JAVA SCRIPT – DAY 2

MODULES AND CLASSES

//ES6 introduces file based module, in which each module is represented by a separate .js file.

//Now, you can use the export or import statement in a module to

//export or import variables, functions, classes or any other entity to/from other modules or files.

//Let's create a module i.e. a JavaScript file "main.js" and place the following code in it:

let greet = "Hello World!";

const PI = 3.14;

function multiplyNumbers(a, b) {

return a \* b;

}

// Exporting variables and functions

export { greet, PI, multiplyNumbers };

//Now create another JavaScript file "app.js" with the following code:

import { greet, PI, multiplyNumbers } from './main.js';

alert(greet); // Hello World!

alert(PI); // 3.14

alert(multiplyNumbers(6, 15)); // 90

Finally create a HTML file "test.html" and with the following code and open this HTML file in your browser using HTTP protocol (or use localhost). Also notice the type="module" on script tag.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>ES6 Module Demo</title>

</head>

<body>

<script type="module" src="app.js"></script>

</body>

</html>

To work with modules we use plunkers. plnkr.co and Lauch the editor

traceur helps to compile the ES6 code to ES5 and add systemjs

https://next.plnkr.co/edit/?open=lib%2Fscript.js&preview

package.json

{

"plnkr": {

"runtime": "system"

},

"dependencies": {

"traceur": "^0.0.111",

"systemjs": "^5.0.0"

}

}

**lib/script.js**

// Add your code here

import { keyvalue } from './external.js';

alert(keyvalue);

export let keyvalue=1050;

**lib/external.js**

export let keyvalue=1050;

**index.html**

<!doctype html>

<html>

<head>

<link rel="stylesheet" href="lib/style.css">

<script src="lib/script.js"></script>

</head>

<body>

<h1>ES6 Modules</h1>

</body>

</html>

# Testing a Function

**/lib/external.js**

export let keyvalue=1050;

export function test(){

alert('Tested .....');

}

**lib/script.js**

// Add your code here

import { keyvalue, test } from './external.js';

alert(keyvalue);

test();

**index.html**

<!doctype html>

<html>

<head>

<link rel="stylesheet" href="lib/style.css">

<script src="lib/script.js"></script>

</head>

<body>

<h1>ES6 Modules</h1>

</body>

</html>

Now let us modify the code of

**/lib/external.js**

let keyvalue=1050;

function test(){

alert('Tested .....');

}

export{keyvalue,test};

## IMPORT AND EXPORT SYNTAX

----------------------------------------------------------------------------------------------------

Let us modify the code script.js

**lib/script.js**

// Add your code here

import { keyvalue as key , test } from './external.js';

alert(key);

test();

## CLASSES

Learn how to work with classes that make the creation of Object.use jsbin -https://jsbin.com/?js,console

# Demo1: Class with object

class Person{

}

let person=new Person();

console.log(person); //returns the EMPTY object.

# Demo2: class with method and object

class Person{

greet(){

console.log("Hello Function");

}

}

let person=new Person();

person.greet();

# Demo3: Class with constructor,object and method

class Person{

constructor(){

this.name='Ram';

}

greet(){

console.log("My name is "+this.name);

}

}

let person=new Person();

person.greet();

# Demo4: constructor with arguments

class Person{

constructor(name){

this.name=name;

}

greet(){

console.log("My name is "+this.name);

}

}

let person=new Person("Arun");

person.greet();

### **PROTOTYPE**

All JavaScript objects inherit properties and methods from a prototype.

Person objects inherit from Person.prototype

**Demo1:**

class Person{

constructor(name){

this.name=name;

}

greet(){

console.log("My name is "+this.name);

}

}

let person=new Person("Arun");

person.greet();

# What is the prototype of person object???

console.log(person.\_\_proto\_\_);

# Demo2

class Person{

constructor(name){

this.name=name;

}

greet(){

console.log("My name is "+this.name);

}

}

let person=new Person("Arun");

person.greet();

**What is the prototype of person object???**

console.log(person.\_\_proto\_\_==Object.prototype);

**Demo3:**

class Person{

constructor(name){

this.name=name;

}

greet(){

console.log("My name is "+this.name);

}

}

let person=new Person("Arun");

person.greet();

# What is the prototype of person object???

console.log(person.\_\_proto\_\_===Person.prototype);

#### INHERITANCE

# Demo1

class Person{

constructor(name){

this.name=name;

}

greet(){

console.log("My name is "+this.name);

console.log("My age is "+this.age);

}

}

class student extends Person{

constructor(age){

super('Ram');

this.age=age;

}

}

let obj=new student(27);

obj.greet();

##### STATIC

**Demo1:**

class Helper{

static logTwice(message){

console.log(message);

console.log(message);

}

}

Helper.logTwice("logged");

class Person{

constructor(name){

this.name=name;

}

greet(){

console.log("My name is "+this.name);

console.log("My age is "+this.age);

}

}

class student extends Person{

constructor(age){

super('Ram');

this.age=age;

}

}

let obj=new student(27);

obj.greet();

#### GETTERS AND SETTERS

class Person{

constructor(name){

this.\_name=name; //used for private variables

}

get name(){

return this.\_name.toUpperCase();

}

set name(value){

if (value.length>2){

this.\_name=value;

}

console.log('Rejected!');

}

}

let person=new Person('Ram');

console.log(person.\_name);

person.name='Anil';

console.log(person.\_name);