

Javascript

...

Lecture 5

Eng. Sahar E. Hassan

Javascript: How to debug your code

- Debugging your code is one of the most important skills Javascript developers should have.
- You can use functions such as `console.log()` and `alert()` to debug your code.
- You can also use "break points" which are points that you specify in your code that will cause the code execution to pause on those points and so that you can inspect the values of the variables at that point.
- To add a break point in any line of the code you can use "debugger" keyword

Ex:

```
function myFunc(a,b){  
    var result = a*b;  
    debugger;  
    return result;  
}
```

Javascript: setTimeout() function

- This function is used in order to execute another function after a specified delay (timeout).
- The first argument is the function it should execute after the delay (can be either anonymous or reference to function).
- The second argument is the delay in milliseconds.

Ex:

```
setTimeout(function(){  
  alert("this alert will be shown after 2 seconds");  
},2000)
```

Javascript: setInterval() function

- This function is used in order to execute another function repeatedly every "x" milliseconds
- The first argument is the function it should execute (can be either anonymous or reference to function)
- The second argument is the interval period in milliseconds.

Ex:

```
setInterval(function(){  
  alert("this alert will be shown every 2 seconds");  
,2000)
```

Javascript: clearInterval() function

- clearInterval() function is used in order to stop an interval that was started using setInterval() function

```
var counter = 0;

var timer = setInterval(function(){
    counter++;
    console.log(counter);
    if(counter==3){
        clearInterval(timer);
    }
},1000);
```

Javascript: Animation using javascript

- What are animations?
 - Animation is the change of one or more css property over time
 - For example if we want to animate an element to move it 100px to the right, we should change the "margin-left" property gradually from "0px" to "100px"
- How can we achieve animations using Javascript
 - We can use setInterval() or setTimeout() functions to gradually changing whatever properties we need

Ex: setInterval(function(){

```
    myElement.style.marginLeft = parseInt(myElement.style.marginLeft)+10+"px";
```

```
},10);
```

Javascript: Event bubbling

- When an event occur on an element, the event bubbles to its parents as well.
- For example if we trigger a click event on the nested `<p>` element, it will trigger the click event for all its parents (if the parents have click handler).

```
<div id="parent1" onclick="handleClick()">
```

```
  <div id="parent2" onclick="clickHandler()">
```

```
    <p onclick="handleChildClick()">
```

```
      I am a paragraph
```

```
    </p>
```

```
  </div>
```

```
</div>
```

Javascript: Event bubbling (continue)

- `StopPropagation()`:

It's a function that stop event bubbling, it will trigger the child event but stop the parent's events.

- What is `StopImmediatePropagation()` ?

Javascript: Callback functions

- Functions can be sent to another functions as parameter.
- This technique is usually used in Javascript to allow the users of the function (Another fellow developer) to execute an arbitrary callback function.
- Example:
- If you want to create a function to wrap the functionality of the "confirm()" function and you want to allow users to be able to specify callbacks.

```
EX: function customConfirm(msg,okCallBack,cancelCallBack){  
    var result = confirm(msg);  
    if(result){  
        okCallBack();  
    }  
    else{  
        cancelCallBack();  
    }  
}
```

Javascript: Callback functions

- The callback function can accept and return values that can be used inside the calling function.
- For example imagine you are developing a function called printStr() that accepts a string as a parameter , then formats it and then print the result to the document. Now, if do the formatting logic inside the function itself, then the user of the function (A fellow developer) will not have any way to change the way the formatting works.
- To make the user be able to choose how to format the string, you can design your function to accept another function that would be used to format the string.

EX:

```
function printStr(str,formatFunc){  
    var formattedStr = formatFunc(str);  
    document.write(formattedStr);  
}
```

Javascript: Pass by reference vs Pass by value

- **Passing by value:**

- Passing by value means that when passing an argument to a function, a copy of that value will be created and then passed to the function.
- If the function changes the value of a variable that was passed by value, the original variable will be unchanged.
- Primitive data types (string, number, boolean) are passed by value.

- **Passing by reference:**

- Passing by reference means that when passing an argument to a function, a reference to the original variable is passed.
- If the function changes the value of the variable that was passed by reference, the original variable will also change.
- Complex data types (Arrays and objects) are passed by reference

Javascript: Pass by reference vs Pass by value

```
function increment(a){  a = a+1;  };  
var a=1;  
Increment(a);  
alert(a); //this will alert 1 not 2 because a is passed by value (copied)  
var myArr = [1,2,3,4];  
zeroArr(myArr);  
console.log(myArr);  
function zeroArr(myarr){  
    for(var i=0;i<myarr.length;i++){  
        myarr[i]=0;  
    }  
}
```

Javascript: Date in Javascript

- To create a new date object

```
var now = new Date();
```

```
var longAgo= new Date(1990,10,15); // year, month, day .. Note that the month is 0  
based (0 is januaray, 1 is february)
```

- You can also set the values of the date as follows

```
var myDate = new Date();
```

```
myDate.setDate(15) //will set the day of the month to 15th
```

```
myDate.setMonth(2) //will set the month to March
```

```
myDate.setFullYear(2010) // will set the year to 2010
```

Javascript: Date in Javascript

- Dates in javascript are represented by unix time in milliseconds (epoch time) which is the number of milliseconds passed from 1/1/1970 00:00:00
- To get the epoch time for a date:

```
var myDate = new Date();  
var unixTime = myDate.getTime();
```