JAVASCRIPT

- WHAT IS JAVASCRIPT
- COMMON LIBRARIES OF JAVASCRIPT

WHAT IS JAVASCRIPT?

- Javascript initially is scripting language of HTML and the Web, now it certainly matured into a robust language that used to power websites, but also back ends these days
- Object-oriented
    - Prototypical inheritance instead of class-based

USING JAVASCRIPT

```
<html>
<body>

<p id="demo"></p>

<script type="text/javascript"> // Inline javascript
    var message = "Hello javascript";
    console.log(message);

    var demoDiv = document.getElementById("demo");
    demoDiv.innerHTML = "<span>Hello world</span>"
</script>
<script type="text/javascript" src="js/index.js">// Include external JS file
</script>
</body>
</html>
```

VARIABLES

JavaScript variables are containers for storing data values.

```
<html>
<body>

<p id="demo"></p>

<script type="text/javascript">
    var message = "Hello javascript"; // string variable
    var myAge = 18; //number variable
    var demoDiv = document.getElementById("demo"); // object variable
    var isActive = true // boolean variable
</script>
</body>
</html>
```

CONDITIONALS

Very often when you write code, you want to perform different actions for different decisions.

```
<html>
<body>

<p id="demo"></p>

<script type="text/javascript">
    var message = "Hello javascript"; // string variable
    var myAge = 18; //number variable

    if(myAge < 18){
        alert("You need to over 18 to apply for home loan");
    }
    else if(myAge > 18 && myAge < 21){
        alert("Please provide proof of your age");
    }
    else{
        alert("Your age is validated, please go to next step");
    }

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

FUNCTIONS

A block of code designed to perform a particular task.

```
<script type="text/javascript">
    var message = "Hello javascript"; // string variable
    var myAge = 18; //number variable

    function checkAge(age){
        if(age < 18){
            return "You need to over 18 to apply for home loan";
        }
        else if(age > 18 && age < 21){
            return "Please provide proof of your age";
        }
        else{
            return "Your age is validated, please go to next step";
        }
    }
    alert(checkAge(myAge));

</script>
```

SCOPE

Scope is the set of variables, objects, and functions you have access to.

```
<script type="text/javascript">
    var message = "My name is "; // global variable

    function showMessage(){
        var myName = "Justin Pham"; //local variable
        alert(message + myName);
    }

    function testMe(){
        alert(myName); // this function can't access myName variable
    }

</script>
```

CLOSURE

Allow access to outer variable within inner scope regardless of variable lifetime

```
function showMeYourName(firstName, lastName){
    var welcomeText = "Your name is ";
    // this inner function has access to the outer function's variables, including the parameter
    function alertFunction(){
        return welcomeText + firstName + " " + lastName;
    }

    return alertFunction();
}
alert(showMeYourName("Justin","Pham"));
```

EXAMPLE OF JQUERY CLOSURE

```
$(function() {

var listOfUserIds = [];
    $(".getUserId").click(function() { // this closure has access to the listOfUserIds variable
        listOfUserIds.push (this.prop("name")); // update the listOfUserIds variable in the outer function's scope
    });
});
```

JS OBJECT

JS Object is just a set of name value pair

```
//Object

var user = {
    username: "software.participant",
    name: "Industry Connect",
    skills: ["C#","Javascript",".NET","NodeJS","ReactJS"],
    showActivityHistory: function() {
        alert("Show activity");
    },
    tutor: {
        tutorName: "Justin Pham",
        company: "MVP Studio",
        address: "34 Shaddock St"
    }
};

user.showActivityHistory();
console.log(user.tutor.address);
```