~~Js Interview~~

9 Js CallBacks

↳ In Js ↳ Functions are objects

↳ Can pass object to function as parameters ?

11

A: Yes

Eg:

```
1  function print(callback){
       callback();
   }
2
```

(function)

↳ CallBack Function

3

↳ Run after another function called

↳ function ↳ pass ↳ another function as parameter.

4

5

↳ used → develop ↳ Asynchronous Js code.

6

↳ handle ↳ Asynchronous operations

Eg:

7

```
<script>
function getData(x, y, callBack)
{
    document.write
    ("Result: "+x+" and "
    +y+" is "+(x*y)
    +"<br>");
    callBack();
}
function ShowData(){
    document.write('method
    callback');
}
getData(20,30,ShowData);
</script>
```

op: Result: 20 & 30 is: 600
method callback

Exp: getData(20, 30, ShowData)

x = 20

y = 30

callback = ShowData()

600

2)

```
<script>
function ShowData(name,
    amt){
    alert("Hi"+name+"\n
    Amt is "+amt);
}
function getData(callBack){
    var name
    =prompt("Name");
    var amt
    =prompt("Amt");
    callBack(name,amt);
}
getData(ShowData);
</script>
```

29 Sunday

call ShowData

```
<script>
  function myDisplay(data){
    document.getElementById("demo").innerHTML=data;
  }
  function call1(){
    myDisplay("Call1");
  }
  function call2(){
    myDisplay("Call2");
  }
  call1();
  call2();
</script>
```

Op: Call1 Call2

WITHOUT CALLBACK

```
<script>
  function myDisplay(data){
    document.getElementById("demo").innerHTML=data;
  }
  function myCal(num1,num2){
    let sum=num1+num2;
    return sum;
  }
  let result=myCal(5,5);
  myDisplay(result);
</script>
```

Op: 10

myDisplay(result) → myCal(5,5) → data → 10

WITH CALLBACK

```
<script>
  function myDisplay(data){
    document.getElementById("demo").innerHTML=data;
  }
  function myCal(num1,num2,myCallBack){
    let sum=num1+num2;
    myCallBack(sum);
  }
  myCal(5,14,myDisplay);
</script>
```

Op: 19

- JS Functions are executed in the sequence they are called.
- Not in sequence they are defined.

JS Variables (pg 69) (cont)

1) Local :
- Declared inside block/function
- Accessible within function/block only.

eg:
```
function ab(){
  var x=10;
}
```

2) Global
- Accessible from any function

wrapper — pass field — input — i
— content — p
— — xs
— <i>
— <span>

CallBack — AsyncTask.html

```
<script>
  // A function that simulates a time consuming task
  function heavyTask(callBack){
      setTimeout (() => {
          console.log ("Heavy Task finished");
          callBack();
      },2000) ;
  }

  //A function that is called after heavy task is completed
  function done(){
      console.log ("Done!..");
  }

  Console.log ("Starting");
  // Calling the Heavy task & passing the done function as
     a callback
  heavyTask(done);
  console.log ("Continuing with the other work!...");
</script>
```

DECEMBER 2020

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

DEC