



CHAP 10: Closures in JS

"Function bundled with its local environment is known as a closure."

OR

"Function along with its local scope forms a closure".

```
function x() {  
  var a = 7;  
  function y() {  
    console.log(a);  
  }  
  y();  
}
```

Output
7

* * So, in this example you can see that the function y() forms a closure with the variables of function x's local scope. * *

• Case II:

```
function x() {  
  var a = 7;  
  function y() {  
    console.log(a);  
  }  
}
```



DATE _____

```
return y;
}
var z = x();
console.log(z);
z();
```

So, we have returned 'y' and note it's outside x(), hence closes closure into picture, so when a function is returned it still maintains its logical scope, though 'x' no longer exists, it will still remember the value of 'a'. So, in simple terms, when funcⁿ y is returned not only funcⁿ but whole closure is returned.

**

• Uses of Closure:

- 1) Module Design Pattern
- 2) Caching
- 3) Functions like once
- 4) Memoize
- 5) maintaining state in async world
- 6) set timeouts
- 7) Iterators