Arrow Function and Callback

Friday, March 11, 2022 4:15 PM

- Arrow function is a ES2015 feature of Java Script
- It allows you to write anonymous function with a simpler syntax
- Known as arrow function or lambda function

Convert Any anonymous Function to arrow function

- Consider a anonymous function factorial

```
var factorial = function( n ) {

    var fn = 1;

    while( n>1)
        fn*=n--;

    return fn;

}
```

- We can write an arrow function by following two simple steps

1. Remove the 'function' keyword
2. Add an arrow (=>) between function parameter (n) and body {}

```
var factorial = ~~function~~( n ) => {

    var fn = 1;

    while( n>1)
        fn*=n--;

    return fn;

}
```

Special Case —> When a function contains a single statement

- If you anonymous function contains a single statement (generally return statement) it can be more simple
- You can also remove
  1. The function body markers {}
  2. 'return' keyword
  3. trailing semicolon (which is anyway optional in javascript)

```
var  mod =  function( x, y ) { return x+y  ; } ;
```

- Lambda function can be created by
  - Removing
    - 'funciton' keyword
    - Function block marker
    - 'return' keyword
    - Simicolon after x+y
  - Adding the arrow key between function argument list and functions one liner logic

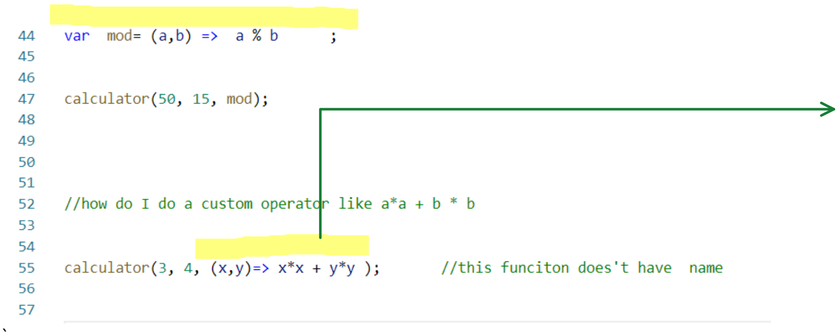
```
var  mod =  ~~function~~( x, y ) => ~~{ return~~ x+y  ~~; }~~ ;
```

```
var  mod =  (x,y) => x+y ;
```

- This arrow function is also known as lambda function or lambda expression.

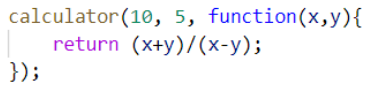
What is the use of lambda function/expression

- They are shorter and simpler than anonymous function
- They are function that look like an expression
- It is easy to pass them to a function as a callback



```
var  mod= (a,b) =>  a % b      ;


calculator(50, 15, mod);




//how do I do a custom operator like a*a + b * b


calculator(3, 4, (x,y)=> x*x + y*y );      //this funciton does't have  name

```

- You can pass a lambda function easily as an argun
- You may also pass regular anonymous function in would look complex and ugly



```
calculator(10, 5, function(x,y){
    return (x+y)/(x-y);
});
```

Limitation of a Lambda Function

- A Lambda function doesn't have **arguments** object
- A Lambda function doesn't have it's own 'this'
    - It can take this from the closure