Csci 4131 Internet Programming
Fall 2021
Lecture 7
September 29th

Instructor: Dr. Dan Challou

Logistics – Csci 4131 Lecture 7, Sept 29th

- Homework 2 Grades are posted on Canvas
- Questions or issues – follow the procedure documented in the syllabus (start by heading to an office hour with one of the TA's or emailing the help email:

  csci4131help-f21@umn.edu

Remember fastest ways to get questions answered or issues addressed are (in order of most rapid response to less rapid response) are

1) go to an virtual office hour
2) email the help email,
3) post to the discussions on Canvas

We'll eventually respond to emails sent to our x.500 email addresses – but that can take quite a bit more time (a day to a few (3-4) days or more…)

Logistics – Csci 4131 Lecture 7, September 29th

- ***The homework 3 specification is available in the week 4 module***
- ***Homework 3 is requires a significantly more complex solution than the previous 2 homework assignments***
- ***If you start it a day or two before it is due many of you will not complete most of the assignment.***
- ***It is YOUR responsibility to start the assignment early to ensure you get most of it completed***
- ***If you have not already read the assignment requirements specification, signed up for google maps APIs, obtained your Google API key, and started the google maps JavaScript API tutorial, you are behind.***

Upcoming Assignments /Due Dates

- zyBooks HW4 Saturday 10/2
- zyBooks Lecture Prep and HW for next week will be assigned tomorrow or Friday
- HW 3 due next Friday 10/10

Reading & Tutorials

Now:

Google Maps / JavaScript API:
https://developers.google.com/maps/documentation/javascript/tutorial

Google Maps Geocoding
https://developers.google.com/maps/documentation/javascript/geocoding,

Google Maps Places API
https://developers.google.com/maps/documentation/javascript/places

Google Directions Service
https://developers.google.com/maps/documentation/javascript/directions

Google Click on Points of Interest (used to fill location field on Form when points of interest are selected/clicked on the map next to it):
https://developers.google.com/maps/documentation/javascript/examples/event-poi

Optionals:

Sebesta - Chapter 5,6; and JavaScript tutorials (see course schedule in the module at the top of the home page on the Class Canvas site)

Homework 3- sign up for Google Maps

https://developers.google.com/maps/documentation/javascript/get-api-key

You must enable billing and give google a credit or debit card number

You get a 200 dollar credit for their services

You should use, at most very little of the credit for this assignment or follow-up assignments (20 dollars or less)

Email the class help email (csci4131help-f21@umn.edu) immediately if you have an issue with signing up, or if you somehow manage to incur charges to your account for work you do in this course.

Make sure to sign up for all the google API's and review Google's documentation and Examples on the following Services / APIs:

- Google Maps
- The Geocoding Library – for markers
- The Places Service – for searching for places
- The Directions Service
  - The Directions Display Object
  - The Directions Renderer Object

- Note, w3schools has tutorials to get you started as well
  - https://www.w3schools.com/graphics/google_maps_intro.asp

HW 3 – will make demo video avaliable

Questions?

Agenda

- Last Time
  - Lecture 6 Exercise Review
  - JavaScript
    - Animation
    - Automation
- Today
  - More JavaScript
    - More automation
    - Regular Expressions
    - JavaScript Closures

Lecture 7, Exercise 1 Review

- **Add a start and clear button to the clock we just built!**
  - Update the HTML to add the start and clear buttons
  - Clearclock (called by clear button) should call a JavaScript function to clear the text field
  - Start should start the clock anew.

  testclock.html

Hints:

The stop button calls **clearInterval** in the onclick event of the “Stop” Button

The “Clear Clock” button should set the “value” attribute of the text element to **“”** (the empty string)

Questions?

Recall our Random Image Rotator (from our last lecture)

- Random Pictures Original All Versions\RandomPicture.html

- How might we build an image (that is, advertisement) rotator?
- Automatically display the pictures in an recurring **sequence**…
- ***Why might we want such a capability for our webpages?***
- ***Example***
  - *[Random Pictures Original All Versions Final\RP3.html]*

Interactive Exercise (CODE ALONG)

- Download the file **RP3.html** from the week 5 module on Canvas, and all 7 **.png** files below it (in the file **pngPictures.zip**)

- And as you do that, I'll start writing JavaScript for displaying the pictures in sequence automatically!!

Questions?

Regular Expressions

- A **regular expression**, **regex** or regexp (sometimes called a rational **expression**) is, in theoretical computer science and formal language theory, a sequence of characters that **define** a search pattern.

- Usually this pattern is then used by string searching algorithms for "find" or "find and replace" operations on strings.

- https://www.regular-expressions.info/javascript.html

But, Regular Expressions have other uses!

- They can be used to check input values (in dialog boxes, text-input fields – on Forms, etc.)
- Can be used in conjunction with:
  - All of the above via
    - Pattern attribute associated with HTML 5 input elements (that is pattern="your regular expression")
    - JavaScript associated with the elements

Regular Expression methods

regex.test(str) – returns true if the str matches the pattern and false otherwise

regex.exec(str) – returns the first match for the regex in the string

Regular Expression Reference: https://www.w3schools.com/jsref/jsref_obj_regexp.asp

Examples

The JavaScript Expression:

/Dan[0-9]/.test(“The name Dan9 is like plan9”);

Returns **true**

What boolean value does:

/^Dan[0-9]/.test(“The name Dan9 is like plan9”)

Return?

Interactive Exercise: ***(Think / Pair / Share)***

- Create a regular expression that checks a string to see if it contains a phone number of the form:

Xxx-xxx-xxxx

where the first **X** is a number from 1 to 9, and the rest of the x's are numbers from 0 to 9. There should be no spaces before the first X in the match or after the last number

You can use this site to check out your expression:
https://www.regular-expressions.info/javascriptexample.html

Please close your computer when you are done!

Exercise 1 – Submit using the Lecture 7, Exercise 1 item in the week 4 module on Canvas-

- Download the file **phoneNumExEmptyForm.html** from the week 4 module on the Class Canvas site
- Use the regex from our Interactive Exercise to create an input form that accepts a SYNTACTICALLY VALID telephone number of the form: **xxx-xxx-xxxx** by updating the file **phoneNumExEmptyForm.html**
- Demo – phoneNumEx2.html
- Hint: Set the **pattern** attribute (that is **pattern="regex"**) in a text input field used to enter the phone number to your regular expression (in double quotes) – replace the text in the double quotes in the form with your regex:

  "insert regex from Interactive Exercise on previous slide"

As you have seen already, HTML input elements (tag) have a pattern element – check w3 Schools (or zyBooks)

Please close your computer when you are done (or ready to move on)!!!

Other ways to check and obtain/change values using regular expressions

- String methods
  - match
  - search
  - replace
  - Lots of others….

  See:

  https://www.w3schools.com/jsref/jsref_obj_string.asp

Questions?

Closure (Computer Programming)

source:
https://en.wikipedia.org/wiki/Closure_(computer_programming)

- In programming languages, **closures** (also **lexical closures** or **function closures**) are techniques for implementing lexically scoped name binding in languages with first-class functions.
- Operationally, a closure is a record storing a function together with an environment: a mapping associating each free variable of the function (variables that are used locally, but defined in an enclosing scope) with the value or reference to which the name was bound when the closure was created.
- A closure—unlike a plain function—allows the function to access those *captured variables* through the closure's copies of their values or references, even when the function is invoked outside their scope.

JavaScript Scope and Closures

- https://robertnyman.com/2008/10/09/explaining-javascript-scope-and-closures/
- Closures are expressions, usually functions, which work with variables ***set at the time the function is called*** within a certain context.
- More specifically, inner function(s) that refer to local variables of their outer function create closures.

For Example: What is Displayed in the Alert Box??? (Please venture an answer)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8">
    <title>Example of simple function closure</title>
          <script>
                function addN (x) {  // this returns a closure
                    return function (y) {
                        return x + y;
                    };
                };

                var add3 = addN(3);  // addN(3) creates a function  with x bound to 3
                                     // and returns it, and sets the identifier add3 to refer to it
                var result = add3(5);
                alert(result);
          </script>
  </head>
  <body>
  </body>
</html>
```

simpleclosure.html

Interactive Exercise: Download the file: BadClosure.html from the week 5 module on the class Canvas site, run it, and view source – RAISE HAND WHEN YOU HAVE IT

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset = "utf-8">
        <title>Example of incorrect function closures</title>
        <script> // what does the DOM look like after this page is loaded????
            function addLinks () {
                for (var i=1, link; i<6; i++) {
                    link = document.createElement("a");
                    link.innerHTML = "      Link" + i + "<br><br>";
                    link.onclick = function () {
                            alert("This is link: " + i);
                        };
                    document.body.appendChild(link);
                } // end for
            } // end addLinks
            window.onload = addLinks;
        </script>
    </head>
    <body>
    </body>
</html>
```

What does the BadClosure.html page "do"?

- BadClosure.html

What should it do? (Your feedback requested)

So, how do we figure out what went, or is going wrong in JavaScript

- Use debugging constructs!
- Use a debugger!
- http://www.w3schools.com/js/js_debugging.asp
- https://developers.google.com/web/tools/chrome-devtools/javascript/breakpoints

Some ways to Fix it!

- "Correctly" working versions of the BadClosure.html Web Page

- Properly working Version 1
- Properly working version 2

Event Bubbling

- The process whereby events fired on *child* elements "bubble" up to their *parent* elements
- When an event is fired on an element, it is first delivered to the element's event handler (if any), then to the parent element's event handler (if any), then to its parent's parent, etc.
  - *If you are handling an event in a child element alone, you should cancel the bubbling of the event in the child element's event-handling code by using the `cancelBubble` property of the event object*

Overview of HTML Event Types

- Load – triggers when the browser loads all the content in the document. Works the same as window.onload event
- Unload – triggers when browser removes a document from the window
- Submit – triggers when a form is submitted
- Reset – trigger when a form is reset
- Select – triggers when a user selects text in field
- Change – triggers when content of an element is changed
- Focus – triggers when an element gains focus
- Blur – triggers when an element loses focus

Next Time

- JavaScript Closures - revisted
- JavaScript Event Handling Wrapped up
- Intro to Google maps