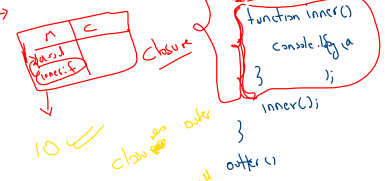


Closures

func along with its lexical scope
forms a closure.

```
function outer() {
  var a = 10;
  function inner() {
    console.log(a);
  }
  return inner;
}
```



outer \rightarrow LS = LA + outer
 inner \rightarrow LA + inner
 inner.call \rightarrow LA + inner

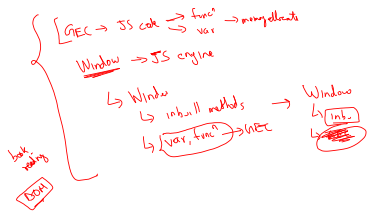
```
function outer() {
  var a = 10;
  function inner() {
    console.log(a);
  }
  inner();
}
```



JS engine

- 1) JS engine creates a new object
- 2) stack & push
- 3) JS engine creates a new object
- 4) JS engine creates a new object
- 5) JS engine creates a new object
- 6) JS engine creates a new object
- 7) JS engine creates a new object
- 8) JS engine creates a new object
- 9) JS engine creates a new object
- 10) JS engine creates a new object

JS engine \rightarrow this request is created on a global scope
 this points to the window object



objects

this . first

LinkedIn
 Abstract w/ Peptology
 demo video
 HT

var \rightarrow hoisting
 const \rightarrow in JS memory is allocated to variables & func even w/o executing a single line of code.

JS { 1) let 2) const }

let \rightarrow Script:
 const let \rightarrow var
 const let \rightarrow var
 const let \rightarrow var

Script:
 let \rightarrow time blow
 let \rightarrow time blow
 let \rightarrow time blow