable to event listener for form button
- get value from form input text field - `name` entered by user
  - *concatenate `greeting` and `name`*
  - *creates greeting to output to user*

```
// pick a random greeting message
var greeting = greetings[Math.floor(Math.random() * greetings.length)];
// get name value from input field
var name = document.getElementById('name').value;
// create greeting message
var greetingMessage = greeting + name;
```

HTML & JS - Random Greeting Generator - variant 1

example solution - JS logic

- add usability updates to application
  - *reset form input field*
  - *reset focus on input field*

```
// reset input field
document.getElementById('name').value = '';
// reset focus on input field
document.getElementById('name').focus();
```

- output greeting message to user

```
// output greeting message to user
document.getElementById('greeting').innerHTML = 'random greeting: ' + greetingMessage;
```

HTML & JS - Random Greeting Generator - variant 2

example solution - update JS logic

- abstract JS logic with function `generateGreeting()`
  - *add `greetings` array*
  - *get random `greeting`*
  - *return `greeting` value from function*
  - *accept parameter for `name`*
  - *use `name` with*

```
// FN: greetings generator
function generateGreeting(name) {
  // define random greetings - initial fixed examples...
  let greetings = [
    `Hello ${name}, how are you?`,
    `Bonjour ${name}, ça va? `,
    `Guten tag ${name}, wie geht es ihnen?`,
    `Χαίρετε ${name}, Πώς είσαι;`,
    `Salve ${name}, quid agis?`,
    `Ciao ${name}, come va?`,
    `こんにちは ${name}, お元気ですか?`
  ];
    // pick a random greeting message
  let greeting = greetings[Math.floor(Math.random() * greetings.length)];
    // return greeting message
    return greeting;
}
```

JS - ES6 template literals

an updated option for concatenation

- concatenate strings, values, variables &c. using **template literals**
  - *new to ES6 (ES2015) JavaScript update*

```
`Hello ${name}, how are you?`
```

- start and end string with a backtick (grave accent in French)

```
`...`
```

- add string

```
`Hello`
```

- then inject variable, value &c. into template literal with string
  - *adds required code for concatenation with string*

```
${name}
```

HTML & JS - Random Greeting Generator - variant 2

example solution - update JS logic

- update event listener for form button click
  - *call* `generateGreeting()` *function*
  - *pass* `name` *as argument to function,* `generateGreeting(name)`

```
// LISTEN: for user click on `greeting` button
greetingBtn.addEventListener('click', function() {
  // get name value from input field
  let name = document.getElementById('name').value;
  // get greeting message - pass input name...
  let greetingMessage = generateGreeting(name);

  // reset input field
  document.getElementById('name').value = '';
  // reset focus on input field
  document.getElementById('name').focus();
  // output greeting message to user
  document.getElementById('greeting').innerHTML = 'random greeting: ' + greetingMessage;
}, false);
```

References

- W3Schools
- JS - conditionals
- JS - For loop
- JS - functions
- JS - Math Object