
COMP 431/531: Web Development

Lecture 5: Scope and Events

Mack Joyner (mjoyner@rice.edu)

<https://www.clear.rice.edu/comp431>



Announcements & Reminders

- Homework #2 (Dynamic Page) is due **today** at 11:59pm
hw2 repo: <https://classroom.github.com/a/NeQeDxJk>
- Quiz #1 (JavaScript) is due Thursday, Sept. 18th at 11:59pm
- Homework #3 (JavaScript Game) is due Sept. 25th at 11:59pm
hw3 repo: <https://classroom.github.com/a/wZT10Yqv>



JavaScript Functions

```
var namedFunction = function aName() {  
    // This is a comment  
    return 9;  
}  
  
var unnamedFunction = function () {  
    return 7;  
}  
  
function globalFunction() {  
    return 33;  
}
```

```
> namedFunction  
< function aName()  
  
> namedFunction.name  
< "aName"  
  
> unnamedFunction  
< function unnamedFunction()  
  
> unnamedFunction.name  
< ""  
  
> globalFunction  
< function globalFunction()  
  
> globalFunction.name  
< "globalFunction"
```



Java Scope

```
private void go() {  
    Random r = new Random();  
    int sum = 0;  
    int value = 0;  
    int prevValue = 1;  
    for (int ii = 0; ii < 100; ++ii) {  
        value = r.nextInt(10);  
        int product = value * prevValue;  
        prevValue = value;  
        sum += product;  
    }  
    System.out.println("The product was " + product);  
    System.out.println("The sum is " + sum);  
}
```



Java Scope

```
private void go() {  
    Random r = new Random();  
    int sum = 0;
```

```
ScopeExample.java:18: error: cannot find symbol  
        System.out.println("The product was " + product);  
                                                ^
```

```
symbol:   variable product  
location: class ScopeExample  
1 error
```

```
        prevValue = value;  
        sum += product;  
    }  
    System.out.println("The product was " + product);  
    System.out.println("The sum is " + sum);  
}
```



JavaScript has Function Scope

```
function go() {  
  var sum = 0;  
  var value = 0;  
  var prevValue = 1;  
  for (var ii = 0; ii < 100; ++ii) {  
    value = Math.floor(Math.random()*10);  
    var product = value * prevValue;  
    prevValue = value;  
    sum += product;  
  }  
  console.log('The product was ' + product)  
  console.log('The sum is ' + sum)  
}
```

The product was 7

The sum is 2482



Function Scope

Changing outer scope

Declared in global scope

```
> outer3  
< "Bar"
```

```
function innerOuter() {  
  var outer1 = "In Outer Scope"  
  var outer2 = "Also in Outer Scope"  
  
  var internal = function() {  
    var inner = "In Inner Scope"  
    outer1 = "Foo"  
    var outer2 = "Redeclared"  
    outer3 = "Bar"  
    console.log([inner, outer1, outer2, outer3])  
  }  
  internal()  
  console.log([outer1, outer2, outer3])  
  console.log(inner)  
}
```

```
["In Inner Scope", "Foo", "Redeclared", "Bar"]
```

```
["Foo", "Also in Outer Scope", "Bar"]
```

✖ ▼ Uncaught ReferenceError: inner is not defined

innerOuter @ scope.js:30

(anonymous function) @ scope.js:33



Block Scope with Let

```
function innerOuterBlock() {  
  var outer1 = "In Outer Scope"  
  var outer2 = "Also in Outer Scope"  
  
  var internal = function() {  
    // block scope with let  
    {  
      let inner = "In Inner Scope"  
      outer1 = "Foo"  
    }  
    //console.log("let does not hoist: " + outer2)  
    let outer2 = "Redeclared"  
    var outer3 = "Bar" // no auto-global scope  
    console.log([inner, outer1, outer2, outer3])  
  }  
  internal()  
}
```

Uncaught ReferenceError: inner is not defined



Variable Hoisting

```
function go() {  
  var sum = 0;  
  var value = 0;  
  var prevValue = 1;  
  for (var ii = 0; ii < 100;  
    value = Math.floor(Math.random()*10);  
    var product = value * prevvalue = value;  
    sum += product;  
  }  
  console.log('The product was ' + product)  
  console.log('The sum is ' + sum)  
}
```

```
function go() {  
  var product, ii  
  var sum = 0;  
  var value = 0;  
  var prevValue = 1;  
  for (ii = 0; ii < 100; ++ii) {  
    value = Math.floor(Math.random()*10);  
    product = value * prevValue;  
    prevValue = value;  
    sum += product;  
  }  
  console.log('The product was ' + product)  
  console.log('The sum is ' + sum)  
}
```



Variable Hoisting

```
hoist()
function hoist() {
  console.log("Inside hoist()", a)
  var a = "Hoist Me!"
}
hoist()

hoist2()
var hoist2 = function() {
  hoist()
}
```

Inside hoist() undefined

Inside hoist() undefined

► Uncaught TypeError: hoist2 is not a function



Variable Hoisting

```
hoist()
function hoist() {
  console.log("Inside hoist()", a)
  var a = "Hoist Me!"
}
hoist()

hoist2()
var hoist2 = function() {
  hoist()
}
```

a is hoisted, but has no value

hoist2 is hoisted, but has no value
and therefore is not a function yet

Inside hoist() undefined

Inside hoist() undefined

► Uncaught TypeError: hoist2 is not a function



Global Scope

```
2
3
4 console.log('What is this?', this)
5 console.log('this is window', this === window)
6
7 function myFunction() {
8     noVar = 'abc'
9     var varred = '123'
10    console.log(`in myFunction noVar=${noVar} varred=${varred}`)
11
12    console.log('myFunction this is window', this === window)
13 }
14
15 myFunction()
16 console.log(`after myFunction call noVar=${noVar}`)
17 console.log('What is window.noVar?', window.noVar)
18
```

What is this?

Window {speechSynthesis: SpeechSyn
▶ localStorage: Storage, sessionStorage
DeprecatedStorageInfo...}

this is window true

in myFunction noVar=abc varred=123

myFunction this is window true

after myFunction call noVar=abc

What is window.noVar? abc



Global Scope

```
2 'use strict' // run with and without!
3
4 console.log('What is this?', this)
5 console.log('this is window', this === window)
6
7 function myFunction() {
8   noVar = 'abc'
9 }
```

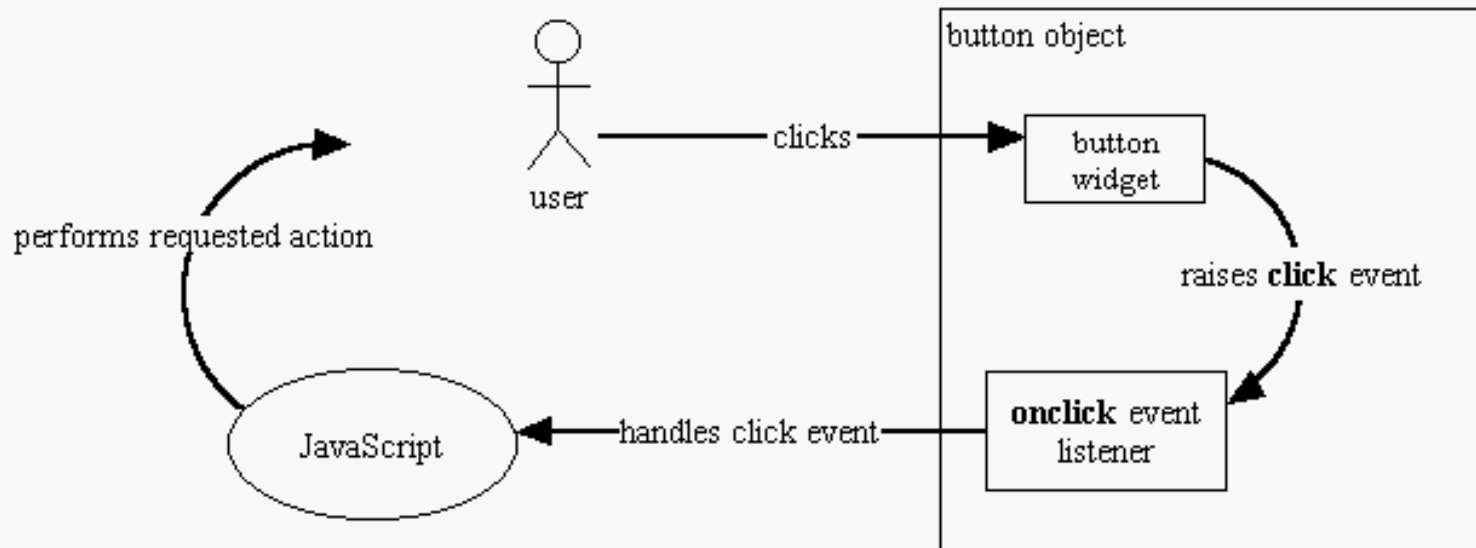
What is this?
Window {speechSynthesis: SpeechSynthesis, caches: CacheStorage, ▶ localStorage: Storage, sessionStorage: Storage, webkitStorageInfo: DeprecatedStorageInfo...}
this is window true
in myFunction noVar=abc varred=123
myFunction this is window true
after myFunction call noVar=abc
What is window.noVar? abc

```
9 What is this? javascript-scope.html:4
10 Window {speechSynthesis: SpeechSynthesis, caches: CacheStorage,
11 ▶ localStorage: Storage, sessionStorage: Storage, webkitStorageInfo:
12   DeprecatedStorageInfo...}
13 this is window true javascript-scope.html:5
14 ✖ ▶ Uncaught ReferenceError: noVar is not javascript-scope.html:8
15   defined
16
17
18
```



Events

```
<button value="Click Me" onclick="alert('Thank you')" />
```



DOM Level 1 Event Types: on<type>

- blur / focus (element loses / receives focus)
- change (form field value changes)
- click (mouse down and mouse up on one element)
- contextmenu (right-click)
- keydown / keyup (depress / release a key, repeats)
- keypress (character key id depressed, repeats)
- mousedown / mouseup (depress / release mouse button)
- mousemove (mouse in motion)
- mouseover / out (mouse enters / leaves an element)
- reset, copy, paste, submit, ...

```
<!DOCTYPE html>
<html>
<body>

<p>Write something in the text field to trigger a function.</p>

<input type="text" id="myInput" oninput="myFunction()">

<p id="demo"></p>

<script>
function myFunction() {
    var x = document.getElementById("myInput").value;
    document.getElementById("demo").innerHTML = "You wrote: " + x;
}
</script>

</body>
</html>
```

https://www.w3schools.com/jsref/event_oninput.asp

<http://www.quirksmode.org/dom/events>



DOM Level 2 Event

- Allows for multiple registration of handlers
 - level 1 only has one handler
- Allows for handler removal


useCapture

true = execute during the *capture phase*

false = (default) execute during the *bubble phase*

```
// add level 2 handlers
var l2p = document.getElementById("level2p")

l2p.addEventListener("click", l2pResponse, false);
l2p.addEventListener("click", l2pColorFn, false);
l2p.addEventListener("click", l2pBgColorFn, false);
l2p.removeEventListener('click', l2pBgColorFn)
```



- Allows event listeners on the *document* (e.g. key pressed)

<https://www.clear.rice.edu/comp431/sample/hello-events.html>



Event Objects

Events with listeners return Event objects

- MouseEvent (e.g. click, mouseover)
- KeyboardEvent (e.g. keyup) has *key* field
 - e.*key* returns “Shift” for keyup when Shift key is released
- InputEvent (e.g. input)
- ...there are others

```
1 'use strict';
2
3 /**
4  * setup the event listener
5  */
6 function setup() {
7     document.getElementById("myBtn").addEventListener("click", handleClickedButton);
8 }
9
10 /**
11  * Handle the button click event
12  * @param e The mouse event
13  */
14 function handleClickedButton(e) {
15     let btnText = document.getElementById("myBtn");
16
17     alert("Event " + e + " just happened for mouse click on " + btnText.innerText +
18 }
```

