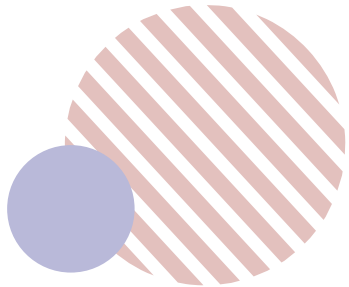
Web Programming

JavaScript Objects

Instructor: Prof. SoYeop Yoo
School of Computing, Gachon University

CONTENTS

Objects



Note

In this part of the class

- The aim is to give you just enough working knowledge of objects so that you will be able to use objects in web development
- You will learn more fully about objects and object-oriented programming in the Java Programming course

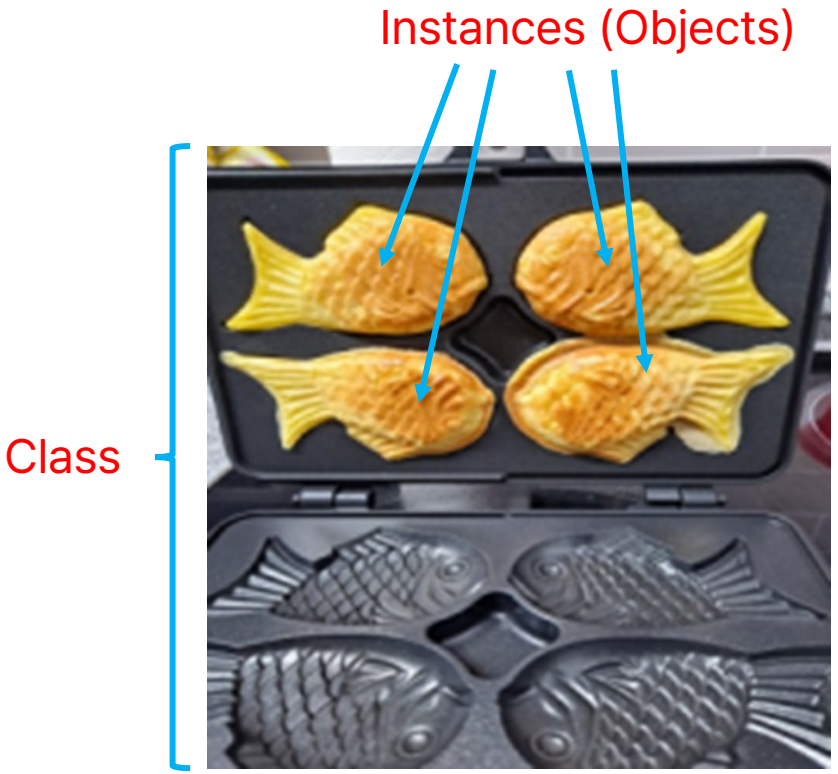
Class and Instance (Object) (1/3)

Class and Instance

- A class is a template for objects that share the same properties and programs
- Terminology
  - properties, field, attribute: holds data (value)
  - method, function: program that operates on the data
- All objects created from the same class share the same properties and methods
- However, each object may have different values for the same properties

Class and Instance (Object) (2/3)

- Relationship between a class and its instances



이름	나이	입사 년도
김○○	23	2017
이○○	21	2018
정○○	26	2014
김○○	24	2015
최○○	20	2019

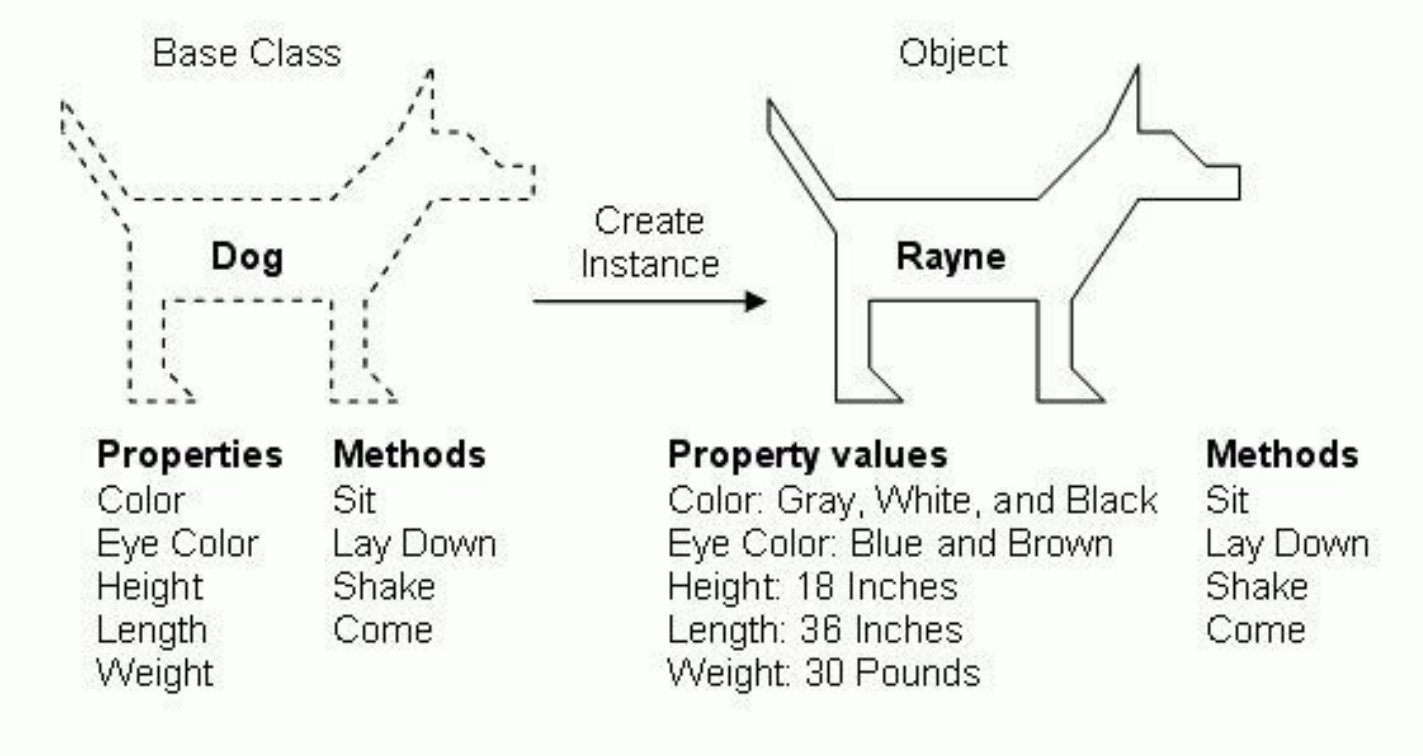
attributes

instance

class

Class and Instance (Object) (3/3)

Object and Methods



JavaScript Object (1/3)

Creating a JavaScript Object

- With JavaScript, you can define and create your own objects
  - Define and create a single object, using an object literal

```
var person = {
    firstName:"John",
    lastName:"Doe",
    age:50,
    eyeColor:"blue"
};
```

  - Define and create a single object, with the keyword new

```
var person = new Object();
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";
```

JavaScript Object (2/3)

Accessing Properties and Methods

- The syntax for accessing the property of an object is:
  - objectName.property // person.age
    - e.g., person.firstname + " is " + person.age + " years old.";
  - objectName["property"] // person["age"]
    - e.g., person["firstname"] + " is " + person["age"] + " years old.";
- You access an object method with the following syntax:
  - objectName.methodName()
    - e.g., name = person.fullName();

JavaScript Object (3/3)

Type of Objects

- JavaScript objects
  - String, Boolean, Number, Array
  - Date, Math, RegExp, Global
- Browser objects
  - Window, Screen
  - Location, History, Navigator
- HTML DOM objects
  - Document
  - Events, elements, attributes

String Objects (1/4)

String Objects

- Used to manipulate a piece of text
- Created with a keyword new
  - var txt = new String();
- Properties
  - length: returns the length of a string
- Methods
  - charAt(n), concat(...), indexOf(string), substr(start.length), toLowerCase(), toUpperCase(), match(regExp), replace(regExp, replacement), search(regExp), etc.
  - See: http://www.w3schools.com/js/js_string_methods.asp

String Objects (2/4)

string.length

- Returns the number of characters in a string
- `text="Hello, World"; text.length;` → 12

string.charAt(n)

- Returns the n-th character of a string (n>=0)
- `str="HELLO WORLD"; str.charAt(2);` → L

string.concat(string1, ..., string)

- Concatenates the string arguments to the recipient string
- `str1="Hello "; str2="World!"; str3="Good day!";`
  `str=str1.concat(str2, str3);` → "Hello World! Good day!"

String Objects (3/4)

string.indexOf(substring)

- Returns the position of the first character of substring in the recipient string, or -1 if not found
- `str="Hello World, Welcome to the universe!";`
  `str.indexOf("e");` → 1
  `str.indexOf(“World”);` → 6
  `str.indexOf(“z");` → -1

string.substr(start,length)

- Returns a substring of the specified length, beginning at the specified position
- `str="Hello World!"; str.substr(2,3);` → "llo"

String Objects (4/4)

string.toUpperCase()

- Converts a string to uppercase letters
- `str="Hello World!";`
  `str.toUpperCase();` → "HELLO WORLD!"

string.toLowerCase()

- Converts a string to lowercase letters
- `str=”HELLO WORLD!";`
  `str.toLowerCase();` → ”hello world!"

Boolean Objects

Boolean Objects

- Creation
  - var myBoolean = new Boolean(true);
- Boolean methods
  - toString()
    - converts a Boolean value to a string
  - valueOf()
    - returns the primitive value of a Boolean object

```
<script>
   var myBoolean = new Boolean(true);
   document.write(myBoolean.toString());   // "true"
   var boolValue = myBoolean.valueOf();    // true
</script>
```

Number Objects (1/2)

Number Objects

- Creation
  - var num = new Number(value);
- Number properties
  - documents.write(Number.MAX_VALUE);

Property	Description
MAX_VALUE	Returns the largest number possible in JavaScript (1.7976931348623157e+308)
MIN_VALUE	Returns the smallest number possible in JavaScript (5e-324)
NEGATIVE_INFINITY	Represents negative infinity (returned on overflow)
NaN	Represents a "Not-a-Number" value
POSITIVE_INFINITY	Represents infinity (returned on overflow)

Number Objects (2/2)

Number Methods

Method	Description
toExponential(x)	Converts a number into an exponential notation
toFixed(x)	Formats a number with x numbers of digits after the decimal point
toPrecision(x)	Formats a number to x length
toString()	Converts a number to a string
valueOf()	Returns the primitive value of a number

```
<script>
   var num1 = new Number(273.52233);
   document.write(num1 + " " + typeof(num1));
   document.write(num1.toExponential() + " " + num1.toFixed(2));
   var str1 = num1.toString();
   document.write(str1 + " " + typeof(str1));
</script>
```

```
273.52233 object
2.7352233e+2 273.52
273.52233 string
```

Array Objects (1/3)

Array Objects

- Used to store multiple values in a single variable
- Array property
  - length: sets or returns the number of elements in an array

```
<script>
   var myCars = new Array();
   myCars[0] = "Hyundai";
   myCars[1] = "Nissan";
   myCars[2] = "BMW";
   var myCars = new Array ("Hyundai", "Nissan", "BMW");
   var myCars = ["Hyundai", "Nissan", "BMW"];
   document.write(myCars);
</script>
```

```
Hyundai,Nissan,BMW
```

Array Objects (2/3)

Array Methods

- See: http://www.w3schools.com/jsref/jsref_obj_array.asp

Method	Description
concat()	Joins two or more arrays, and returns a copy of the joined arrays
indexOf()	Search the array for an element and returns its position
join()	Joins all elements of an array into a string
pop()	Removes the last element of an array, and returns that element
push()	Adds new elements to the end of an array, and returns the new length
reverse()	Reverses the order of the elements in an array
shift()	Removes the first element of an array, and returns that element
slice()	Selects a part of an array, and returns the new array
sort()	Sorts the elements of an array (alphabetical only)
splice()	Adds/Removes elements from an array
toString()	Converts an array to a string, and returns the result
unshift()	Adds new elements to the beginning of an array, and returns the new length
valueOf()	Returns the primitive value of an array

Array Objects (3/3)

Example

```
<script>
   var fruits = ["Orange", "Apple", "Mango"];
   document.write(fruits);
   document.write(fruits.join());
   document.write(fruits.join("+"));
   document.write(fruits.join("and"));
   fruits.sort();
   document.write(fruits);
   document.write();

   var points = [40, 100, 1, 5, 25, 10];
   points.sort();
   document.write(points);
   points.sort(function(a,b) {return a-b;});
   document.write(points);
   points.sort(function(a,b) {return b-a;});
   document.write(points);
</script>
```

```
Orange,Apple,Mango
Orange,Apple,Mango
Orange+Apple+Mango
OrangeandAppleandMango
Apple,Mango,Orange

1,10,100,25,40,5
1,5,10,25,40,100
100,40,25,10,5,1
```

JavaScript Variables vs. Objects (1/2)

Variables

- Data types: undefined, Boolean, number, string, function

Objects

- Everything else is an object!
  - String, Number, Boolean, Array, Date, Math, ...
- A string variable and a String object are different

```
<script>
   var a = 23, b = true, c = "abc";
   document.write(typeof(a)+", "+typeof(b)+", "+typeof(c));

   var x = new Number(23); y = new Boolean(true); z = new String("abc");
   document.write(typeof(x)+", "+typeof(y)+", "+typeof(z));
</script>
```

```
number, boolean, string
object, object, object
```

JavaScript Variables vs. Objects (2/2)

JavaScript Variables and Objects

- A string variable only has its value; a String object has properties and methods
- JavaScript automatically converts between variables and objects whenever necessary

```
<script>
   var str2 = new String("abc");
   var str3 = str2.valueOf();

   document.write(str3+", "+typeof(str3));
   document.write(str3.length+", "+str3.substr(1,1));
</script>
```

```
abc, string
3, b
```

Date Objects (1/4)

Date

- Used to work with dates and times
- new Date()
  - a Date object with current date and time

```
<script>
   var now = new Date(), then = new Date();
   document.write(now);
   then.setDate(now.getDate()+2);          //add 2 days
   then.setHours(now.getHours()+13);       //add 13 hours
   then.setMinutes(now.getMinutes()+24);  //add 24 minutes
   document.write(then);
</script>
```

Date Objects (2/4)

Methods

Date get methods

Method	Description
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)
getSeconds()	Get the seconds (0-59)
getTime()	Get the time (milliseconds since January 1, 1970)

Date Objects (3/4)

Methods

Date set methods

Method	Description
setDate()	Set the day as a number (1-31)
setFullYear()	Set the year (optionally month and day yyyy.mm.dd)
setHours()	Set the hour (0-23)
setMilliseconds()	Set the milliseconds (0-999)
setMinutes()	Set the minutes (0-59)
setMonth()	Set the month (0-11)
setSeconds()	Set the seconds (0-59)
setTime()	Set the time (milliseconds since January 1, 1970)

Date Objects (4/4)

Example

```
<script>
      var then = new Date();
      then.setFullYear(2022, 5, 19); // 5 -> June

      var now = new Date();
      document.write("Now is " + now + "<br/>");

      if (now < then) document.write("Now is <em>before</em> " + then);
      else document.write("Now is <em>after</em> " + then);
</script>
```

Now is Thu May 19 2022 10:37:54 GMT+0900 (한국 표준시)
Now is *before* Sun Jun 19 2022 10:37:54 GMT+0900 (한국 표준시)

Math Object (1/4)

Math Object

- Allows you to perform mathematical operations
- Provides mathematical constants and methods
  - e.g., Math.PI, Math.sqrt(16)
- All properties/methods of Math can be called without creating a Math object

Math Object (2/4)

Math Properties

Property	Description
Math.E	Returns Euler's number (approx. 2.718)
Math.LN2	Returns the natural logarithm of 2 (approx. 0.693)
Math.LN10	Returns the natural logarithm of 10 (approx. 2.302)
Math.LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
Math.LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
Math.PI	Returns PI (approx. 3.14)
Math.SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
Math.SQRT2	Returns the square root of 2 (approx. 1.414)

Math Object (3/4)

Math Methods

Method	Description
abs(x)	Returns the absolute value of x
ceil(x)	Returns x, rounded upwards to the nearest integer
cos(x)	Returns the cosine of x (x is in radians)
exp(x)	Returns the value of E^x
floor(x)	Returns x, rounded downwards to the nearest integer
log(x)	Returns the natural logarithm (base E) of x
max(x,y,z,...,n)	Returns the number with the highest value
min(x,y,z,...,n)	Returns the number with the lowest value
pow(x,y)	Returns the value of x to the power of y
random()	Returns a random number between 0 and 1
round(x)	Rounds x to the nearest integer
sin(x)	Returns the sine of x (x is in radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of an angle

Math Object (4/4)

Example

```
<script>
      document.write(Math.min(0, 150, 30, 20, -8, -200) + "<br/>");
      document.write(Math.max(0, 150, 30, 20, -8, -200) + "<br/>");
      document.write(Math.random() + "<br/>");
      document.write(Math.round(4.7) + "<br/>");
      document.write(Math.round(4.4) + "<br/>");
      document.write(Math.ceil(4.4) + "<br/>");
      document.write(Math.floor(4.7) + "<br/>");
</script>
```

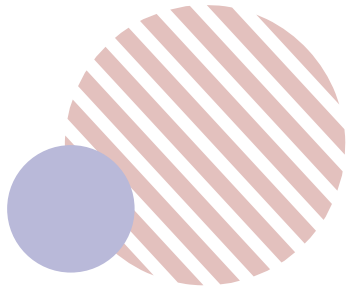
```
-200
150
0.17767978804564777
5
4
5
4
```

Global Properties and Functions

Global Properties and Functions

- Can be used in all built-in JavaScript Objects
- Global properties
  - Infinity
    - a numeric value that represents positive/negative infinity
  - NaN
    - “Not-a-Number” value
  - undefined
    - indicates that a variable has not been assigned a value
- Global functions
  - eval()
    - evaluates a string and executes it as if it was script code
    - var x = 10, a = eval(“x + 17”);
  - isFinite(), isNaN(), Number(), parseFloat(), parseInt(), etc.

Browser Object Model

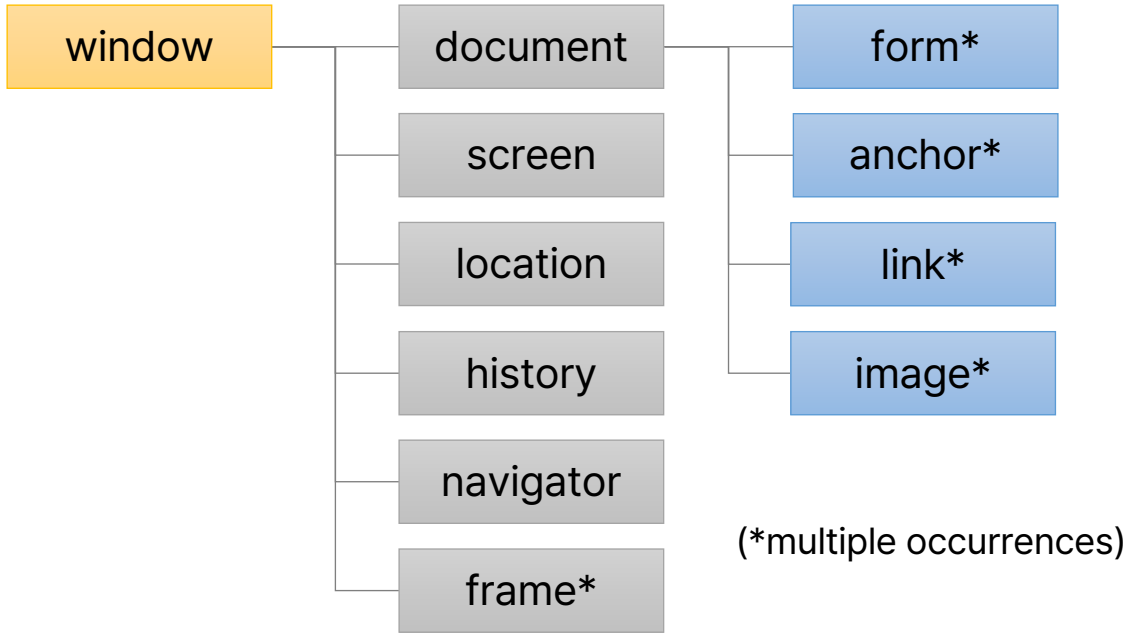


Browser Objects

Brower Object Model (BOM)

- Allows JavaScript to “talk to” the browser
- No official standards for BOM

Brower object hierarchy



Window Object (1/2)

Window Object

- Highest-level object in the JavaScript browser object hierarchy
- Default object; created automatically when a page is loaded
- May omit ‘window’ for explicit writing
  - document.write(“a string message”);
  - window.document.write(“a string message”);
  - See: http://www.w3schools.com/jsref/obj_window.asp
    - alert(), confirm(), prompt() are window object methods

Window Object (2/2)

Example

```
<script>
	var myWindow = null;

	function win_open() {
		if (myWindow == null) {
			myWindow = window.open("https://www.naver.com/");
			// open() returns a new Window object
		}
	}
</script>
<input type="button" value="Open new window" onclick="win_open()" />
```

Screen Object

Screen

- Contains information about the user's screen

Property	Description
availHeight	Returns the height of the screen (excluding the Windows Taskbar)
availWidth	Returns the width of the screen (excluding the Windows Taskbar)
colorDepth	Returns the bit depth of the color palette for displaying images
height	Returns the total height of the screen
pixelDepth	Returns the color resolution (in bits per pixel) of the screen
width	Returns the total width of the screen

Location Object

Location

- Used to get the current page address (URL) and to redirect the browser to a new page
  - See: http://www.w3schools.com/jsref/obj_location.asp

```
<script>
      document.write("Current url is: " + location.href + "<br/>");
      function jump() {
            location.assign("https://www.naver.com/");
            location.href = "https://www.naver.com/"; // does the same
      }
</script>
<input type="button" value="Jump to Naver" onclick="jump()" />
```

Current url is: file:/// .html

Jump to Naver

History Object

History

- Contains the browsing history
- To protect the privacy of the users, there are limitations to how JavaScript can access this object
  - See: https://www.w3schools.com/jsref/obj_history.asp

```
<script>
	function go_back() {
		window.history.back();
	}
	function go_forward() {
		window.history.forward();
	}
</script>
<input type="button" value="Back" onclick="go_back()" />
<input type="button" value="Forward" onclick="go_forward()" />
```

Navigator Object

Navigator

- Contains information about the visitor's browser
  - See: https://www.w3schools.com/jsref/obj_navigator.asp

```
<script>
      document.write("Browser Name is: " + navigator.appName + "<br/>");
      document.write("Code name is: " + navigator.appCodeName + "<br/>");
      document.write("Platform is: " + navigator.platform + "<br/>");
</script>
```

Browser Name is: Netscape
Code name is: Mozilla
Platform is: MacIntel

Exercises

Exercise 1

Exercise 1

- Write a JavaScript function that performs:
  - Given str1="Web ", str2="Programming ", str3="Rocks"
  - Concatenate the strings str1, str2, str3 to a variable str, and show str
  - Get the user input using prompt(), set the position of it within str into a variable index, and show index
  - Get the starting point, set the substring of length 8 of str into str_sub, and show str_sub
  - Convert str_sub to uppercase, and show it

Exercise 2

Exercise 2

- Write a JavaScript code to do the following:
  - Get current date and time, and print it using toString()
  - Print it after adding 5 days, 3 hours, and 25 minutes
  - Print it after adding 26 hours, and 78 minutes
  - Print its weekday (Monday, Tuesday, ...) using getDay()

End of Class