

Comp 125 - Visual Information Processing

Spring Semester 2019 - Week 13 - Friday

Dr Nick Hayward

HTML5, CSS, & JS - example - part 5.2

travel.js - plain JS

```
function travelNotes() {  
  "use strict";  
  
  // get a reference to `.note_output` in the DOM  
  // n.b. these can be combined as well...  
  let noteOutput = document.querySelector('.note-output');  
  noteOutput.innerHTML = '<p>first travel note for Marseille...</p>';  
  
}  
  
// load app  
travelNotes();
```

- DEMO I - travel notes - series I

HTML5, CSS, & JS - example - part 6

add a note

- app's structure includes three clear semantic divisions of content
 - `<header>`, `<main>`, and `<footer>`
- `<main>` content category - create and add our notes for our application
- allow a user to create a new note
 - *enter some brief text, and then set it as a note*
- output will simply resemble a heading or brief description for our note
- add HTML element `<input>` to allow a user to enter note text
 - *new attributes in HTML5 such as `autocomplete`, `autofocus`, `required`, `width`...*
 - *set accompanying*

```
<h5>add note</h5>  
<input>
```

```
<input type="text" value="add a note...">
```

HTML5, CSS, & JS - example - part 7

tidy up styling

- additional styles to create correct, logical separation of visual elements and content
- add a border to the top of our footer
 - *perhaps matching the header in style*
- update the box model for the `<main>` element
- add some styling for `<h5>` heading

```
h5 {  
  font-size: 1.25em;  
  margin: 10px 0 10px 0;  
}  
main {  
  overflow: auto;  
  padding: 15px 0 15px 0;  
}  
footer {  
  margin-top: 5px;  
  border-top: 1px solid #dedede;  
}
```

HTML5, CSS, & JS - example - part 8

input update

```
<input><button>add</button>
```

```
.note-input input {  
  width: 40%;  
}  
.note-input button {  
  padding: 2px;  
  margin-left: 5px;  
  border-radius: 0;  
  border: 1px solid #dedede;  
  cursor: pointer;  
}
```

- also update css for `input` and `button`
- remove button's rounded borders to match style of `input`
- match border for button to basic design aesthetics
- set cursor appropriate for a link style...
- DEMO 2 - travel notes - series I

HTML5, CSS, & JS - example - part 9.1

interaction - add a note

- added and styled our input and button for adding a note
- use jQuery to handle click event on button
- update `travel.js` file for event handler

```
//handle user event for `add` button click  
$(".note-input button").on("click", function(e) {  
    console.log("add button clicked...");  
});
```

HTML5, CSS, & JS - example - part 9.2

interaction - add a note - plain JS

```
let addNoteBtn = document.getElementById('add-note');
addNoteBtn.addEventListener('click', () => {
  console.log('add button clicked...');
});
```

HTML5, CSS, & JS - example - part 10.1

interaction - add a note - output

- update this jQuery code to better handle and output the text from the input field
- what is this handler actually doing?
 - *jQuery code has attached an event listener to an element in the DOM*
 - *referenced in the selector option at the start of the function*
 - *uses standard CSS selectors to find the required element*
- jQuery can select and target DOM elements using standard CSS selectors
 - *then manipulate them, as required, using JavaScript*

```
//handle user event for `add` button click
$(".note-input button").on("click", function(e) {
  $(".note-output").append("<p>sample note text...</p>");
});
```

- output some static text to note-output

HTML5, CSS, & JS - example - part 10.2

interaction - add a note - output - plain JS

```
function travelNotes() {
  "use strict";

  // get a reference to `.note_output` in the DOM
  let noteOutput = document.querySelector('.note-output');
  // add note button
  let addNoteBtn = document.getElementById('add-note');

  // add event listener to add note button
  addNoteBtn.addEventListener('click', () => {
    // create p node
    let p = document.createElement('p');
    // create text node
    let noteText = document.createTextNode('sample note text...');
    // append text to paragraph
    p.appendChild(noteText);
    // append new paragraph and text to existing note output
    noteOutput.appendChild(p);
  });
}
```

- DEMO 3 - travel notes - series 1

HTML5, CSS, & JS - example - part 11.1

interaction - add a note - output

```
//overall app logic and loader...
function travelNotes() {
    "use strict";

    //handle user event for `add` button click
    $(".note-input button").on("click", function(e) {
        //object for wrapper html for note
        var $note = $("

");
        //get value from input field
        var note_text = $(".note-input input").val();
        //set content for note
        $note.html(note_text);
        //append note text to note-output
        $(".note-output").append($note);
    });
};

$(document).ready(travelNotes);


```

HTML5, CSS, & JS - example - part I I.2

interaction - add a note - output - plain JS

```
function travelNotes() {
  "use strict";

  // get a reference to `.note_output` in the DOM
  let noteOutput = document.querySelector('.note-output');
  // add note button
  let addNoteBtn = document.getElementById('add-note');
  // input field for add note
  let inputNote = document.getElementById('input-note');

  addNoteBtn.addEventListener('click', () => {
    // create p node
    let p = document.createElement('p');
    // get value from input field for note
    let inputVal = inputNote.value;
    // create text node
    let noteText = document.createTextNode(inputVal);
    // append text to paragraph
    p.appendChild(noteText);
    // append new paragraph and text to existing note output
    noteOutput.appendChild(p);
  });
}
```

- DEMO 4 - travel notes - series I

HTML5, CSS, & JS - example - part 12.1

interaction - add a note - clear input

```
//overall app logic and loader...
function travelNotes() {
    "use strict";

    //handle user event for `add` button click
    $(".note-input button").on("click", function(e) {
        //object for wrapper html for note
        var $note = $("

");
        //define input field
        var $note_text = $(".note-input input");
        //conditional check for input field
        if ($note_text.val() !== "") {
            //set content for note
            $note.html($note_text.val());
            //append note text to note-output
            $(".note-output").append($note);
            $note_text.val("");
        }
    });
};

$(document).ready(travelNotes);


```

HTML5, CSS, & JS - example - part 12.2

interaction - add a note - clear input - plain JS

```
function travelNotes() {
  "use strict";

  // get a reference to `.note_output` in the DOM
  let noteOutput = document.querySelector('.note-output');
  // add note button
  let addNoteBtn = document.getElementById('add-note');
  // input field for add note
  let inputNote = document.getElementById('input-note');

  // add event listener to add note button
  addNoteBtn.addEventListener('click', () => {
    // create p node
    let p = document.createElement('p');
    // get value from input field for note
    let inputVal = inputNote.value;

    // check input value
    if (inputVal !== '') {
      // create text node
      let noteText = document.createTextNode(inputVal);
      // append text to paragraph
      p.appendChild(noteText);
      // append new paragraph and text to existing note output
      noteOutput.appendChild(p);
      // clear input text field
      inputNote.value = '';
    }
  });
}
```

- DEMO 5 - travel notes - series 1

HTML5, CSS, & JS - example - part 13.1

interaction - add a note - keyboard listener

- need to consider how to handle keyboard events
- listening and responding to a user hitting the return key in the input field
- similar pattern to user click on button

```
$(".note-input input").on("keypress", function (e) {  
  if (e.keyCode === 13) {  
    ...do something...  
  }  
});
```

- need to abstract handling both button click and keyboard press
- need to be selective with regard to keys pressed
- add a conditional check to our listener for a specific key
- use local variable from the event itself, eg: e, to get value of key pressed
- compare value of e against key value required

HTML5, CSS, & JS - example - part 13.2

interaction - add a note - keyboard listener - plain JS

```
// add event listener for keypress in note input field  
inputNote.addEventListener('keypress', (e) => {  
  // check key pressed by code - 13 - return  
  if (e.keyCode === 13) {  
    console.log('return key pressed...');  
  }  
});
```

- example recording keypresses
 - *Demo Editor*

HTML5, CSS, & JS - example - part 14

interaction - add a note - abstract code

- need to create a new function to abstract
 - *creation and output of a new note*
 - *manage the input field for our note app*
- moving logic from button click function to separate, abstracted function
- then call this function as needed
 - *for a button click or keyboard press*
 - *then create and render the new note*

```
//manage input field and new note output
function createNote() {
  //object for wrapper html for note
  var $note = $("

");
  //define input field
  var $note_text = $(".note-input input");
  //conditional check for input field
  if ($note_text.val() !== "") {
    //set content for note
    $note.html($note_text.val());
    //append note text to note-output
    $(".note-output").append($note);
    $note_text.val("");
  }
}


```


HTML5, CSS, & JS - example - part 15.1

interaction - add a note - travel.js

```
//overall app logic and loader...
function travelNotes() {
  "use strict";

  //manage input field and new note output
  function createNote() {
    //object for wrapper html for note
    var $note = $("

");
    //define input field
    var $note_text = $(".note-input input");
    //conditional check for input field
    if ($note_text.val() !== "") {
      //set content for note
      $note.html($note_text.val());
      //append note text to note-output
      $(".note-output").append($note);
      $note_text.val("");
    }
  }

  //handle user event for `add` button click
  $(".note-input button").on("click", function(e) {
    createNote();
  });

  //handle user event for keyboard press
  $(".note-input input").on("keypress", function(e){
    if (e.keyCode === 13) {
      createNote();
    }
  });
};
$(document).ready(travelNotes);


```

HTML5, CSS, & JS - example - part 15.2

interaction - add a note - plain JS

```
function travelNotes() {
  "use strict";

  // get a reference to `.note_output` in the DOM
  let noteOutput = document.querySelector('.note-output');
  // add note button
  let addNoteBtn = document.getElementById('add-note');
  // input field for add note
  let inputNote = document.getElementById('input-note');

  // add event listener to add note button
  addNoteBtn.addEventListener('click', () => {
    createNote(inputNote, noteOutput);
  });

  // add event listener for keypress in note input field
  inputNote.addEventListener('keypress', (e) => {
    // check key pressed by code - 13 - return
    if (e.keyCode === 13) {
      createNote(inputNote, noteOutput);
    }
  });
}

// create a note
// - input = value from input field
// - output = DOM node for output of new note
function createNote(input, output) {
  // create p node
  let p = document.createElement('p');
  // get value from input field for note
  let inputVal = input.value;
  // check input value
  if (inputVal !== '') {
    // create text node
    let noteText = document.createTextNode(inputVal);
    // append text to paragraph
    p.appendChild(noteText);
    // append new paragraph and text to existing note output
    output.appendChild(p);
    // clear input text field
    input.value = '';
  }
}
```

```
    }  
  }  
  
  // load app  
  travelNotes();
```

- DEMO 6 - travel notes - series I

HTML5, CSS, & JS - example - part 16

interaction - add a note - animate

- jQuery well-known for its simple ability to animate elements
- many built-in effects available in jQuery
 - *build our own as well*
- to `fadeIn` an element, effectively it needs to be hidden first
- we hide our newly created note
- then we can set it to `fadeIn` when ready
- many additional parameters for jQuery's `fadeIn` function
 - *customise a callback*
 - *change the speed of the animation*
 - *and so on...*
- jQuery API - `fadeIn`

HTML5, CSS, & JS - example - part 17

interaction - add a note - animate js

```
//manage input field and new note output
function createNote() {
  //object for wrapper html for note
  var $note = $("<p>");
  //define input field
  var $note_text = $(".note-input input");
  //conditional check for input field
  if ($note_text.val() !== "") {
    //set content for note
    $note.html($note_text.val());
    //hide new note to setup fadeIn...
    $note.hide();
    //append note text to note-output
    $(".note-output").append($note);
    //fadeIn hidden new note
    $note.fadeIn("slow");
    $note_text.val("");
  }
}
```

- DEMO 7 - travel notes - series I

HTML5, CSS, & JS - example - part 18

style and render notes

- we have some new notes in our app
- add some styling to help improve the look and feel of a note
- can set background colours, borders font styles...
- set differentiating colours for each alternate note
- allows us to try some pseudoclasses in the CSS
 - *specified paragraphs in the note-output section*

```
.note-output p:nth-child(even) {  
  background-color: #ccc;  
}  
.note-output p:nth-child(odd) {  
  background-color: #eee;  
}
```

- DEMO 8 - travel notes - series 1

References

- MDN - Prototype
- W3Schools - HTML5
 - *media elements*
 - *canvas element*
- W3Schools - Prototypes
- MDN JS - keyboard event
 - *event listener*
- MDN - CSS
 - *CSS documentation*
- W3
 - *CSS Flexible Box Layout Module 1*
- W3Schools - CSS
 - *CSS*

Travel Notes - Demos

- Travel notes app - series I
 - *DEMO 1 - travel notes - demo 1*
 - *DEMO 2 - travel notes - demo 2*
 - *DEMO 3 - travel notes - demo 3*
 - *DEMO 4 - travel notes - demo 4*
 - *DEMO 5 - travel notes - demo 5*
 - *DEMO 6 - travel notes - demo 6*
 - *DEMO 7 - travel notes - demo 7*
 - *DEMO 8 - travel notes - demo 8*