



CS472 WAP

# Scope Chain

## Scope, Closures, and Encapsulation

Except where otherwise noted, the contents of this document are Copyright 2012 Marty Stepp, Jessica Miller, Victoria Kirst and Roy McElmurry IV. All rights reserved. Any redistribution, reproduction, transmission, or storage of part or all of the contents in any form is prohibited without the author's expressed written permission. Slides have been modified for Maharishi University of Management Computer Science course CS472 in accordance with instructors agreement with authors.

# **Maharishi International University**

## **Fairfield, Iowa** © 2020



All rights reserved. No part of this slide presentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or recording, or by any information storage and retrieval system, without permission in writing from Maharishi International University.

# Main Point Preview

- When we ask for any variable, JS will look for that variable in the current scope. If it doesn't find it, it will consult its outer scope until we reach the global scope.
- *Science of Consciousness: The scope chain proceeds from local awareness to more global awareness. During the process of transcending we naturally proceed from local awareness to more subtle levels of awareness to unbounded awareness.*

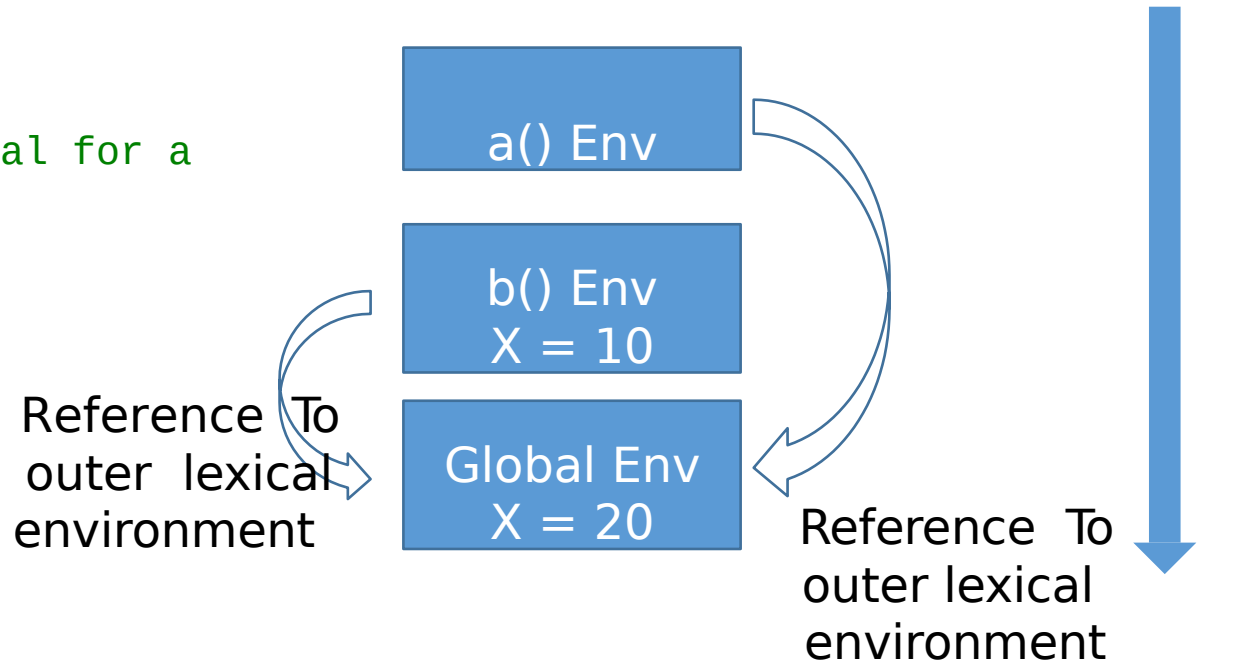
# Scope Chain

- When we ask for any variable, JS Engine will look for that variable in the current scope. If it doesn't find it, it will consult its outer scope until we reach the global scope.

# Scope Example1



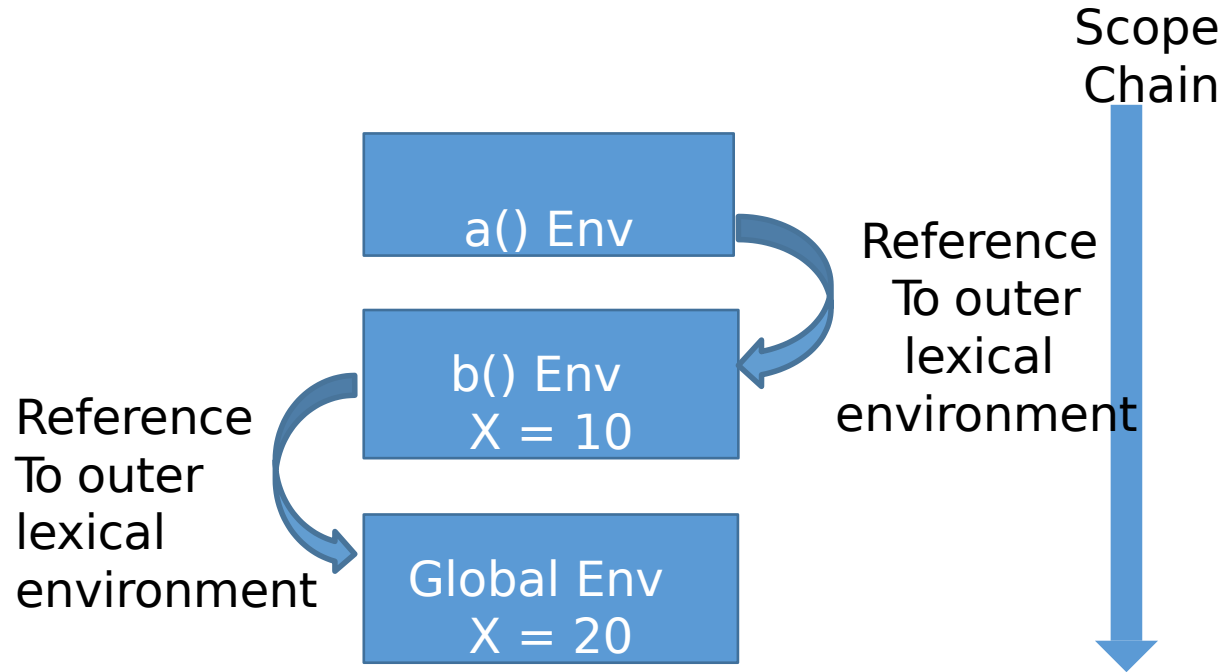
```
function a() {  
  console.log(x); // consult Global for x and print 20 from Global  
}  
function b() {  
  var x = 10;  
  a(); // consult Global for a  
}  
var x = 20;  
b();
```



# Scope Example 2: Inner function

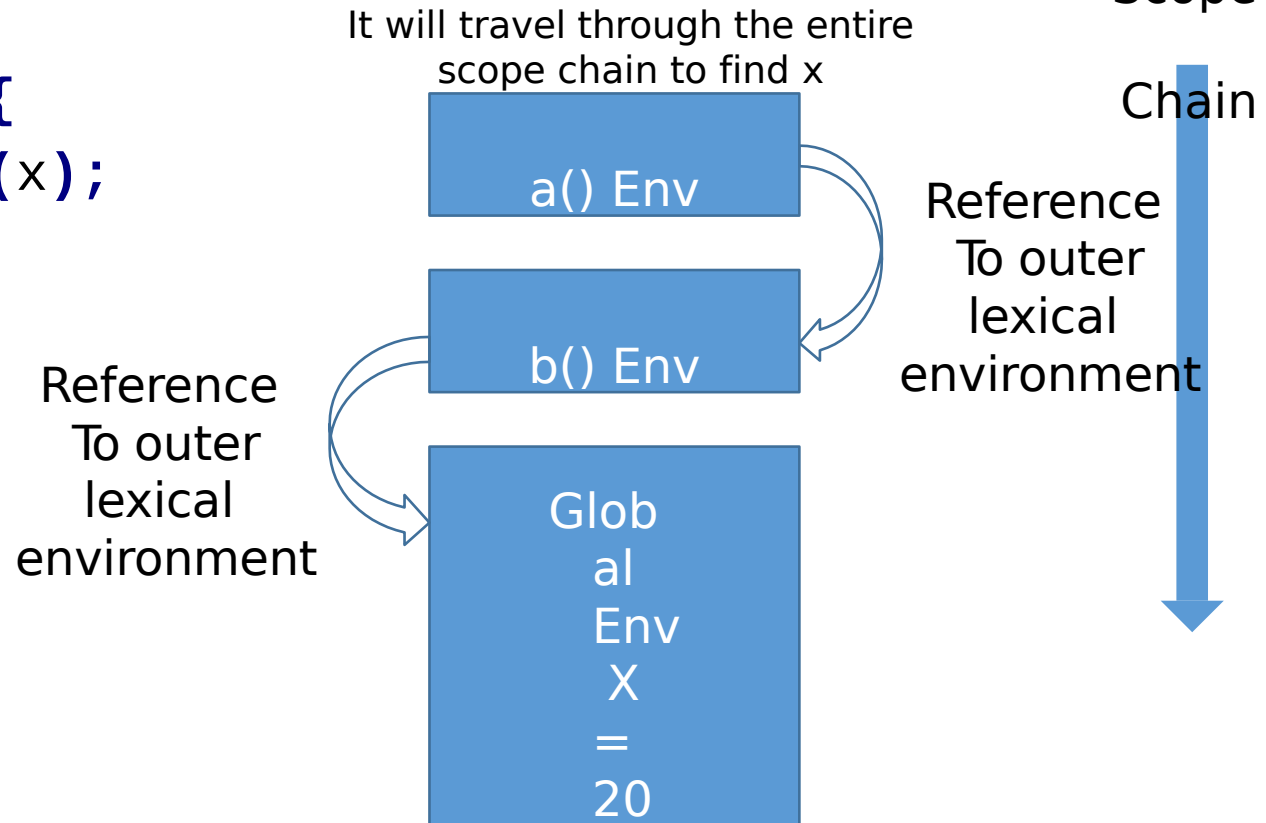


```
function b() {  
  function a() {  
    console.log(x);  
  }  
  var x = 10;  
  a();  
}  
var x = 20;  
b();
```



# Scope Example 3

```
function b() {  
  function a() {  
    console.log(x);  
  }  
  a();  
}  
var x = 20;  
b();
```





# Scope Example 3



```
function f() {  
  var a = 1, b = 20, c;  
  console.log(a + " " + b + " " + c);  
  function g() {  
    var b = 300, c = 4000;  
    console.log(a + " " + b + " " + c);  
    a = a + b + c;  
    console.log(a + " " + b + " " + c);  
  }  
  console.log(a + " " + b + " " + c);  
  g();  
  console.log(a + " " + b + " " + c);  
}  
f();
```



# Scope Example 4

```
var x = 10;
function main() {
  console.log("x1 is:" + x);
  x = 20;
  console.log("x2 is:" + x);
  if (x > 0) {
    var x = 30;
    console.log("x3 is:" + x);
  }
  console.log("x4 is:" + x);
  var x = 40;
  var f = function(x) {
    console.log("x5 is:" + x);
  };
  f(50);
  console.log("x6 is:" + x);
}
main();
console.log("x7 is:" + x);
```

# Main Point

- When we ask for any variable, JS will look for that variable in the current scope. If it doesn't find it, it will consult its outer scope until we reach the global scope.
- *Science of Consciousness: The scope chain proceeds from local awareness to more global awareness. During the process of transcending we naturally proceed from local awareness to more subtle levels of awareness to unbounded awareness.*

