DECT-KS© TOOLS

Web Engineering Front-end Pt. 3

12. JavaScript: Revision

erb 僱員再培訓局 Employees Retraining Board
SCOPE School of Continuing and Professional Education 專業進修學院 香港城市大學 City University of Hong Kong
KRYSTAL INSTITUTE DIGITAL ECONOMY CORE TECHNOLOGY

DECT-KS©
TOOLS

12.1 Topics 1, 2, 3

Dynamic websites

- Dynamic content, can display customized content for different users
- Includes server-side code, database integration.

+ Interactive, easy to update, user friendly

- Requires more technical knowledge

Why learn JavaScript?

3 must learn web development languages:

1. HTML to define the content of web pages ✓
2. CSS to specify the layout of web pages ✓
3. **JavaScript** to program the behaviour of web pages

Where to put JavaScript?

1. In the <head> tag
2. In the <body> tag
3. In an external file (“filename.js”)

You can place however many scripts in an HTML file as you want but try to keep them all in one place.

Outputting JavaScript

4 main methods:

1. Writing into an existing HTML element (document.getElementByID(“id”).innerHTML)
2. Writing directly to the webpage (document.write)
3. Writing into an alert box (window.alert)
4. **Writing into the browser console (console.alert)**

Statements

A JavaScript statement can consist of...

- Values (Literals and variables)
- Operators
- Expressions
- Keywords
- Comments

Data types

Main JavaScript data types:

1. Strings
2. Numbers
3. Booleans
4. Undefined
5. Null
6. Objects
7. Arrays

typeof()

Use the typeof() method to find the data type of a variable.

e.g.

typeof(“Cow”);

// String

Declaring and assigning variables

```
var x = 20;

x; // 20

x = “Twenty”;

x; // Twenty
```

Swapping variables

```
var x = “A”;
var y = “B”;
var temp;

temp = x;
x = y;
y= temp;

x; // “B”
y; // “A”
```

Arithmetic operators

Operator	Function
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder
**	Exponentiation
++	Increment
--	Decrement

Operation order

Use parentheses to specify the order that operations should be executed in.

e.g.

```
x = 3 + 4 * 2; // 11
x = (3 + 4) * 2; // 14
```

Comments syntax

```
// Single line comment
/*
Multi
line
comments
*/
```

Have you been leaving comments in your code?

length

Use .length to find the number of characters in a string.

e.g.

“Hello”.length; // 5

search() and charAt()

Use search() to find the position of a specific character in a string. Use charAt() to find the character at the specified position of a string.

e.g.

“garden”.search(“r”); // 2

“key”.charAt(2); // y

slice()

Use slice() to extract and return a piece of a string based on the start and end positions.

e.g.

```
var string = “Feel good”;
var sliced = string.slice(2, 8);
sliced; // el goo
```

Escape characters examples

Code	Output
\\	\ (backslash)
\’	‘ (single quote)
\”	“ (double quote)
\&	& (ampersand)
\t	tab
\n	newline

Activity: Short quiz

```
var x = 20;
var y = 12;

x = y % 3;
x = (x + 5)**2
y = (y / 5).toPrecision(1);

// y++

console.log(x + “, “ + y); // What’s the output on the console?
```

Activity: Short quiz

```
var str = “never odd or even”;

str = str.slice(-13, -6);

var x = str.search(“d”);

var y = str.charAt(4);

console.log(str);
console.log(x + y); // What’s the output on the console?
```

DECT-KS©
TOOLS

12.2 Topics 4, 5, 6

More assignment operators

Operator	is the same as	If x = 10, y = 5
x += y	x = x + y	x = 15
x -= y	x = x - y	x = 5
x *= y	x = x * y	x = 50
x /= y	x = x / y	x = 2
x %= y	x = x % y	x = 0

Type conversion methods

The following methods can be used to convert variables to different data types.

- String()
- Number()
- Boolean()

JavaScript: Comparison and Logical Operators

Comparison operators

Operator	Description	If x = 10

==	Equal value	x == 10; // True
		x == 5 ; // False
		x == “10”; // True

===	Equal value and equal type	x === 10; // True
		x === 5; //False
		x === “10” // False

!=	Not equal value	x != 10; // False
		x != 5; // True
		x != “10”; // False

!==	Not equal value or equal type	x !== 10; // False
		x !== 5; // True
		x !== “10” // True

Dictionary order

1. Compare the first character of both words
2. If the first character of the first word is less than the first character of the second word, then the first word is lesser.
3. If the first characters are equal, then move on to the second characters.
4. Repeat until the end of either word.
5. If both strings are the same length, then they are equal. Otherwise, the shorter word is lesser.

JavaScript: Comparison and Logical Operators

Truth tables

AND

x	y	x && y
false	false	false
false	true	false
true	false	false
true	true	true

OR

x	y	x || y
false	false	false
false	true	true
true	false	true
true	true	true

NOT

x	!x
false	true
true	false

Combining logical operators

```
var x = 5; var y = 10; var z =20;

(x < y || y > z) && x < z; // True ((True OR False) AND True)
```

if statement syntax

```
if (condition1) {
  // code to be executed if condition 1 is true
} else if (condition2) {
  // code to be executed if condition 2 is true
} else {
  // code to be executed if both conditions are false
}
```

if statement example

```
if (score >= 80) {
  result = “merit”;
} else if (score >= 40) {
  result = “pass”;
} else if (score == “absent” && doctorsNote == true {
  result = “exempt”;
} else {
  result = “fail”;
}
```

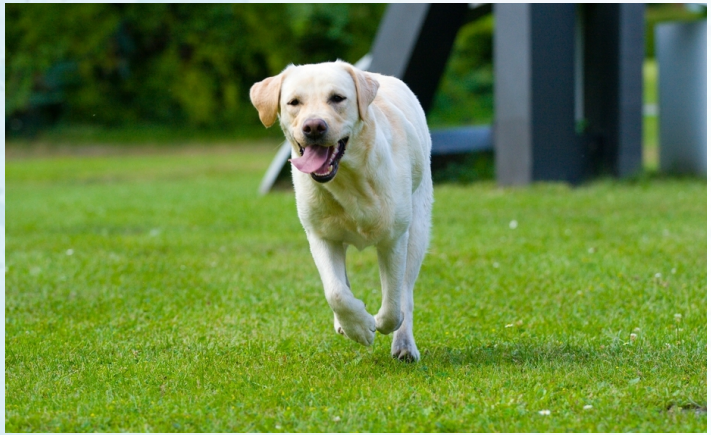
Objects

- Almost everything in JavaScript is an object, for example, strings, numbers and Booleans can all be objects.
- Objects are made out of properties and methods.

Object literal syntax

```
var name = {
  propertyName: propertyValue,
  methodName() {
    // Method code
  }
}
```

Dog object



```
var dog = {
   name: “Marley”,
   gender: “Male”,
   age: 2,
   bark() {
      return “Woof!”;
   }
}
```

Accessing object properties and methods

Object property:

object.property

or

object[“property”]

Object method:

object.method()

Arrays

Arrays are special objects that can be used to store multiple values into one variable.

e.g.

var food = [“Rice”, “Chicken”, “Cabbage”]

Accessing array elements

Array elements are accessed using their index number.

e.g.

food[1]; // Chicken

Array properties and methods

- .length: Find the number of elements in an array
- .push(): Add new element to the end of the array
- .pop(): Removes the last element in the array
- .shift(): Removes the first element of the array and shifts everything down
- .shift(): Insert new elements into an array, or replaces array elements

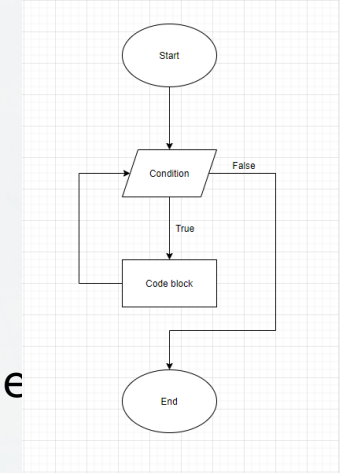
Activity: Short exercise

- Create a object representing yourself with first and last name properties, and a method to return your full name.
- Create an array of the names of you and 4 of your friends. Add 2 more friends after the 3rd element in the array.

12.3 Topics 7, 8

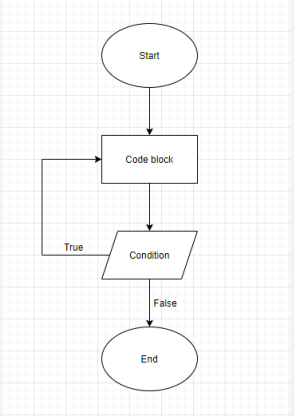
while loop syntax

```
while (condition) {
  // code to be run if condition is true
}
// program then returns to the beginning of the
```



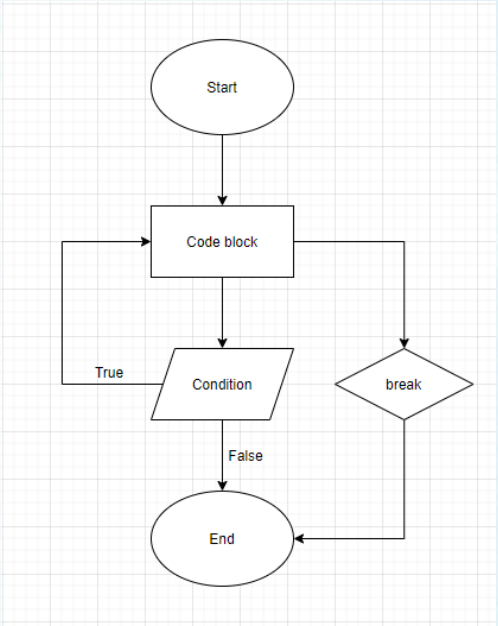
do-while loop syntax

```
do {
  // code to be run if it is the first iteration or if condit
} while (condition);
// program then returns to the beginning of the loop
```



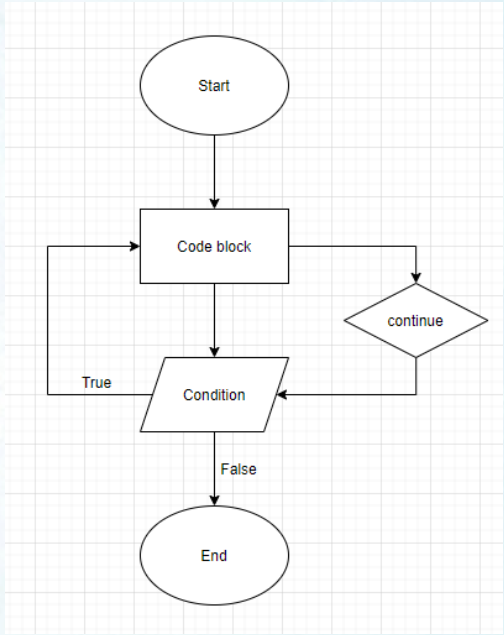
break statement example

```
var i = 0

while (i <= 10) {
  if (i == 4) {
    i++;
    break;
  }
  console.log(i);
  i++;
}

// 0, 1, 2, 3
```



continue statement example

```
var i = 0

while (i <= 20) {
  if (i % 2 == 0) {
    i++;
    continue;
  }
  console.log(i);
  i++;
}
```



for loop syntax

```
for (statement1; statement2; statement3) {
  // code to be executed
}
```

for loop syntax

- statement1: To initialize the variable to be incremented
- statement2: To set the loop condition
- stateement3: Final expression of the loop, usually used to increment

for-in loop syntax

```
for (property in object) {
  // code to be executed
}
```

Reminder: Don’t use for-in loops for arrays

for-of loop syntax

```
for (variable of iterable object) {
   // code to be executed
}
```

Reminder: Works for strings as well

Nested if example

```
if (userinput == username) {
  if (passinput == password) {
    console.log(“Log-in successful!”);
  } else {
    console.log(“Incorrect password”);
  }
} else {
  console.log(“Incorrect username”);
```

if inside while loop

```
var i = 0

while (i <= 10) {
  if (i == 4) {
    i++;
    break;
  }
  console.log(i);
  i++;
}

// 0, 1, 2, 3
```

Labelled break syntax

```
label:
for (...) {
  for (...) {
    if (...) {
      break label;
    }
  }
}
// break target
```

Activity: Short exercise

Write a program that checks if a string is a palindrome or not. A palindrome is a word sentence that is spelled the same forwards and backwards.

e.g. “racecar”, “taco cat”, “2002”

DECT-KS©
TOOLS

12.4 Topics 9, 10, 11

Math functions

Method	Description	Example
Math.abs(x)	Returns the absolute value of x	Math.abs(2-4); // 2
Math.sqrt(x)	Returns the square root of x	Math.sqrt(36); // 6
Math.cbrt(x)	Returns the cube root of x	Math.cbrt(27); // 3
Math.pow(x, y)	Returns x to the power of y	Math.pow(4, 5); // 1024
Math.max(x, y, ...)	Returns the largest of a series of numbers	Math.max(6, 12, 9); // 12
Math.min(x, y, ...)	Returns the smallest of a series of numbers	Math.min(6, 12, 9); // 6
Math.random()	Returns a random value between 0 and 1	Math.random(); // 0.9936125778727947

Math functions

Method	Description	Example

Math.round(x)	Rounds x to the nearest integer	Math.round(2.3); // 2
		Math.round(2.9); // 3
		Math.round(2.5); // 3

Math.ceil(x)	Rounds **up** x to the nearest integer	Math.ceil(2.3); // 3
		Math.ceil(2.9); // 3
		Math.ceil(2.5); // 3

Math.floor(x)	Rounds **down** x to the nearest integer	Math.floor(2.3); // 2
		Math.floor(2.9); // 2
		Math.floor(2.5); // 2

Math.trunc(x)	Returns only the integer of x	Math.trunc(2.3); // 2
		Math.trunc(100.45); // 100
		Math.trunc(-12.3); // -12

Set Date object methods

Method	Description	Example (Default: Wed May 19 2021 13:28:24)

setFullYear()	Set the year, or optionally the month and day	date = setFullYear(2018); // Wed May 19 2018 13:28:24
		date = setFullYear(2018, 5, 25); // Mon Jun 25 2018 13:28:24

setMonth()	Set the month (0-11)	date = setMonth(11); // Sun Dec 19 2021 13:28:24
setDate()	Set the date (1-31)	date = setDate(21); // Fri May 21 2020 13:28:24
setHours()	Set the hour (0-23)	date = setHours(11); // Wed May 19 2021 11:28:24
setMinutes()	Set the minutes (0-59)	date = setMinutes(41); // Wed May 19 2021 13:41:24
setSeconds()	Set the seconds (0-59)	date = setSeconds(35); // Wed May 19 2021 13:28:35
setMilliseconds()	Set the milliseconds (0-999)	date = setMilliseconds(450); // Wed May 19 2021 13:28:24
setTime()	Set the time (in milliseconds from Jan 1 1970)	date = setTime(999999999999); // Sun Sep 09 2001 09:46:39

Get Date object methods

Method	Description	Example (date = Wed May 19 2021 13:28:24)
getFullYear()	Get the year as a number	date = getFullYear(); // 2021
getMonth()	Get the month as a number	date = getMonth(); // 4
getDate()	Get the date as a number	date = getDate(); // 19
getHours()	Get the hour as a number	date = getHours(); // 13
getMinutes()	Get the minute as a number	date = getMinutes(); // 28

Get Date object methods

Method	Description	Example (date = Wed May 19 2021 13:28:24)
getSeconds()	Get the second as a number	date = getSeconds(); // 24
getMilliseconds()	Get the millisecond as a number	date = getMilliseconds); // 314
getTime()	Get the time as a number (in milliseconds from Jan 1 1970)	date = getTime(); // 1621402104246
getDay()	Get the day of the week as a number	getDay() = 3
Date.now()	Get the current time (in milliseconds from Jan 1 1970)	Date.now() // 1621402104976

sort() example

```
var students = [“Scott”, “Dean”, “Robert”, “Eric”];

students.sort();

console.log(students);

// [“Dean”, “Eric”, “Robert”, “Scott”]
```

reverse() example

```
var students = [“Scott”, “Dean”, “Robert”, “Eric”];
students.sort();
students.reverse();
console.log(students);
// ["Scott", "Robert", "Eric", "Dean"]
```

join() example

```
var days = [“Monday”, “Tuesday”, “Wednesday”, “Thursday”];
console.log(days.join(“ ==> “));
// “Monday ==> Tuesday ==> Wednesday ==> Thursday”
```

Function example

```
function greet(name) {
   return (“Hello, ”+name+”!”);
}
```

Function example

greet: function name

name: function parameter

return …: function output

greet(): To invoke the function

Method example

```
var person = {
  firstName: “Joe”,
  lastName: “Schmoe”,
  introduce() {
    return (“I’m “+this.firstName+” “+this.lastName+”.”);
  }
}
```

Recursion example

```
function countdown(x) {
  if (x > 0) {
    console.log(x);
    countdown(x-1);
}

countdown(5);
// 5, 4, 3, 2, 1
```

Syntax error examples

```
console.log(“Hello); // Missing closing quotation mark

math.pow(2, 3); // “math” should have a capitalized M

var dog = {
   name: "Marley";
   age: 2;
   breed: "Labrador retriever";
}
// Object properties are separated with commas (,)
```

Runtime error example

```
var dog = {
   name: "Marley“,
   age: 2,
   breed: "Labrador retriever“,
}

console.log(cat.name);
```

Logic error example

```
// Goal is to create a rectangle shape made of asterisks
var output = “”;
for (var x = 1; x <= 5; x++) {
  for (var y = 1; y <= 5; y++) {
    output += “*” + “ “;
  }
  console.log(output);
}
```

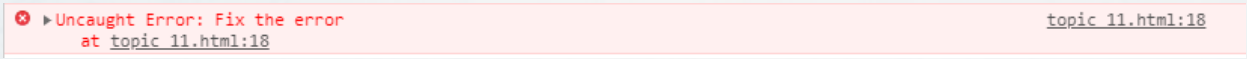
```
Expectation:

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Reality:

* * * * *
* * * * * * * * * *
* * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * *
>
```

try-catch-finally statement syntax

```
try {
  // Code to be tested
} catch (error) {
  // Code to handle errors
} finally {
  // Code to be executed after error handling
}
```

Error object example

throw new Error(“Fix the error”);



12.5 Q and A

Activity: Q and A

Ask any questions you have regarding any of the topics covered in this course.

DECT-KS©
TOOLS

The End

Reference 1: Labrador (Slide 31) https://www.flickr.com/photos/23807781@N06/3798577491