Information Visualization

W02: JavaScript Programming

Graduate School of System Informatics

Department of Computational Science

Naohisa Sakamoto　Akira Kageyama

April 13, 2021

Schedule

- W01 4/12 Guidance
- W02 4/13 JavaScript Programming
- W03 4/19 Data Visualization
- W04 4/20 Reading Data
- W05 4/26 Marks and Channels
- W06 4/27 Creating Data Plot - Scatter plot
- W07 5/10 Visualization Idioms
- W08 5/11 Creating Data Plot - Bar/Pie/Line/Area chars

Working Directory

- Move to the local GitHub repository

```
$ cd ~/Work/InfoVis2022
```

- Create a today's working directory

```
$ mkdir W02
```

- Move to the working directory

```
$ cd W02
```

JavaScript Code

- Template

```
<html>
    <head>

    </head>
    <body>
        <script>

                JavaScript code ...

        </script>
    </body>
</html>
```

Example 01

- “Hello World”
  - Write a text directory to the HTML document.
  - document.write()

w02_ex01.html

```
<html>
        <head>
                <title>W02: Example 01</title>
        </head>
        <body>
                <script>
                        document.write(“Hello World!”);
                </script>
        </body>
</html>
```

Example 02

- “Hello World”
  - Write a text to the browser console.
  - console.log()

w02_ex02.html

```
<html>
        <head>
                <title>W02: Example 02</title>
        </head>
        <body>
                <script>
                        console.log(“Hello World!”);
                </script>
        </body>
</html>
```

Example 03

- “Hello World”
  - Write a text to an alert box.
  - window.alert()

w02_ex03.html

```
<html>
	<head>
		<title>W02: Example 03</title>
	</head>
	<body>
		<script>
			window.alert(“Hello World!”);
		</script>
	</body>
</html>
```

Example 04

- “Hello World”
  - Write a text to an HTML element.
  - innerHTML

w02_ex04.html

```
<html>
        <head>
                <title>W02: Example 04</title>
        </head>
        <body>
                <p id=“target”></p>
                <script>
                document.getElementById(“target”).innerHTML
                                = “Hello World!”;
                </script>
        </body>
</html>
```

Variables

- Variables in JS are container for storing data values.

Example
```
var x = 1;
var y = 2;
var z = x + y;
```

- Block scope variables and constants

Example
```
let x = 1;
const y = 2;
```

http://www.w3schools.com/js/js_variables.asp
http://www.w3schools.com/js/js_let.asp
http://www.w3schools.com/js/js_const.asp

Operators

- Arithmetic operators

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

http://www.w3schools.com/js/js_operators.asp

Operators

- Assignment operators

Operator	Example	Same as
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

http://www.w3schools.com/js/js_operators.asp

Operators

- Comparison and logical operators

Operator	Description
==	Equal to
===	Equal value and equal type
!=	Not equal
!==	Not equal value or not equal type
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
?	Ternary operator

http://www.w3schools.com/js/js_operators.asp

Data types

- Variables can hold many data types: numbers, strings, arrays, objects and more:

Example

```
var length = 16; // Number
var is_male = true; // Boolean
var name = "Johnson"; // String
var cars = ["Saab", "Volvo", "BMW"]; // Array
var p = {first_name:"John", last_name:"Doe"}; // Object
```

http://www.w3schools.com/js/js_datatypes.asp

Conditional statements

- if, else if, else statements

Syntax

```
if ( condition1 )
{
        block of code to be executed
        if condition1 is true
}
else if ( condition2 )
{
        block of code to be executed
        if the condition1 is false and condition2 is true
}
else
{
     block of code to be executed
        if the condition1 is false and condition2 is false
}
```

http://www.w3schools.com/js/js_if_else.asp

Switch statement

- switch statement

Syntax

```
switch ( expression )
{
    case n:
        code block
        break;
    case n:
        code block
        break;
    default:
        default code block
}
```

For loop

- Loops with ‘for’

Syntax

```
for ( statement 1; statement 2; statement 3 )
{
    code block to be executed
}
```

Example

```
var text;
for ( i = 0; i < 5; i++ )
{
    text += "The number is " + i + "<br>";
}

document.getElementById(“target”).innerHTML = text;
```

While loop

- Loops with ‘while’

Syntax

```
while ( condition )
{
    code block to be executed
}
```

Example

```
var text;
while ( i < 5 )
{
    text += "The number is " + i + “<br>”;
    i++;
}

document.getElementById(“target”).innerHTML = text;
```

Function (1/4)

- A function is defined by using ‘function’.

Syntax

```
function name( parameter1, parameter2 )
{
    code block to be executed
}
```

Example

```
function MyFunc()
{
       var text;
       while ( i < 5 )
       {
              text += "The number is " + i + “<br>”;
              i++;
       }
       document.getElementById(“target”).innerHTML = text;
}
```

Function (2/4)

- Return statement

Example

```
var x = Add( 4, 3 );

function Add( a, b )
{
    return a + b;
}
```

Function (3/4)

- Definition of the function

w02_ex05.html

```
<html>
        <head>
                <title>W02: Example 05</title>
        </head>
        <body>
                <script>
                        function Add( a, b ){ return a + b; }
                        var x = Add( 4, 3 );
                        document.write( x );
                </script>
        </body>
</html>
```

Function (4/4)

- Definition in an external file

w02_ex06.html

```
<html>
	<head>
		<title>W02: Example 06</title>
	</head>
	<body>
		<script src=“add.js”></script>
		<script>
			document.write( Add( 4, 3 ) );
		</script>
	</body>
</html>
```

add.js

```
function Add( a, b )
{
	Return a + b;
}
```

Arrow Function

- Arrow functions allow us to write shorter function syntax.

Example

```
function Add( a, b ) { return a + b; }
Add = function( a, b ) { return a + b; }

// with arrow function
let Add = ( a, b ) => { return a + b; }
let Add = ( a, b ) => a + b;
```

Class (1/3)

- A class is defined by using ‘function’.
  - Ex.) Vec3 class

vec3.js

```
// Constructor
Vec3 = function( x, y, z )
{
        this.x = x;
        this.y = y;
        this.z = z;
}
```

Class (2/3)

- A method is defined by using ‘prototype’.

vec3.js

```
// Add method
Vec3.prototype.add = function( v )
{
        this.x += v.x;
        this.y += v.y;
        this.z += v.z;
        return this;
}

// Sum method
Vec3.prototype.sum = function()
{
        return this.x + this.y + this.z;
}
```

Class (3/3)

- Use case of Vec3 class

w02_ex07.html

```
<html>
	<head>
		<title>W02: Example 07</title>
	</head>
	<body>
		<script src=“vec3.js”></script>
		<script>
			var v1 = new Vec3( 5, 4, 8 );
			var v2 = new Vec3( 2, 1, 7 );
			var v = v1.add( v2 ); // v = (7,5,15)
			var sum = v.sum(); // 27 = 7 + 5 + 15
		</script>
	</body>
</html>
```

New Class Definition (1/2)

- The keyword ‘class’ can be used for class definition in ES6.

Example

```
class Vec3
{
    // Constructor
    constructor( x, y, z )
    {
        this.x = x;
        this.y = y;
        this.z = z;
    }
}
```

New Class Definition (2/2)

- Methods

Example

```
class Vec3
{
    // Constructor
    ...

    add( v )
    {
        this.x += v.x;
        this.y += v.y;
        this.z += v.z;
        return this;
    }
}
```

Input

- type="button"

Syntax

```
<input type="button"
       onclick="event"
       value="label"/>
```

Example

```
<input type="button"
       onclick="window.alert('Clicked!')"
       value="Click Me"/>
```

http://www.w3schools.com/html/html_form_input_types.asp

Input

- type="button"

w02_ex08.html

```
<html>
        <head>
                <title>W02: Example 08</title>
        </head>
        <body>
                <input  type="button"

        onclick="window.alert('Clicked!')"
                                value="Click Me"/>
        </body>
</html>
```

Input

- type=“button”

w02_ex08.html

```
<html>
        <head>
                <title>W02: Example 08</title>
        </head>
        <body>
                <input type=“button”
                               onclick=“e()”
                               value=“Click Me”/>
                <script>
                function e() { window.alert(‘Clicked!’); }
                </script>
        </body>
</html>
```

Input

- type="button"

Example

```
<input  type=“button”
              value=“Click Me”
              id=“click_me”/>
<script>
       var element = document.getElementById(‘click_me’);
       element.addEventListener(‘click’, e );
       function e() { window.alert(‘Clicked!’); }
</script>
```

Input

- type=“button”

Example

```
<input type=“button”
              value=“Click Me”
              id=“click_me”/>
<script>
       document.getElementById(‘click_me’)
              .addEventListener(‘click’, function () {

       window.alert(‘Clicked!’);
              });
</script>
```

Input

- type=“text”
- type=“radio”
- type=“checkbox”
- type=“number”
- type=“color”
- type=“range”
- …

http://www.w3schools.com/html/html_form_input_types.asp

Task 1

- Redefine the class of Vec3 by using the keyword ‘class’.
  - This class need to be used for implementing Tasks 2 and 3 in the next slides.

Task 2

- Implement the following methods in Vec3 class and show the result on the web browser.
  - min(): Returns a min. value of the elements
  - mid(): Returns a mid. value of the elements
  - max(): Returns a max. value of the elements

Example

```
var x = 5, y = 4, z = 8; // (input values)
var v = new Vec3( x, y, z );
var min = v.min(); // 4 (output value)
var mid = v.mid(); // 5 (output value)
var max = v.max(); // 8 (output value)
```

Task 3

- Calculate the area of a triangle given the coordinates of the three vertices, and implement user interfaces for inputting values and showing the result with <input> elements.

Example

```
var x0, y0, z0; // (input vertex 0)
var x1, y1, z1; // (input vertex 1)
var x2, y2, z2; // (input vertex 2)
var v0 = new Vec3( x0, y0, z0 );
var v1 = new Vec3( x1, y1, z1 );
var v2 = new Vec3( x2, y2, z2 );
var S = AreaOfTriangle( v0, v1, v2 ); // (output value)
```

Polling

- Take the poll
  - Student ID Number
  - Name
  - URL to Task 1
    - e.g. https://xxx.github.io/InfoVis2022/W02/vec3.js
  - URL to Task 2
    - e.g. https://xxx.github.io/InfoVis2022/W02/task2.html
  - URL to Task 3
    - e.g. https://xxx.github.io/InfoVis2022/W02/task3.html

xxx = GitHub account name

Submission URL

- Submit URL to **GitHub Pages** not **Repository**

🙆‍♂️ https://**YourAccountName.github.io**/...

e.g.) https://**vizlab-kobe-lecture.github.io**/InfoVis2022/W02/task02.html

🙅‍♂️ https://**github.com/YourAccountName**/...

e.g.) https://**github.com**/**vizlab-kobe-lecture**/InfoVis2022/blob/master/W02/task2.html

Check the task page on your browser before submission!