CUI Abbottabad

Department of Computer Science

Web Technologies

Lecture -11-[CLO-2]
Javascript-Part9

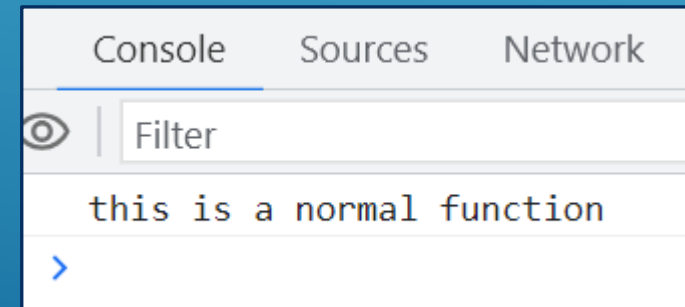**COMSATS University Islamabad, Abbottabad Campus**

Agenda

- ❖ Arrow function
- ❖ Example programs

NORMAL FUNCTION

▸ Let we create a normal function as follows, which only display a message on console

▸ Not that even for the single line of code, we have to put { } around the function body

```
1   <!DOCTYPE html>
2 ▼ <html lang="en">
3 ▸ <head> ••• </head>
10 ▼    <body>
11 ▼        <script>
12      "use strict";
13      //normal function
14 ▼    function myfunc(){
15          console.log("this is a normal function");
16      }
17      myfunc();//this is a arrow function1_1
18
19      </script>
20  </body>
21  </html>
```

Figure: 9.1

Console	Sources	Network

👁 | Filter

> this is a normal function
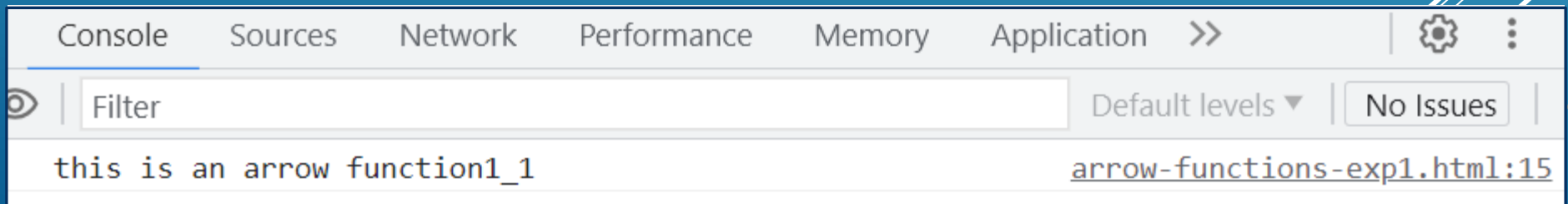
>

COMSATS University Islamabad, Abbottabad Campus

ARROW FUNCTIONS...

▶ An arrow function is much better option single line code function, the arrow function syntax is little change, let here is equivalent arrow function figure:9.2 for the same normal function as shown in figure 9.1

▶ Note that in arrow function we assign the arrow function to a constant variable, which is further called as arrow function name.

▶ The output of this function is also same

Figure: 9.2

```
1    <!DOCTYPE html>
2 ▼  <html lang="en">
3 ▶    <head> ··· </head>
10 ▼      <body>
11 ▼          <script>
12       "use strict";
13       //arrow function-1
14 ▼     const myfunc1_1= ()=>{
15           console.log("this is an arrow function1_1");
16       }
17       myfunc1_1();//this is an arrow function1_1
18
19        </script>
20     </body>
21     </html>
```

| Console | Sources | Network | Performance | Memory | Application | >> | ⚙ | ⋮ |

| 👁 | Filter | Default levels ▼ | No Issues | 4 |

this is an arrow function1_1   arrow-functions-exp1.html:15

ARROW FUNCTIONS...

▶ In the following example-2, the function is returning the message to console.

```
1   <!DOCTYPE html>
2 ▼ <html lang="en">
3 ▶ <head> ··· </head>
9 ▼     <body>
10 ▼        <script>
11      "use strict";
12      //arrow function-2
13 ▼ const myfunc1_2= ()=>{
14          return ("this is an arrow function1_2");
15      }
16      //myfunc1_2();//return nothing
17      console.log(myfunc1_2());//return "this is an arrow function1_2"
18     </script>
19 </body>
20 </html>
```

Figure: 9.3

```
 this is an arrow function1_2                        arrow-functions-exp2.html:17
>
```

ARROW FUNCTIONS...

▶ In the following example-3, we write the function in single line.

▶ one liner arrow function does not require curly braces{} and "return" statement

```
 1  <!DOCTYPE html>
 2▼ <html lang="en">
 3▶ <head> ··· </head>
 9▼     <body>
10▼        <script>
11     "use strict";
12     //one liner arrow function does not require curly braces{} and "return" statement
13     const myfunc1_3= ()=>  "this is an arrow function1_3";
14     console.log(myfunc1_3());//return "this is an arrow function1_3"
15
16     </script>
17  </body>
18  </html>
```

Figure: 9.4

```
  this is an arrow function1_3                              arrow-functions-exp3.html:14
>
```

6

COMSATS University Islamabad, Abbottabad Campus

ARROW FUNCTIONS...

▶ In the following example-4, the function is returning the value using return statement.

▶ Hence { } compulsory around the return statement

```
1   <!DOCTYPE html>
2 ▼ <html lang="en">
3 ▶ <head> ··· </head>
10 ▼     <body>
11 ▼         <script>
12      "use strict";
13      const myfunc1_4= ()=>  { return "this is an arrow function1_4";}
14      //but if you want to use return statement, then {} needed around function
15      console.log(myfunc1_4());//return "this is an arrow function1_4"
16      </script>
17 </body>
18 </html>
```

Figure: 9.5

```
this is an arrow function1_4                                    arrow-functions-exp4.html:15
>
```

7

COMSATS University Islamabad, Abbottabad Campus

ARROW FUNCTIONS...

▶ In the following example-5, the function is returning the object, if we do not put () around the object then it will return "undefined" .

```
1    <!DOCTYPE html>
2 ▼  <html lang="en">
3 ▶  <head> ··· </head>
9 ▼      <body>
10 ▼     <script>
11      "use strict";
12      const myfunc1_5= ()=>  { name:"Muhammad Daraab"};
13  //want to return an object which has name as attrinute and
14  //"Muhammad Daraab" as its value
15      console.log(myfunc1_5()); // it will return "undefined"
16     </script>
17  </body>
18  </html>
```

Figure: 9.6

```
undefined                                    arrow-functions-exp5.html:15
>
```

ARROW FUNCTIONS...

▶ In the following example-6, we put the ()around the object which the function is returning hence now the function will return the object.

```
1   <!DOCTYPE html>
2 ▼ <html lang="en">
3 ▶ <head> ••• </head>
9 ▼     <body>
10 ▼        <script>
11       "use strict";
12     const myfunc1_6= ()=>   ({ name:"Muhammad Daraab"});//put () around the object
13        console.log(myfunc1_6());//will return {name:'Muhammad Darrab'}
14      </script>
15   </body>
16   </html>
```

Figure: 9.7

```
▼ Object ℹ                                           arrow-functions-exp6.html:13
    name: "Muhammad Daraab"
    ▶ [[Prototype]]: Object
```

9

COMSATS University Islamabad, Abbottabad Campus

ARROW FUNCTIONS...

▸ In the following example-7, we send the name "Daraab" as argument to arrow function. Hence we put the variable "name" in () of arrow function. In this this way we can pass argument to arrow function.

Figure: 9.8

```html
1    <!DOCTYPE html>
2 ▼  <html lang="en">
3 ▶  <head> ··· </head>
10 ▼ <body>
11 ▼    <script>
12      "use strict";
13      const myfunc1_7= (name)=>  "welcome in JS "+name;
14      console.log(myfunc1_7("Daraab"));//will return welcome in JS Daraab
15   </script>
16   </body>
17   </html>
```

```
welcome in JS Daraab                          arrow-functions-exp7.html:14
>
```

10

ARROW FUNCTIONS...

▸ In the following example-8, it is same function as example-7 that we omit the () around the arrow function parameter name. It is acceptable in arrow function if only one argument. The function will produces same result as example-7

```
1   <!DOCTYPE html>
2 ▼ <html lang="en">
3 ▶ <head> ••• </head>
9 ▼     <body>
10 ▼        <script>
11      "use strict";
12      const myfunc1_8= name=>  "welcome in JS "+name;
13      //name without (), it is accaeptable in arrow function if only one argument
14      console.log(myfunc1_8("Daraab")); //will return welcome in JS Daraab
15        </script>
16  </body>
17  </html>
```

```
welcome in JS Daraab                              arrow-functions-exp8.html:14
>
```

11

COMSATS University Islamabad, Abbottabad Campus

ARROW FUNCTIONS...

▶ In the following example-9, we are passing two arguments to the arrow function, hence now () are compulsory around the function parameters because now the number of arguments are more than one.

```
1    <!DOCTYPE html>
2 ▼  <html lang="en">
3 ▶  <head> ··· </head>
9 ▼  <body>
10 ▼     <script>
11      "use strict";
12   const myfunc1_9= (firstName,lastName)=>  "welcome in JS "+firstName+" "+lastName;
13   //in case of more than one arguments () are compulsory else get error
14   console.log(myfunc1_9("Muhammad ","Daraab"));//will return welcome in JS Muhammad Daraab

15      </script>
16   </body>
17   </html>
```

Figure: 9.10

```
    welcome in JS Muhammad  Daraab                          arrow-functions-exp9.html:14
>  |
```

COMSATS University Islamabad, Abbottabad Campus