

Practical-7

Aim: Demonstrate ES6 New Syntax and Test Browser Compatibility.

➤ Arrow Function: -

JS CODE:

Program:

```
WT > hello > JS js > ...
1  const regularFunction = function (param1, param2) {
2      return param1 + param2;
3  };
4  const arrowFunction = (param1, param2) => param1 + param2;
5  const arrowFunctionMultiParams = (param1, param2, param3) => { return param1 + param2 + param3;
6  };
7  const noParameters = () => "Hello world!";
8  console.log(regularFunction(1,2));
9  console.log(arrowFunction(2,3));
10 console.log(arrowFunctionMultiParams(1,2,3));
11 console.log(noParameters());
```

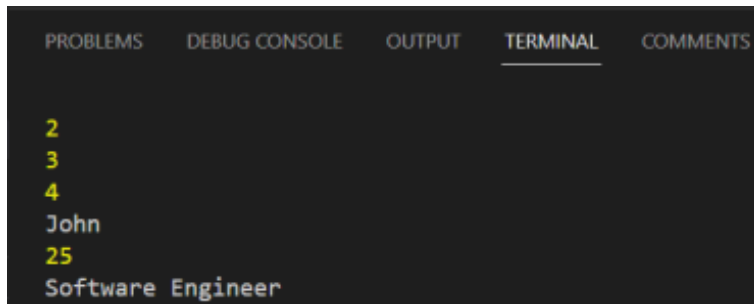
Output:

```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  COMMENTS
PS F:\SEM 4\IT256 DATA STRUCTURES AND ALGORITHMS\DSA> node "f:\SEM 4\IT256 DATA STRUCTURES AND ALGORITHMS\DSA\const regularFunction = function (param1.js"
3
5
6
Hello world!
```

➤ 2) Destructuring: -

JS CODE:

```
1  let myNumbers = [1, 2, 3, 4];
2  let [firstNum, secondNum, thirdNum, fourthNum] = myNumbers;
3  console.log(firstNum);
4  console.log(secondNum);
5  console.log(thirdNum);
6  console.log(fourthNum);
7  let myObject = {
8  name: 'John',
9  age: 25,
10 job: 'Software Engineer'
11 };
12 let {name, age, job} = myObject;
13 console.log(myObject.name);
14 console.log(age);
15 console.log(job); // Software Engineer
```

Output:


```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  COMMENTS

2
3
4
John
25
Software Engineer
```

➤ Spread vs Rest Operator: -

JS CODE:

```
1  const numbers = [1,2,3,4,5];
2  function add(a, b, c, d, e) {
3    return a + b + c + d + e;
4  }
5  const sum = add(...numbers);
6  console.log(sum);
7  const numbers2 = [6,7,8,9,10];
8  const combinedNumbers = [...numbers, ...numbers2];
9  console.log(combinedNumbers);
```

Output:

```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  COMMENTS

le.js"
15
[
  1, 2, 3, 4, 5,
  6, 7, 8, 9, 10
]
```

Conclusion: in this practical we learn that how to demonstrate ES6 New Syntax and Test Browser Compatibility.