

Information Visualization

W02: Exercise - Setup

Graduation School of System Informatics

Department of Computational Science

Naohisa Sakamoto, Akira Kageyama

Apr. 12, 2017

Working directory

- Move to the local GitHub repository

```
$ cd ~/Work/InfoVis2017
```

- 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()

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

w02_ex01.html

Example 02

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

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

w02_ex02.html

Example 03

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

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

w02_ex03.html

Example 04

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

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

w02_ex04.html

Variables

- Variables in JS are container for storing data values.

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

Example

Operators

- Arithmetic operators

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

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

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

Data types

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

```
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
```

Example

Conditional statements

- if, else if, else statements

```
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  
}
```

Syntax

Switch statement

- switch statement

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

Syntax

For loop

- Loops with 'for'

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

Syntax

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

Example

While loop

- Loops with 'while'

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

Syntax

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

Example

Function (1/4)

- A function is defined by using ‘function’.

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

Syntax

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

Example

Function (2/4)

- Return statement

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

Example

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

Function (3/4)

- Definition of the function

```
<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>
```

w02_ex05.html

Function (4/4)

- Definition in an external file

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

w02_ex06.html

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

add.js

Class (1/3)

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

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

vec3.js

Class (2/3)

- A method is defined by using 'prototype'.

```
// 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;
}
```

vec3.js

Class (3/3)

- Use case of Vec3 class

```
<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>
```

w02_ex07.html

Input

- type="button"

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

Syntax

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

Example

Input

- type="button"

```
<html>
  <head>
    <title>W02: Example 08</title>
  </head>
  <body>
    <input type="button"
           onclick="window.alert('Clicked!')"
           value="Click Me"/>
  </body>
</html>
```

w02_ex08.html

Input

- type="button"

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

w02_ex08.html

Input

- type="button"

```
<input type="button"  
       value="Click Me"  
       id="click_me"/>
```

```
<script>
```

```
    var element = document.getElementById('click_me');
```

```
    element.addEventListener('click', event );
```

```
    function event() { window.alert('Clicked!'); }
```

```
</script>
```

Example

Input

- type="button"

```
<input type="button"
       value="Click Me"
       id="click_me"/>
<script>
  document.getElementById('click_me')
    .addEventListener('click', function () {
      window.alert('Clicked!');
    });
</script>
```

Example

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

- 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

```
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)
```

Example

Task 2

- 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.

```
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)
```

Example

Polling

- Take the poll
 - Student ID Number
 - Name
 - URL to Task 1
 - e.g. <https://xxx.github.io/InfoVis2017/W02/task1.html>
 - URL to Task 2
 - e.g. <https://xxx.github.io/InfoVis2017/W02/task2.html>

xxx = GitHub account name