IF231 Pengantar Teknologi Internet

05 - JavaScript

Alexander Waworuntu, S.Kom., M.T.I.
Fenina Adline Twince Tobing, S.Kom., M.Kom.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
SEMESTER GENAP TAHUN AJARAN 2021/2022

REVIEW

- Bootstrap Introduction
- Bootstrap Layout & Grid System
- Bootstrap Customization
- Bootstrap Components (eg. Buttons, Card)
- Bootstrap Forms

TODAY's OUTLINE

- JavaScript Introduction
- Variables, Functions, Operators
- JavaScript Built-in Functions
- Objects

A Brief History of JavaScript

- **JavaScript** IS NOT Java
- Developed under the name **Mocha**
- Officially called **LiveScript** when it first shipped in beta releases of Netscape Navigator 2.0 - September 1995
- Renamed to **JavaScript** when it was deployed in the Netscape Navigator 2.0 beta 3 - December 1995
- JavaScript was standardized in 1996 by the European Computer Manufacturers Association (**ECMA**), which is why you sometimes hear it called **ECMA Script**.

Further Readings:

https://medium.com/@_benaston/lesson-1a-the-history-of-javascript-8c1ce3bffb17
https://en.wikipedia.org/wiki/JavaScript
https://developer.mozilla.org/en-US/docs/Web/JavaScript?retiredLocale=id
www.ecma-international.org/memento/index.html

What is JavaScript

- **JavaScript is a lightweight but incredibly powerful scripting language. Mostly used for web, but not only that.**

  See next slide about what JavaScript can be used for besides web programming.
- **A dynamic programming language**

  JavaScript doesn't need to be run through any form of compiler that interprets our human-readable code into something that the browser understand. The browser effectively reads the code the same way we do and interprets it on the fly.
- **Loosely typed language**

  We don't necessarily have to tell JavaScript what a variable is. If we're setting a variable to a value of 5, we don't have to programmatically specify that variable as a number; 5 is a number, and JavaScript recognizes it as such.

What else can JS be used for?

- **Desktop apps** (Electron, NW.js, AppJS, Meteor, Proton Native)
- **Mobile Apps** (React Native)
- **IoT**. You can have node.js runtime on Raspberry Pi, for example.
  Raspberry Pi + NodeJS, Beginners
  Beginner's Guide to Installing Node.js on a Raspberry Pi
- **Robotics** (Johnny-Five: The JavaScript Robotics & IoT Platform, NodeBots, CyclonJS)
- **Game** (Unity - Game Engine)
- **Command line tools** (Writing Command-Line Tools with Node, Building Command-Line Tools with Node.js)
- **Operating Systems** (Node-OS, JSOS)
- etc.?

Adding JavaScript to a Page

- **Inline Script**

```
<button onclick="alert('You clicked me!');">Click Me!
</button>
<button onmouseover="alert('You hovered over me!')">
Hover over me!</button>
```

- **Embedded Script**

```
<body>
    <script>alert("Hello World!");</script>
</body>
```

- **External Script**

```
<body>
    <script src="my_script.js"></script>
</body>
```

JavaScript Variables

A variable is like an information container. You give it a name and then assign it a value, which can be a number, text string, an element in the DOM, or a function- anything. This gives us a convenient way to reference that value later by name. The value itself can be modified and reassigned in whatever way our scripts' logic dictates.

```
var foo; //Undifined
var kosong = null; //Null
var nilai = 50; //Numbers
var nama = “John Thor”; //Strings
var love = true; //Booleans
var myArray = [5, “five”, “5”]; //Array
```

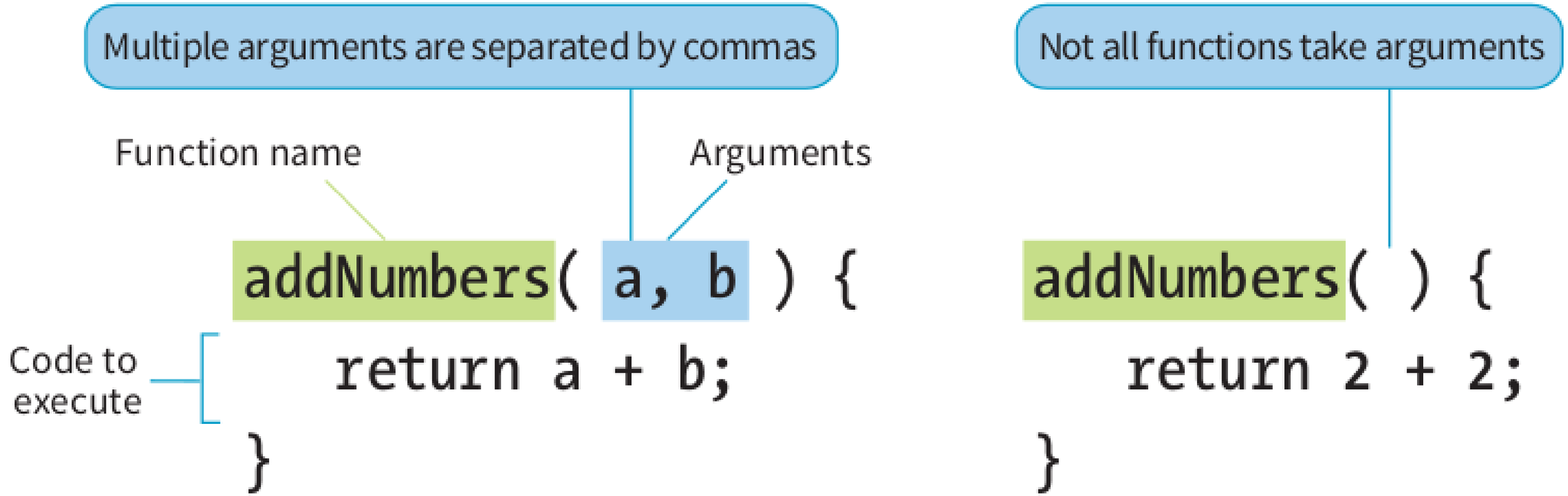
Ref: https://www.w3schools.com/js/js_variables.asp

JavaScript Operators

- Arithmetic Operators:
  `+  -  *  /  %  ++  --`
- Assignment Operators:
  `=  +=  -=  *=  /=  %=`
- String Operators:
  `=  +=`
- Comparison Operators:
  `==  ===  !=  !==  >  <  >=  <=`
- Logical Operators:
  `&&  ||  !`

Ref:
https://www.w3schools.com/js/js_operators.asp
https://www.w3schools.com/js/js_arithmetic.asp
https://www.w3schools.com/js/js_assignment.asp
https://www.w3schools.com/js/js_comparisons.asp

Functions


Multiple arguments are separated by commas
Function name
Arguments
addNumbers(a, b) {
Code to execute
return a + b;
}
Not all functions take arguments
addNumbers( ) {
return 2 + 2;
}

Variable Scope

```
function double(num){
    var total = num + num; // Local Scope
    return total;
}
```

```
function double(num){
    total = num + num; // Global Scope
    return total;
}
```

```
function double(num){
    total = num + num;
    return total;
}
var total = 10;
var number = double(20
alert(total); // Alert
```

SCOPE CHEAT SHEET

Variable	Location	Scope
var identifier value	Outside a function	Global
var identifier value	Inside a function	Local
identifier value	Inside a function	Global

Native Functions

Hundreds of predefined functions are built into JavaScript, including these:

- **`alert(), confirm(), prompt()`**
  These functions trigger browser-level dialog boxes.
- **`Date()`**
  Returns the current date and time.
- **`parseInt("123")`**
  This function will, among other things, take a string data type containing numbers and turn it into a number data type. The stringis passed to the function as a argument.
- **`setTimeout(functionName, 5000)`**
  Executes a function after a delay. The function is specified in the first argument, and the delay is specified in a milliseconds in the second argument (5000 milliseconds = 5 seconds)

String Built-in Functions

Method	Description
`charAt()`	Returns the character at the specified index (position)
`endsWith()`	Checks whether a string ends with specified string/characters
`includes()`	Checks whether a string contains the specified string/characters
`indexOf()`	Returns the position of the first found occurrence of a specified value in a string
`lastIndexOf()`	Returns the position of the last found occurrence of a specified value in a string
`match()`	Searches a string for a match against a regular expression, and returns the matches
`replace()`	Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced
`search()`	Searches a string for a specified value, or regular expression, and returns the position of the match

String Built-in Functions

Method	Description
`split()`	Splits a string into an array of substrings
`startsWith()`	Checks whether a string begins with specified characters
`substr()`	Extracts the characters from a string, beginning at a specified start position, and through the specified number of character
`substring()`	Extracts the characters from a string, between two specified indices
`toLowerCase()`	Converts a string to lowercase letters
`toString()`	Returns the value of a String object
`toUpperCase()`	Converts a string to uppercase letters
`trim()`	Removes whitespace from both ends of a string
`valueOf()`	Returns the primitive value of a String object

Math Built-in Functions

Method	Description
`abs(x)`	Returns the absolute value of x
`acos(x)`	Returns the arccosine of x, in radians
`asin(x)`	Returns the arcsine of x, in radians
`atan(x)`	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
`atan2(x,y)`	Returns the arctangent of the quotient of its arguments
`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

Date Built-in Functions

Method	Description
`getDate()`	Returns the day of the month (from 1-31)
`detDay()`	Returns the day of the week (from 0-6)
`getFullYear()`	Returns the year
`getHours()`	Returns the hour (from 0-23)
`getMilliseconds()`	Returns the milliseconds (from 0-999)
`getMinutes()`	Returns the minutes (from 0-59)
`getMonth()`	Returns the month (from 0-11)
`getSeconds()`	Returns the seconds (from 0-59)
`getTime()`	Returns the number of milliseconds since midnight Jan 1 1970, and a specified date

JavaScript Objects

```
function Manusia(nm, sx, age){
    this.nama = nm;
    this.kelamin = sx;
    this.usia = age;
    this.gantiKelamin = operasiKelamin;
    this.getSisaUsiaProduktif = hitungKapanPensiun;

    function operasiKelamin(newSx){
        this.kelamin = newSx;
    }
    function hitungKapanPensiun(){
        var jmlTahun = 65 - this.usia;
        return jmlTahun;
    }
}

var wawo = new Manusia("Alex Wawo", "Male", 25);
alert(wawo.kelamin);
wawo.gantiKelamin("Pria");
alert(wawo.kelamin);
```

Manusia
Properties: nama kelamin usia
Methods: gantiKelamin() getSisaUsiaProduktif()

JavaScript Objects

```
var wawo = {
    nama: "Alex Wawo",
    kelamin: "Male",
    usia: 25,
    gantiKelamin: function(newSx){
        this.kelamin = newSx;
    },
    getSisaUsiaProduktif: function(){
        var jmlTahun = 65 - this.usia;
        return jmlTahun;
    }
}

alert(wawo.kelamin);
wawo.gantiKelamin("Pria");
alert(wawo.kelamin);
```

Object: wawo

Properties:
nama
kelamin
usia

Methods:
gantiKelamin()
getSisaUsiaProduktif()

SUMMARY

After finishing this module, you should be able:

- to understand what is JavaScript
- to understand how to implement built in functions in JavaScript
- to understand how to implement variables & functions
- to understand how to create objects in JavaScript

NEXT WEEK'S OUTLINE

- Document Object Model (DOM)

REFERENCES

- https://www.w3schools.com/js/default.asp
- Darren James (2017). JavaScript Novice to Ninja. SitePoint.

Visi

Menjadi Program Studi Strata Satu Informatika **unggulan** yang menghasilkan lulusan **berwawasan internasional** yang **kompeten** di bidang Ilmu Komputer (*Computer Science*), **berjiwa wirausaha** dan **berbudi pekerti luhur**.

INFORMATIKA
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Misi

1. Menyelenggarakan pembelajaran dengan teknologi dan kurikulum terbaik serta didukung tenaga pengajar profesional.
2. Melaksanakan kegiatan penelitian di bidang Informatika untuk memajukan ilmu dan teknologi Informatika.
3. Melaksanakan kegiatan pengabdian kepada masyarakat berbasis ilmu dan teknologi Informatika dalam rangka mengamalkan ilmu dan teknologi Informatika.