

Comp 324/424 - Client-side Web Design - Slides

Fall Semester 2017 - Week 11

Dr Nick Hayward

Contents

- ES6 Generators & Promises
- CSS
 - *grids*
 - *flex*
- HTML5, CSS, JS - example
- Working with APIs

ES6 Generators & Promises - promises - real-world promise - getJSON

```
// create a custom get json function
function getJSON(url) {
  // create and return a new promise
  return new Promise((resolve, reject) => {
    // create the required XMLHttpRequest object
    const request = new XMLHttpRequest();
    // initialise this new request - open
    request.open("GET", url);
    // register onload handler - called if server responds
    request.onload = function() {
      try {
        // make sure response is OK - server needs to return status 200 code...
        if (this.status === 200) {
          // try to parse json string - if success, resolve promise successfully with value
          resolve(JSON.parse(this.response));
        } else {
          // different status code, exception parsing JSON &c. - reject the promise...
          reject(this.status + " " + this.statusText);
        }
      } catch(e) {
        reject(e.message);
      }
    };

    // if error with server communication - reject the promise...
    request.onerror = function() {
      reject(this.status + " " + this.statusText);
    };

    // send the constructed request to get the JSON
    request.send();
  });
}
```

ES6 Generators & Promises - promises - real-world promise - usage

```
// call getJSON with required URL, then method for resolve object, and catch for error
getJSON("test.json").then(response => {
  // check return value from promise...
  response !== null ? "response obtained" : "no response";
}).catch((err) => {
  // Handle any error that occurred in any of the previous promises in the chain.
  console.log('error found = ', err); // not much to show due to return of jsonp from flickr...
});
```

- Demo - Promises - Basic XHR Local
- Demo - Promises - Basic CORS Flickr

ES6 Generators & Promises - promises - chain

- calling `then` on the returned promise creates a new *promise*
- if this promise is now resolved successfully
 - we can then register an additional callback
- we may now chain as many `then` methods as necessary
- create a sequence of promises
 - each resolved &c. one after another
- instead of creating deeply nested callbacks
 - simply chain such methods to our initial resolved promise
- to catch an error we may chain a final `catch` call
- to catch an error for the overall chain
 - use the *catch* method for the overall chain

```
getJSON().then()  
.then()  
.then()  
.catch((err) => {  
  // Handle any error that occurred in any of the previous promises in the chain.  
  console.log('error found = ', err); // not much to show due to return of jsonp from flickr...  
});
```

- if a failure occurs in any of the previous promises
 - the *catch* method will be called

ES6 Generators & Promises - promises - wait for multiple promises

- promises also make it easy to wait for multiple, independent asynchronous tasks
- with `Promise.all`, we may wait for a number of promises

```
// wait for a number of promises - all
Promise.all([
  // call getJSON with required URL, `then` method for resolve object, and `catch` for error
  getJSON("notes.json"),
  getJSON("metadata.json")]).then(response => {
  // check return value from promise...response[0] = notes.json, response[1] = metadata.json &c.
  if (response[0] !== null) {
    console.log("response obtained");
    console.log("notes = ", JSON.stringify(response[0]));
    console.log("metadata = ", JSON.stringify(response[1]));
  }
}).catch((err) => {
  // Handle any error that occurred in any of the previous promises in the chain.
  console.log('error found = ', err); // not much to show due to return of jsonp from flickr...
});
```

- order of execution for tasks doesn't matter for `Promise.all`
- by using the `Promise.all` method
 - we are simply stating that we want to wait...
- `Promise.all` accepts an array of promises
 - then creates a new promise
 - promise will resolve successfully when all passed promises resolve
- it will reject if a single one of the passed promises fails
- return promise is an array of succeed values as responses
 - i.e. one succeed value for each passed in promise
- Demo - Promises - Basic Promise All

ES6 Generators & Promises - promises - racing promises

- we may also setup competing promises
 - with an effective prize to the first promise to resolve or reject
 - might be useful for querying multiple APIs, databases, &c.

```
Promise.race([
  // call getJSON with required URL, `then` method for resolve object, and `catch` for error
  getJSON("notes.json"),
  getJSON("metadata.json")).then(response => {
    if (response !== null) {
      console.log(`response obtained - ${response} won...`);
    }
  }).catch((err) => {
    // Handle any error that occurred in any of the previous promises in the chain.
    console.log('error found = ', err); // not much to show due to return of jsonp from flickr...
  });
});
```

- method accepts an array of promises
 - returns a completely new resolved or rejected promise
 - returns for the first resolved or rejected promise
- Demo - Promises - Basic Race

ES6 Generators & Promises - promises - combine generators and promises

an example usage for generators and promises,

- `async` function takes a *generator*, calls it, and creates the required *iterator*
 - use *iterator* to resume generator execution as needed
 - declare a *handle* function - handles one return value from generator
 - one iteration of *iterator*
 - if generator result is a promise & resolves successfully - use *iterator's next* method
 - promise value sent back to generator
 - generator resumes execution
 - if error, promise gets rejected
 - error thrown to generator using *iterator's throw* method
 - continue generator execution until it returns *done*
- *generator* - executes up to each `yield` `getJSON()`
 - promise created for each `getJSON()` call
 - value is fetched *async* - generator is paused whilst fetching value...
 - control flow is returned to current invocation point in *handle* function whilst paused
- *handle* function
 - yielded value to *handle* function is a promise
 - able to use *then* and *catch* methods with promise object
 - registers success and error callback
 - execution is able to continue

ES6 Generators & Promises - lots of examples

e.g.

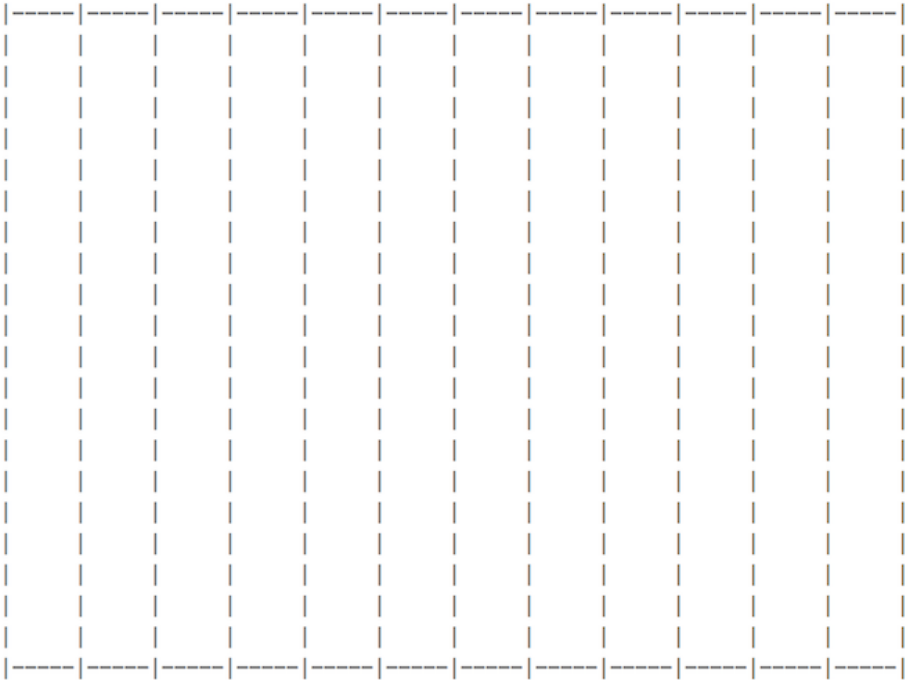
- generator
 - *basic*
 - *basic-iterator*
 - *basic-iterator-over*
 - *basic-loop*
 - *basic-dom*
 - *basic-send-data*
 - *basic-send-data-2*
- promises
 - *basic*
 - *basic-cors-flickr*
 - *basic-xhr-local*
 - *basic-promise-all*
 - *basic-promise-race*
- generator & promise - async
 - *basic*

CSS grid layout - part I

intro

- grid designs for page layout, components...
 - *increasingly popular over the last few years*
 - *useful for creating responsive designs*
- quick and easy to layout a scaffolding framework for our structured content
- create boxes for our content
 - *then position them within our grid layout*
- content can be stacked in a horizontal and vertical manner
 - *creating most efficient layout for needs of a given application*
- another benefit of CSS grids is that they are framework and project agnostic
 - *thereby enabling easy transfer from one to another*
- concept is based upon a set number of columns per page with a width of 100%
- columns will increase and decrease relative to the size of the browser window
- also set break points in our styles
 - *helps to customise a layout relative to screen sizes, devices, aspect ratios...*
 - *helps us differentiate between desktop and mobile viewers*

Image - Grid Layout



Grid Layout - Columns and rows

CSS grid layout - part 2

grid.css

- build a grid based upon 12 columns
 - *other options with fewer columns as well*
- tend to keep our grid CSS separate from the rest of the site
 - *maintain a CSS file just for the grid layout*
- helps abstract the layout from the remaining styles
 - *makes it easier to reuse the grid styles with another site or application*
- add a link to this new stylesheet in the head element of our pages

```
<link rel="stylesheet" type="text/css" href="assets/styles/grid.css">
```

or

```
<link rel="stylesheet" href="assets/styles/grid.css">
```

- ensure padding and borders are included in total widths and heights for an element
 - *reset `box-sizing` property to include the `border-box`*
 - *resetting box model to ensure padding and borders are included*

```
* {  
  box-sizing: border-box;  
}
```

CSS grid layout - example - part 3

grid.css

- set some widths for our columns, 12 in total
 - *each representing a proportion of the available width of a page*
 - *from a 12th to the full width of the page*

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

- classes allow us to set a column span for a given element
 - *from 1 to 12 in terms of the number of grid columns an element may span*

CSS grid layout - example - part 4

grid.css

- then set some further styling for each abstracted col- class

```
[class*="col-"] {  
  position: relative;  
  float: left;  
  padding: 20px;  
  border: 1px solid #333;  
}
```

- create columns by wrapping our content elements into rows
- each row always needs 12 columns

```
<div class="row">  
  <div class="col-6">left column</div>  
  <div class="col-6">right column</div>  
</div>
```

CSS grid layout - example - part 5

grid.css

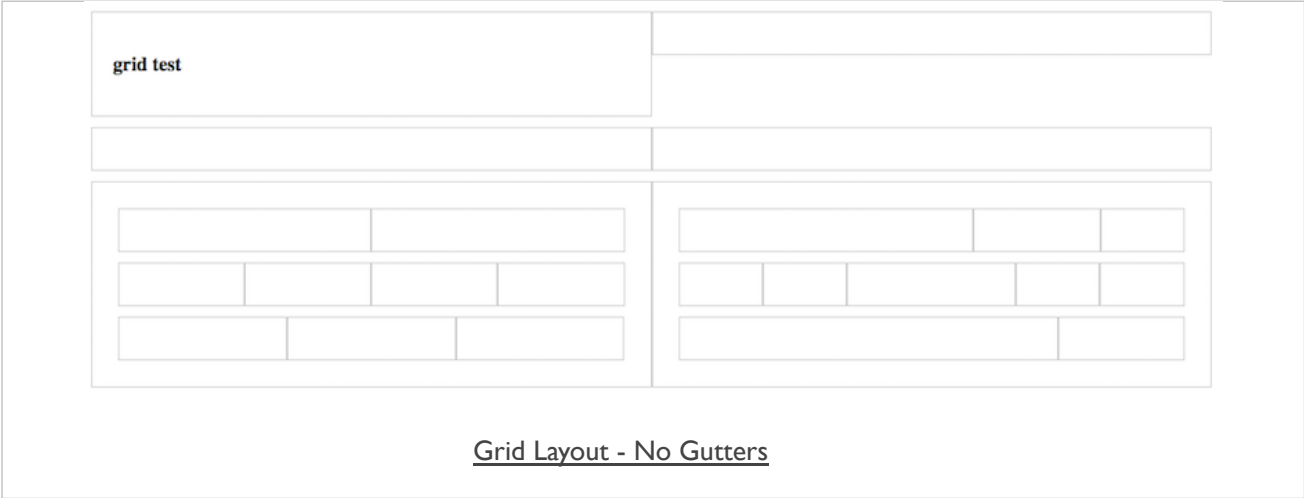
- due to the initial CSS of float left, each column is floated to the left
- columns are interpreted by subsequent elements in the hierarchy as non-existent
 - *initial placement will reflect this design*
- prevent this issue in layout, add the following CSS to grid stylesheet

```
.row:before, .row:after {  
  content: " ";  
  clear: both;  
  display: block;  
}
```

- benefit of the clearfix, clear: both
 - *make row stretch to include columns it contains*
 - *without the need for additional markup*

DEMO - Grid Layout I - no gutters

Image - Grid Layout I



CSS grid layout - example - part 6

grid.css

- add gutters to our grid to help create a sense of space and division in the content
- simplest way to add a gutter to the current grid css is to use padding
 - *rows can use padding, for example*

```
.row {  
  padding: 5px;  
}
```

- issue with simply adding padding to the columns
 - *margins are left in place, next to each other*
 - *column borders next to each with no external column gutter*
- fix this issue by targeting columns that are a sibling to a preceding column
- means we do not need to modify the first column, only subsequent siblings

```
[class*="col-"] + [class*="col-"] {  
  margin-left: 1.6%;  
}
```

Image - Grid Layout 2



CSS grid layout - part 7

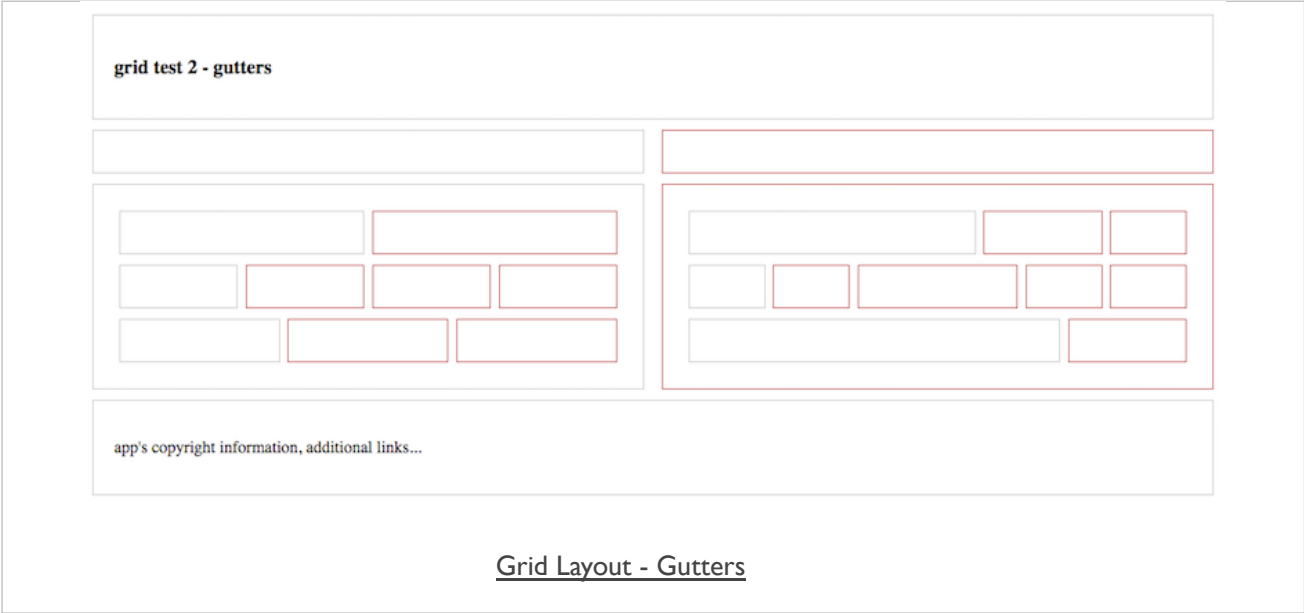
grid.css

- to fix this issue we recalculate permitted % widths for our columns in the CSS
 - we now have % widths as follows

```
.col-1 {width: 6.86%;}  
.col-2 {width: 15.33%;}  
.col-3 {width: 23.8%;}  
.col-4 {width: 32.26%;}  
.col-5 {width: 40.73%;}  
.col-6 {width: 49.2%;}  
.col-7 {width: 57.66%;}  
.col-8 {width: 66.13%;}  
.col-9 {width: 74.6%;}  
.col-10 {width: 83.06%;}  
.col-11 {width: 91.53%;}  
.col-12 {width: 100%;}
```

- DEMO - Grid Layout 2 - gutters

Image - Grid Layout 3



CSS grid layout - part 8

media queries

- often need to consider a mobile-first approach
- introduction of CSS3, we can now add **media queries**
- modify specified rulesets relative to a given condition
 - eg: screen size *for a desktop, tablet, and phone device*
- media queries allow us to specify a breakpoint in the width of the viewport
 - *will then trigger a different style for our application*
- could be a simple change in styles
 - *such as colour, font etc*
- could be a modification in the grid layout
 - *effective widths for our columns per screen size etc...*

```
@media only screen and (max-width: 900px) {  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```

- gutters need to be removed
 - *specifying widths of 100% for our columns*

```
[class*="col-"] + [class*="col-"] {  
  margin-left:0;  
}
```

Image - Grid Layout 4



Grid Layout - Media Queries

HTML5, CSS, & JS - example - part I

add grid layout

- update the layout of our Travel Notes application to include a grid layout
- apply this grid layout to the overall application
 - *organisation and presentation of the notes*
- remove the centred, fixed width for the body in our style.css stylesheet
- removes centre styling, results in content spanning full width of browser window
- add the grid layout to help us control this layout

```
<link rel="stylesheet" type="text/css" href="assets/styles/grid.css">
```

- then modify content categories, child elements to use new grid css

```
<!-- document header -->
<header>
  <div class="row">
    <div class="col-5">
      <h3>travel notes</h3>
      <h5>record notes from various places visited...</h5>
    </div>
    <div class="col-7"></div>
  </div>
</header>
```

Image - HTML5, CSS, & JS - grid layout

travel notes

record notes from various places visited...

add note

add

app's copyright information, additional links...

Grid Layout - Updated Header

HTML5, CSS, & JS - example - part 2

add grid layout

- update our main content to position the note-input and note-controls

```
<!-- note input -->
<section class="note-input">
  <div class="row">
    <div class="col-12">
      <h5>add note</h5>
      <input><button>add</button>
    </div>
  </div>
</section>
<!-- note controls for delete... -->
<section class="note-controls">
  <div class="row">
    <div class="col-12">
      <button id="notes-delete">Delete all</button>
    </div>
  </div>
</section>
```

- still need to amend style.css to remove additional fixed styling

Image - HTML5, CSS, & JS - grid layout 2

travel notes

record notes from various places visited...

add note

add

Delete all

note

app's copyright information, additional links...

Grid Layout - mixed grid and fixed

HTML5, CSS, & JS - example - part 3

add grid layout

- fix mixed rendering by removing width, margin, and padding for `.note-controls`

```
/* note controls */  
.note-controls {  
  border-bottom: 1px solid #dedede;  
  display: none;  
}
```

- continue to update Travel Notes app
 - *modify output for notes*
 - *add further options for users*

DEMO - Travel Notes - grid layout with media queries

HTML5, CSS, & JS - example - part 4

add flex to grid layout

- an additional option to consider - flex layouts
 - *a recent W3 working draft*
 - *aims to provide efficient way to align and proportion content*
- known as **Flexbox Layout**
 - *idea is to apportion width and height for content*
 - *proportions relative to container even when their size is unknown or dynamic*
- flex layout could, in theory, replace a full grid layout
 - *considered more a complement to overall grid structure*
- defined flex container expands items to fill the container's available space
 - *can also shrink them to prevent any possible overflow*
- think of a flex layout as supporting multiple directions
 - *direction agnostic*
- many properties available for **flex**
 - *focus upon styling flex container and any flex items*

HTML5, CSS, & JS - example - part 5

add flex to grid layout

- we might specify CSS properties for a flex container

```
.flex-container {  
display: flex; /* defines container as flex */  
flex-direction: row; /* defines positioning of items added to container */  
flex-wrap: wrap; /* defines whether to wrap items to another line */  
justify-content: flex-start; /* defines start point and distribution of items */  
}
```

- allows us to position our container starting at the left
 - *items contained in a row*
 - *contained items wrapping to additional lines if necessary*
- many additional options available for each property
- also add rulesets for specific styling of items within a flex container
- we could add properties to a flex item such as
 - *specify the order of the flex items*
 - *whether a particular item can grow or shrink relative to content*
 - *default size of an item before any remaining space is distributed*
 - *individual alignment for a given item...*

HTML5, CSS, & JS - example - part 6

add flex to notes

- flex container and items useful for organising and positioning our notes
- due to uncertainty about content, size, and general note requirements
 - *flex positioning and styling removes the need for assumptions or fixed sizes*
- we can start to modify the styling and rendering of our notes using flex

```
/* flex item */  
.flex-item {  
  flex-basis: 300px; /* default size before extra */  
  flex-grow: 1; /* all items will be equal */  
}
```

- gives us a default smallest size for each note
- then the ability for each note to grow to fill the row as required
- also work with responsive layouts
 - *due to the minimum size and the option to grow for each item*
 - *and wrap flex items per flex container*
- modify and update styles as we develop travel notes app

DEMO - Travel Notes - grid layout with flex notes

Image - HTML5, CSS, & JS - Flex Notes

travel notes

record notes from various places visited...

notes...

search...

add note

add

delete x

canon

lore ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse eget lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue frugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

canon

nice

monaco

antibes

lore ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse eget lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue frugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

juan les pins

lore ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse eget lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue frugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

ete

lore ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse eget lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue frugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

et tropez

app's copyright information, additional links...

Grid Layout - flex notes

Image - HTML5, CSS, & JS - Flex Notes 2

travel notes

record notes from various places visited...

menu...

search...

add note

add

delete all

cannes

monaco

nice

menton

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse egestas lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sod. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue feugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse egestas lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sod. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue feugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

antibes

st tropez

app's copyright information, additional links...

Grid Layout - flex notes - medium

Image - HTML5, CSS, & JS - Flex Notes 3

travel notes

record notes from various places visited...

memo...

search...

add note

add

Delete all

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse egestas lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue feugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

cannes

montaco

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus. Cras ac sagittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse egestas lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibus nulla, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam massa id sapien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. Pellentesque nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper lacus eu augue feugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum

antibes

app's copyright information, additional links...

Grid Layout - flex notes - small

HTML5, CSS, & JS - example - part 7

add flex to notes

Notes with Flex and Media Queries

HTML5, CSS, & JS - example - part I

add AJAX and JSON - load notes from json

- update our **travel notes** app to allow us to load some test persistent notes from a local JSON file
- initial JSON is as follows

```
{
  "travelNotes": [{
    "created": "2015-10-12T00:00:00Z",
    "note": "a note from Cannes..."
  }, {
    "created": "2015-10-13T00:00:00Z",
    "note": "a holiday note from Nice..."
  }, {
    "created": "2015-10-14T00:00:00Z",
    "note": "an autumn note from Antibes..."
  }]
}
```

HTML5, CSS, & JS - example - part 2

add AJAX and JSON - load notes from json

- add option to load notes from JSON as app initially loads
 - *use deferred promise pattern*
 - *checks source JSON as it loads via the promise*
 - *then outputs the end result*
- start with the following update

```
//get the notes JSON
function getNotes() {
  //.get returns an object derived from a Deferred object - do not need explicit deferred object
  var $deferredNotesRequest = $.getJSON (
    "docs/json/notes.json",
    {format: "json"}
  );
  return $deferredNotesRequest;
}
```

HTML5, CSS, & JS - example - part 3

add AJAX and JSON - load notes from json

- help us better manage logic of our notes from app's loading and execution
 - *create two separate JS files*
- our updated structure might be as follows

```
...  
|- assets  
  |- scripts  
    |- travel.js  
    |- notes.js  
...
```

- we can extend this further, as needed by app features and data

HTML5, CSS, & JS - example - part 4

add AJAX and JSON - load notes from json

- add our `.when()` function to the app's loader
 - *.when() function accepts a deferred object*
 - *in our case a limited promise*
- then allows us to chain additional deferred functions
 - *including required .done() function*
- for returned data, use standard response object to get `travelNotes`
 - *then iterate over the array for each property*
 - *for each iteration, we can simply call our `createNote` function*
 - *builds and renders required notes to the app's DOM*

```
//use deferred object from getJson
$.when(getNotes()).done(function(response) {
  //get travelNotes object
  var $travelNotes = response.travelNotes
  //process travelNotes array
  $travelNotes.forEach(function(item) {
    //check each property
    if (item !== null) {
      //get note
      var note = item.note;
      //create each note for rendering
      createNote(note);
    }
  }); //end foreach
});
```

HTML5, CSS, & JS - example - part 5

add AJAX and JSON - load notes from json

- simple problem - existing `createNote()` function does not accept a parameter
- need to update the logic of that function to accept and handle a parameter
- also requires a quick update to any functions and calls to the `createNote()`
 - *event handlers for creating a new note using the add button and keypress within the input field*

```
//manage input field and new note output
function createNote(data) {
  ...
  //conditional check for data
  if (data !== "") {
    //set content for note
    $note.html(data);
    ...
  }
}
```

HTML5, CSS, & JS - example - part 6

add AJAX and JSON - load notes from json

- update our event handlers for the note input button and input field keypress as follows,

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

```
//handle user event for keyboard press
$(".note-input input").on("keypress", function(e) {
    //check code for keyboard press
    if (e.keyCode === 13) {
        var $note_data = getNoteInput();
        //call note builder function
        createNote($note_data);
    }
});
```

- our notes now load by default as the app starts
- note input button and keypress work as expected
- DEMO - travel notes & JSON

Working with APIs - part I

remote api options - Flickr

- **Travel Notes** app loads data from a local JSON file
- add option to load different types of data using remote APIs
 - *Flickr API for images, tags...*
- basics and principles are similar to the patterns we've already seen and tested
- test a sample JSON return from the Flickr API
- JSON return - useful properties for app
 - *title*
 - *link*
 - *media (direct url for image - where available)*
 - *description*
 - ...
- public feed for searching public photos, videos, groups, recent activity...
- Flickr API Public Feed - Cannes and France

Working with APIs - part 2

working with Flickr API

- query Flickr's public feed for photos
 - we can use our now familiar pattern for requesting JSON

```
//get the Flickr public feed JSON for images
function getImages() {
  //.get returns an object derived from a Deferred object - do not need explicit deferred object
  var $deferredNotesRequest = $.getJSON (
    "http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?",
    { tags: "cannes,france,boules",
      tagmode: "all",
      format: "json"
    });
  return $deferredNotesRequest;
}
```

- need to make a few specific modifications to the request
 - JSONP to avoid browser security restrictions

Working with APIs - part 3

working with Flickr API

- Flickr's public feed includes options
 - eg: a specific user ID for photos, various tags, how tags are interpreted by the search...
- use our `.when()` function to load and render some test images from Flickr

```
$.when(getImages()).done(function(response) {
  console.log("done..." + response);
  //use jQuery's generic iterative function for the response...
  $.each( response.items, function( i, item ) {
    buildImage(item.media.m);
    //limit test images to 8
    if ( i === 7 ) {
      return false;
    }
  });
});
```

- DEMO - AJAX and JSON - Flickr api

HTML5, CSS, & JS - example - part 7

working with Flickr API - update travel notes

- add option to Travel Notes app to allow a user to view images from Flickr
- need to update app's HTML, CSS, and JS
- modify how our notes, and associated options, are rendered to our users
- add a search option for photos on Flickr
- render our images to match the notes
- app's structure still reflects three primary content categories
 - *header, main, and footer with slight modifications to the main category*
- main content category updated to create two distinct rows for initial content
 - *contain defined semantic containers*
- row containing `.note-input` and Flickr search option `.contextual-choice`
 - *then split this row into two columns of 6*

HTML5, CSS, & JS - example - part 8

working with Flickr API - update travel notes HTML

- updated HTML for .note-input and Flickr search .contextual-choice

```
<div class="row">
  <!-- note input -->
  <section class="note-input col-6">
    <h5>add note</h5>
    <input><button>add</button>
  </section>
  <!-- contextual choice -->
  <section class="contextual-choice col-6">
    <h5>search flickr</h5>
    <input><button>search</button>
  </section>
</div>
```

HTML5, CSS, & JS - example - part 9

working with Flickr API - update travel notes HTML

- update the HTML for rendering the images
 - *add alongside our notes*
- create another row for these containers
 - *add two section containers for `.note-output` and `.contextual-output`*
- make `.note-output` slightly larger to show primary app focus

```
<div class="row">
  <!-- note output -->
  <section class="note-output col-7 flex-container">
  </section>
  <!-- contextual output -->
  <section class="contextual-output col-5 flex-container">
  </section>
</div>
```

HTML5, CSS, & JS - example - part 10

working with Flickr API - update travel notes JS

- add further functionality to **Travel Notes** app
- split our JS logic into three files to help with organisation
 - a main loader file, *travel.js*,
 - and a file each for notes and contextual options
- updated app structure for JS

```
...  
  
|- assets  
  |- scripts  
    |- contextual.js  
    |- notes.js  
    |- travel.js  
...
```

- underlying logic for the notes will remain the same
 - move loading of default notes to the *travel.js* main loader file
- updates for searching, returning, and rendering images from Flickr
 - added to the *contextual.js* file

HTML5, CSS, & JS - example - part II

working with Flickr API - update travel notes JS

- test Flickr API in our app using some set data for image tags
 - respond to the user clicking on the search button
 - submit our query to Flickr
 - process the returned JSON for the images
 - render them for viewing
- request and process our images using the familiar pattern

```
//get the Flickr public feed JSON for images
function getImages(data) {
  var img_tags = data;
  ///.get returns an object derived from a Deferred object - do not need explicit deferred object
  var $deferredNotesRequest = $.getJSON (
    "http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?",
    { tags: img_tags,
      tagmode: "all",
      format: "json"
    });
  return $deferredNotesRequest;
}
```


HTML5, CSS, & JS - example - part 12

working with Flickr API - update travel notes JS

- returned data using standard deferred promise object
 - *add a new function to handle the processing of the images*

```
function processImages(data) {  
    $.when(getImages($img_data)).done(function(response) {  
        //use jQuery's generic iterative function for the response...  
        $.each( response.items, function( i, item ) {  
            createImage(item.media.m);  
            //limit test images to 4  
            if ( i === 3 ) {  
                return false;  
            }  
        });  
    });  
}
```

- using deferred promise object with `.when()` function chained to `.done()` function
- add jQuery's generic iterative function to help us process the response
 - *instead of standard JavaScript `.forEach()` option*
- loop through each value, and pass the image to our new function, `createImage()`
 - *ready for rendering to our app's DOM*
 - *limit number of images for testing*

HTML5, CSS, & JS - example - part 13

working with Flickr API - update travel notes JS

```
//manage new image output
function createImage(data) {
  //create each image element
  var img = $('<img class="flex-img">');
  //add image
  img.attr("src", data);
  //append to DOM
  $(".contextual-output").append(img);
}
```

- `.createImage()` function accepts a parameter for image data
- then process ready for rendering to the app's DOM
- image is added to a new `img` element with a new class of `.flex-img`
 - creates a *flex* item for rendering
- added to the new `.contextual-output` section
- rendered images displayed as thumbnails for the user
 - complementary to the existing notes

HTML5, CSS, & JS - example - part 14

working with Flickr API - update travel notes JS

- to add images to the app
 - a user can enter their requested tags in the search field
 - then click on the *search* button to return any available images
- event handler for this search button click uses the requested tags
 - passes them as a parameter to the *processImages ()* function

```
//handle user event for image `search` button click
$(".contextual-choice button").on("click", function(e) {
  //test tags for testing image search
  $img_data = "cannes,france,boules"
  //process images
  processImages($img_data);
});
```

Image - HTML5, CSS, & JS - Travel Notes & Flickr

travel notes

record notes from various places visited...

menu...

search...

add note

add

search flickr





search

Delete all

Cannes, a resort town on the French Riviera, is synonymous with glamour thanks to its world-famous film festival. Its Boulevard de la Croisette, curving along the coast, is lined with sandy beaches, upmarket boutiques and palatial hotels. It's also home to the Palais des Festivals, a modern building complete with red carpet and Allée des Stars – Cannes' walk of fame.

Nice, capital of the French Riviera, skirts the pebbly shores of the Baie des Anges. Founded by the Greeks and later a retreat for 19th-century Europe's elite, the city today balances old-world decadence with modern urban energy. Its sunshine and liberal attitude have long attracted artists, whose work hangs in its museums. With vibrant markets and diverse restaurants, it's also renowned for its food.

Antibes is a resort town between Cannes and Nice on the French Riviera (or Côte d'Azur). It's known for its Mediterranean beaches, annual Jazz à Juan music festival and old town enclosed by 16th-century ramparts. Luxury yachts moor at the huge Port Vauban marina, overlooked by star-shaped, 16th-century Fort Carré. The Promenade Amiral-de-Grasse walkway along Vauban's walls has views of the Alps.



app's copyright information, additional links...

Travel Notes & Flickr - test loading images

HTML5, CSS, & JS - example - part 15

working with Flickr API - update travel notes CSS

- need to update and modify existing CSS
 - *helps with correct rendering of the thumbnail images*
- CSS additions are initially modest
 - *reflects integration with existing app, grid, and flex layouts*
- add new ruleset for image rendering in the `.contextual-output` section

```
/* contextual output images */
.contextual-output img {
  margin: 5px;
  padding: 5px;
  border: 1px solid #b1c4b1;
}
```

- update `.flex-container` class to change `justify-content` property to value of `space-around`
- add new ruleset for a `.flex-img` class.

```
/* flex image */
.flex-img {
  flex-basis: 150px;
  flex-grow: 0;
}
```

- specify size of a thumbnail image
 - *initially restrict their ability to grow relative to flex*

HTML5, CSS, & JS - example - part 16

working with Flickr API - update travel notes JS

- we can now request, process, and render images from Flickr to Travel Notes app
 - *still need to accept and process search queries from search input field.*
- add option to check search input field
 - *then submit query to Flickr for images*

```
//get input value for image search
function getImageInput() {
  //define img value
  var img_val = "";
  //define input field
  var $img_tags = $(".contextual-choice input");
  if ($img_tags.val() !== "") {
    img_val = $img_tags.val();
    return img_val;
  } else {
    return img_val;
  }
}
```

HTML5, CSS, & JS - example - part 17

working with Flickr API - update travel notes JS

- use `getImageInput()` function with a modified `processImages()` function

```
//process image production, loading, and pass to rendering
function processImages() {
  //check img visibility for contextual-output - clear existing images
  if (checkVisible($(".contextual-output img")) === false) {
    //empty existing images
    $(".contextual-output").empty();
  }
  //get data from image search input field
  var $img_data = getImageInput();
  //use image data to get images, and pass for rendering
  $.when(getImages($img_data)).done(function(response) {
    console.log("done..." + response);
    //use jQuery's generic iterative function for the response...
    $.each( response.items, function( i, item ) {
      createImage(item.media.m);
      //limit test images to 4
      if ( i === 3 ) {
        return false;
      }
    });
  });
}
```

HTML5, CSS, & JS - example - part 18

working with Flickr API - update travel notes JS

- updated processImages () function then called within event handlers
 - for the search button and a keypress in the search input field

```
//handle user event for image search button click
$(".contextual-choice button").on("click", function(e) {
    //process images
    processImages();
});

//handle user event for keyboard press
$(".contextual-choice input").on("keypress", function(e) {
    //check code for keyboard press
    if (e.keyCode === 13) {
        //process images
        processImages();
    }
});
```

- DEMO - travel notes & Flickr

Image - HTML5, CSS, & JS - Travel Notes & Flickr

travel notes

record notes from various places visited...

add note

add

search flickr







search

Delete all

Curral das Freiras is a civil parish in the municipality of Câmara de Lobos in the Portuguese archipelago of Madeira. The population in 2011 was 2,001, in an area of 25.03 km². It is situated in the mountainous interior of the island.

Câmara de Lobos (Portuguese pronunciation: [literally, Portuguese: chamber of the wolves]) is a municipality, parish and city in the south-central coast of the island of Madeira. Technically a suburb of the much larger capital city of Funchal, it is one of the larger population centres and an extension of the Funchal economy.

Funchal is the largest city, the municipal seat and the capital of Portugal's Autonomous Region of Madeira. The city has a population of 111,892, making it the 6th largest city in Portugal, and has been the capital of Madeira for more than five centuries. Because of its high cultural and historical value, Funchal is one of Portugal's main tourist attractions. It is also popular as a destination for New Year's Eve, and it is the leading Portuguese port on cruise liner dockings.



app's copyright information, additional links...

Travel Notes & Flickr - different layout

HTML5, CSS, & JS - example - part 19

working with Flickr API - update travel notes JS

- room for improvement, updates, abstraction, and general refactoring of the existing code
- return to this issue when we consider refactoring the code in general
 - *there are still a few simple features we need to add*
- for example,
 - *add images to the `.contextual-output` section, resize `.note-output` section*
 - *moves focus to the current images*
 - *check loading progress of the notes and images*
 - *show feedback to the user*
 - *need to output a title for the images*
 - *set using the search query*

HTML5, CSS, & JS - example - part 20

working with Flickr API - modify travel notes JS

- first modification is to resize the `.notes-output`
 - *create more space for the images*
 - *gently shift focus to the new images*
- update existing `.createImage()` function in the `contextual.js` file

```
//manage new image output
function createImage(data) {
  ...
  if (checkVisible($(".contextual-output img")) === true) {
    $(".note-output").removeClass("col-12");
    $(".note-output").addClass("col-4");
    $(".contextual-output").fadeIn("slow");
  }
  ...
}
```

- add check to ensure images are not visible in the DOM
- remove current class from `.note-output` section
 - *12 column class for the grid*
- add new grid class to resize `.note-output` to 4 columns
 - *then fade in the `.contextual-output` class*
 - *set in the app's HTML to a class of `.col-8`*

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

working with Flickr API - modify travel notes JS

- next modification is some initial error handling
 - *checking for an empty array of images from the returned Flickr JSON*
- check `processImages ()` function for an empty array of image items

```
...  
  
if (response.items.length === 0) {  
    var img = "";  
    createImage(img);  
} else {  
    //return images from items array...  
}  
  
...
```

- checks images in the items array for the promise object
- if not, send an empty variable as a parameter to our `createImage ()` function

HTML5, CSS, & JS - example - part 22

working with Flickr API - modify travel notes JS

- check for empty value in `createImage()` function
 - *handle the simple errors as follows*

```
if (data !== "") {  
  //create each image element  
  var $img = $('<img class="flex-img">').attr("src", data);  
  //add image  
  img_output = $img;  
} else {  
  var $img_error = $('<p class="flex-item error">').html("No images available...");  
  //add error  
  img_output = $img_error;  
}
```

- we've abstracted the return variable for the image output
 - *can hold either the image or the error output...*
- add a check to see whether the `.contextual-output` section is visible or not
- modify the column class for the `.note-output` section
- then append our image output
- then show the `.contextual-output` section within the app
- DEMO - travel notes & Flickr

Image - HTML5, CSS, & JS - Travel Notes & Flickr

travel notes

record notes from various places visited...

menu...

search...

add note

search flickr

Delete all

Curral das Freiras is a civil parish in the municipality of Câmara de Lobos in the Portuguese archipelago of Madeira. The population in 2011 was 2,001, in an area of 25.03 km². It is situated in the mountainous interior of the island.

Câmara de Lobos (Portuguese pronunciation: [literally, Portuguese: chamber of the wolves]) is a municipality, parish and city in the south-central coast of the island of Madeira. Technically a suburb of the much larger capital city of Funchal, it is one of the larger population centres and an extension of the Funchal economy.

Funchal is the largest city, the municipal seat and the capital of Portugal's Autonomous Region of Madeira. The city has a population of 111,892, making it the 6th largest city in Portugal, and has been the capital of Madeira for more than five centuries. Because of its high cultural and historical value, Funchal is one of Portugal's main tourist attractions. It is also popular as a destination for New Year's Eve, and it is the leading Portuguese port on cruise liner dockings.

No images available. Please try a different search.

app's copyright information, additional links...

Travel Notes & Flickr - error checking

Demos

Travel notes app - series 3

- DEMO 1 - Travel notes - grid layout with media queries
- DEMO 2 - Travel notes - demo2

Travel notes app - series 4

- DEMO 1 - Travel Notes & JSON
- DEMO 2 - Travel Notes & Flickr
- DEMO 3 - Travel Notes & Flickr - error checking

Grids

- Grid 1 with no gutters
- Grid 2 with gutters

Promises

- Basic Promise All
- Basic Race

References - JS & Libraries

- Flickr API
 - *Public feeds*
 - *Public feed - public photos & video*
- jQuery
 - *jQuery*
 - *jQuery API*
 - *jQuery - deferred*
 - *jQuery - .getJSON()*
 - *jQuery - JSONP*
 - *jQuery :parent selector*
 - *jQuery - promise*
- Lint options
 - *JSLint - JavaScript Validator*
 - *JSONLint - JSON Validator*
- MDN
 - *MDN - JS*
 - *MDN - JS Const*
 - *MDN - JS Data Types and Data Structures*
 - *MDN - JS Grammar and Types*
 - *MDN - JS - Iterators and Generators*
 - *MDN - JS Objects*
 - *MDN - JS - Using Promises*
- W3
 - *W3 - CSS Flexible Box Layout Module I*
 - *W3 - JS Object*
 - *W3 - JS Performance*