Assignment 1 – JavaScript ES6

# Arrow functions

## Refer the Arrow Functions Examples slide and come up with your own arrow functions.

### Arrow function with curly brackets.

const func1 = (param)=>{ // param is not mandatory

Statement

return something

}

### Arrow function that doesn’t have curly brackets that directly returns a value.

const func2 = (param) => statement // return not required// param not mandatory.

## Prove that the same arrow functions are never equal to each other.

This is because the arrow function does not have its own binding to the keyword “this”.

## Create an object with a constructor using the arrow function and regular function. Refer: https://www.w3schools.com/js/js\_object\_constructors.asp

function Student(fname, lname, age, course, grade) {

this.firstName = fname;

this.lastName = lname;

this.age = age;

this.course = course;

this.grade = grade;

this.fullName = function {

return this.firstName + “ ” + this.lastName;

}

}

Arrow functions are not preferred to use as methods in a class function because they do not have their own binding with the keyword “this”.

# Import and Export

## Create a folder name Lesson1. Create 3 JS files (student, app, helper) under the folder. Code in the Exports and Imports slide. Open up the terminal from VSCode and type node app.js that will execute the code.

//package.json

{

"name": "lesson-1",

"version": "1.0.0",

"type": "module"

}

//student.js

class Student{

constructor(fname, lname, age){

this.fname = fname;

this.lname = lname;

this.age = age;

}

sayHi = ()=> `Hi. I am ${this.fname} ${this.lname}`

}

export default Student;

// helper.js

export const minutesInHour = 60;

export const sayHello = ()=> "Hello";

// app.js

import Student from "./student.js";

import anyName from "./student.js";

import { minutesInHour } from "./helper.js";

import { sayHello } from "./helper.js";

// Testing in console

const stu1 = new Student("Yogesh","Ghimiray",10)

console.log (stu1.sayHi())

console.log(minutesInHour);

console.log(sayHello());

// Output in console

Hi. I am Yogesh Ghimiray

60

Hello

# Class

## Create a class and subclass

### Create a Person class

class Person {

constructor(gender){

this.gender = gender;

}

sayGender = ()=> `I am ${this.gender}`;

}

export default Person;

### Update the student.js

import Person from "./person.js";

class Student extends Person{

constructor(gender,fname, lname, age){

super(gender)

this.fname = fname;

this.lname = lname;

this.age = age;

}

sayHi = ()=> `Hi, I am ${this.fname} ${this.lname}. I am ${this.gender}`

}

export default Student;

### Update app.js

import Student from "./student.js";

// import anyName from "./student.js";

import { minutesInHour } from "./helper.js";

import { sayHello } from "./helper.js";

// Testing in console

const stu1 = new Student("Male","Yogesh","Ghimiray",10)

console.log (stu1.sayHi())

console.log(minutesInHour);

console.log(sayHello());

// Output in console

Hi, I am Yogesh Ghimiray. I am Male

60

Hello

# Other tasks – Destructuring and the spread operator

## Add more fields to the student class. Update the sayHi method accordingly. And do desturcturing in the app.js.

const stu1 = new Student("Male","Yogesh","Ghimiray",10)

let {gender,fname,lname,age} = stu1;

//Testing in Console

console.log(gender,fname,lname,age);

// Output in console

Male Yogesh Ghimiray 10

## Copy the student using spread operator. After that, update the copied students object name, age, and hobby. Submit the final code.

let stu2 = {...stu1}

stu2.fname = "Jack";

stu2.lname = "Smith";

stu2.age = 20;

stu2.hobby = "Swimming"

//Testing in Console

console.log(stu1);

console.log(stu2);

// Output in console

{

sayGender: [Function: sayGender],

gender: 'Male',

sayHi: [Function: sayHi],

fname: 'Yogesh',

lname: 'Ghimiray',

age: 10

}

{

sayGender: [Function: sayGender],

gender: 'Male',

sayHi: [Function: sayHi],

fname: 'Jack',

lname: 'Smith',

age: 20,

hobby: 'Swimming'

}

# Array methods

## Use and play with map, find, findIndex, filter, array functions on a sample array. You can use developer.mozilla.org.

const array1 = [1,2,3,4,5,6,7,8,9]

const array2 = array1.map(element=>{

return element\*2;

});

const num1 = array1.find(element=>{

return element > 4;

});

const index = array1.findIndex(element=>{

return element >4;

});

const filteredArr = array1.filter(element=>{

return element > 4;

});

# Research

## Read the lecture thoroughly. In your opinion, what are the 3 main points? Please share.

1. Always use arrow functions in React.
2. We can use export default only one time in one JS file.
3. We can use import in JS files instead of require.

## What is the difference between an arrow function and regular function?

1. Arrow function is not binding the keyword “this” where regular function does.
2. Hoisting is possible in regular function but not possible in arrow function.
3. Arrow function is not suitable to be used as a method in an object.

## Do a research and list at least 3 pros and cons of arrow functions.

Pros :

1. Every time you pass an arrow function as a callback, it will create a new function object
2. arrow functions are better suited for callbacks
3. Arrow functions are best for callbacks or methods like map, reduce, or forEach.

Cons:

1. Arrow function is not suitable to be used as a method in an object.
2. Arrow function is not binding the keyword “this”.
3. Hoisting is not possible in arrow function.

What is the difference between the spread operator and rest operator?