

## Practical – 7

### Aim 1: Demonstrate ES6 New Syntax and Test Browser Compatibility

#### ARROW FUNCTIONS

Source Code:

OLD SYNTAX :

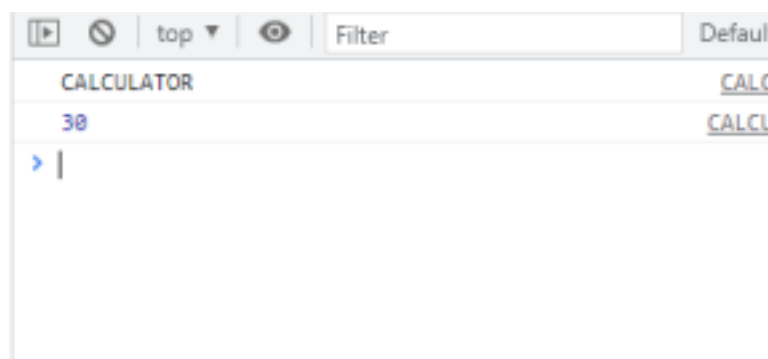
```
<!DOCTYPE HTML>
<head> <title> </title> </head>
<body><script>
console.log("CALCULATOR");
let add = function (x, y) {
return x + y; };
console.log(add(10, 20));
</script> </body>
</html>
```

NEW SYNTAX :

(USING ARROW FUNCTION)

```
<!DOCTYPE HTML>
<head> <title> </title> </head>
<body>
<script>
console.log("CALCULATOR");
let add = (x, y) => x + y;
console.log(add(10, 20)); // 30;</script>
</body>
</html>
```

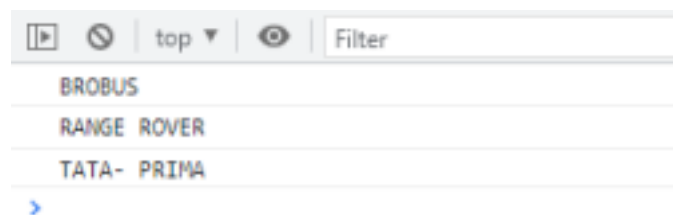
**Output:**



**Aim 2.1: DESTRUCTURING ARRAYS SOURCE CODE :****Source Code:**

OLD WAY :

```
<!DOCTYPE HTML>
<head>
<title>
</title>
</head>
<html>
<script>
const vehicles = ['BROBUS', 'TATA- PRIMA', 'RANGE ROVER'];
// old way
const car = vehicles[0];
const truck = vehicles[1];
const suv = vehicles[2];
console.log(car);
console.log(suv);
console.log(truck);
</script>
</body>
</html>
```

**Output:**

NEW WAY :

```
<!DOCTYPE HTML>

<head>

<title>

</title>

</head>

<html>

<script>

const vehicles = ['BROBUS', 'TATA- PRIMA', 'RANGE ROVER'];

const [car,truck, suv] = vehicles;

console.log(car);

console.log(truck);

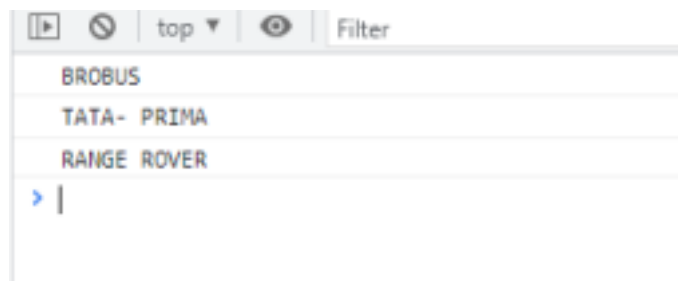
console.log(suv);

</script>

</body>

</html>
```

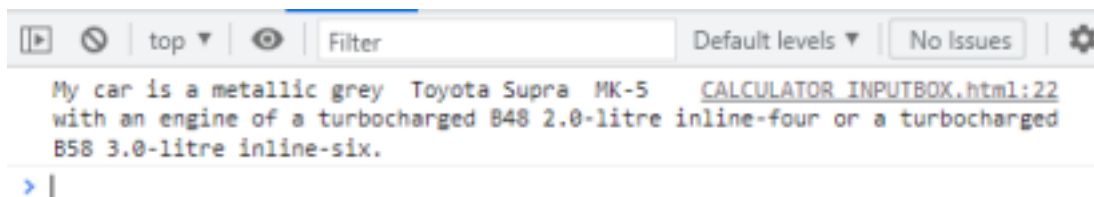
**Output:**



**Aim-2.2: DESTRUCTURING OBJECTS****Source Code:**

OLD WAY :

```
<!DOCTYPE HTML>
<head>    <title></title>    </head>
<body>
<script>
const vehicleOne = {
brand: 'Toyota',
model: 'Supra MK-5 ',
type: 'car',
year: 2021,
color: 'metallic grey ',
engine: 'a turbocharged B48 2.0-litre inline-four or a turbocharged B58 3.0-
litre inline-six'}
myVehicle(vehicleOne);
function myVehicle(vehicle) {
const message = 'My ' + vehicle.type + ' is a ' + vehicle.color + ' ' +
vehicle.brand + ' ' + vehicle.model + 'with an engine of ' + vehicle.engine + '.';
console.log(message);}
</script>
</body>
</html>
```

**Output:**

NEW WAY :

```
<!DOCTYPE HTML>
<head>
<title>
</title>
</head>
<html>
<script>
const vehicleOne = {
brand: 'Koenigsegg',
model: 'agera rs',
type: 'car',
color: 'white '
}
myVehicle(vehicleOne);
function myVehicle({type, color, brand, model}) {
const message = 'My ' + type + ' is a ' + color + 'coloured ' + brand + ' ' +
model + '.';
console.log(message);
}
</script>
</body>
</html>
```

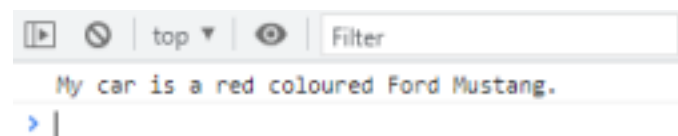
**Output:**



**Aim-3: SPREAD OPERATOR****Source Code:**

OLD WAY :

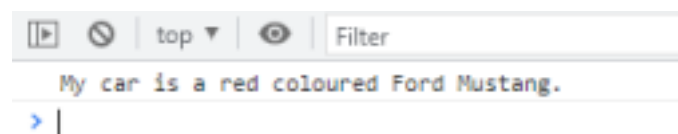
```
<!DOCTYPE HTML>
<head>
<title>
</title>
</head>
<html>
<script>
const myVehicle = {
brand: 'Ford',
model: 'Mustang',
color: 'red'
}
const updateMyVehicle = {
type: 'car',
year: 2021
}
const message = 'My ' + updateMyVehicle.type + ' is a ' + myVehicle.color + '
coloured ' + myVehicle.brand + ' ' + myVehicle.model + '!';
console.log(message);
</script>
</body>
</html>
```

**Output:**

NEW WAY :

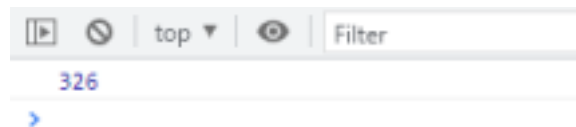
```
<!DOCTYPE HTML>
<head>
<title>
</title>
</head>
<html>
<script>
const myVehicle = {
brand: 'Ford',
model: 'Mustang',
color: 'red'
}
const updateMyVehicle = {
type: 'car',
year: 2021
}
const myUpdatedVehicle = {...myVehicle, ...updateMyVehicle}
vehicle(myUpdatedVehicle);
function vehicle({type, color, brand, model}) {
const message = 'My ' + type + ' is a ' + color + ' coloured ' + brand + ' ' +
model + '.';
console.log(message);
}
</script>
</body>
</html>
```

**Output:**



**REST OPERATOR :**

```
<!DOCTYPE html>
<html>
<body>
<script>
function sum(...args) {
let sum = 0;
for (let arg of args) sum += arg;
return sum;
}
let x = sum(4, 9, 16, 25, 29, 100, 66, 77);
console.log(x);
</script>
</body>
</html>
```

**Output:****Conclusion:**

From this practical I came to know the new features of the ES6 and their advantages over the traditional methods followed in java script