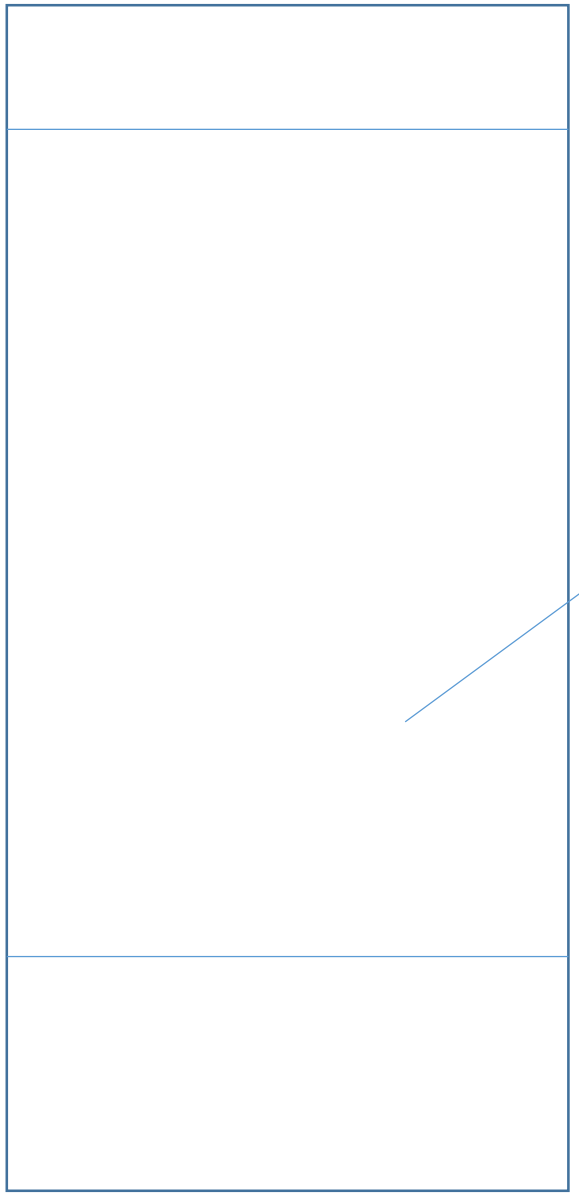
Local Scopes

By John Diyala

*pseudo-code was used for a simpler explanation



Heap Memory

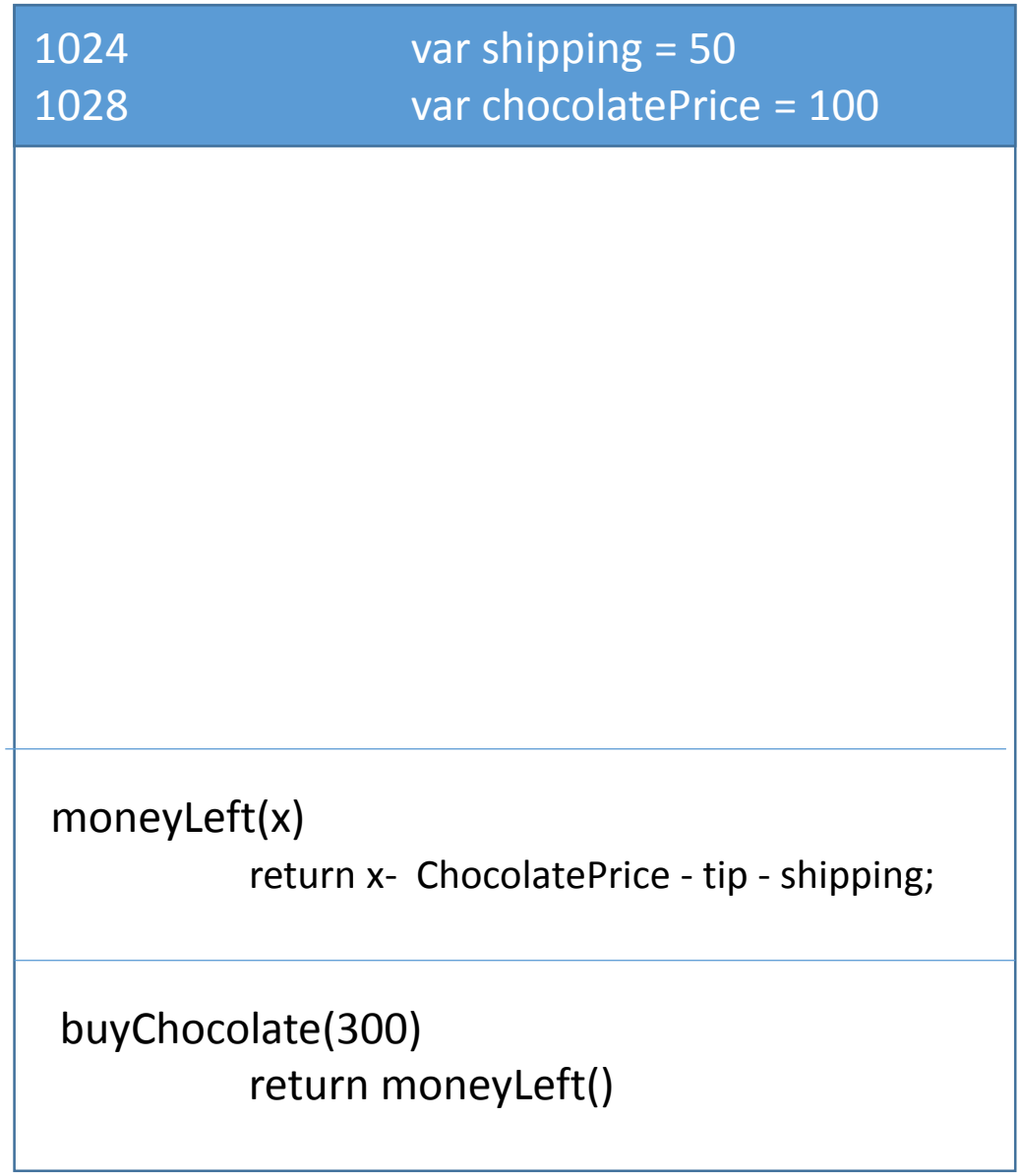
Available memory (can be used as stack or heap memory as needed)

Stack Memory

shipping was already a global variable
chocolatePrice is placed on the heap when buyChocolate is called

console.log(buyChocolate(300));

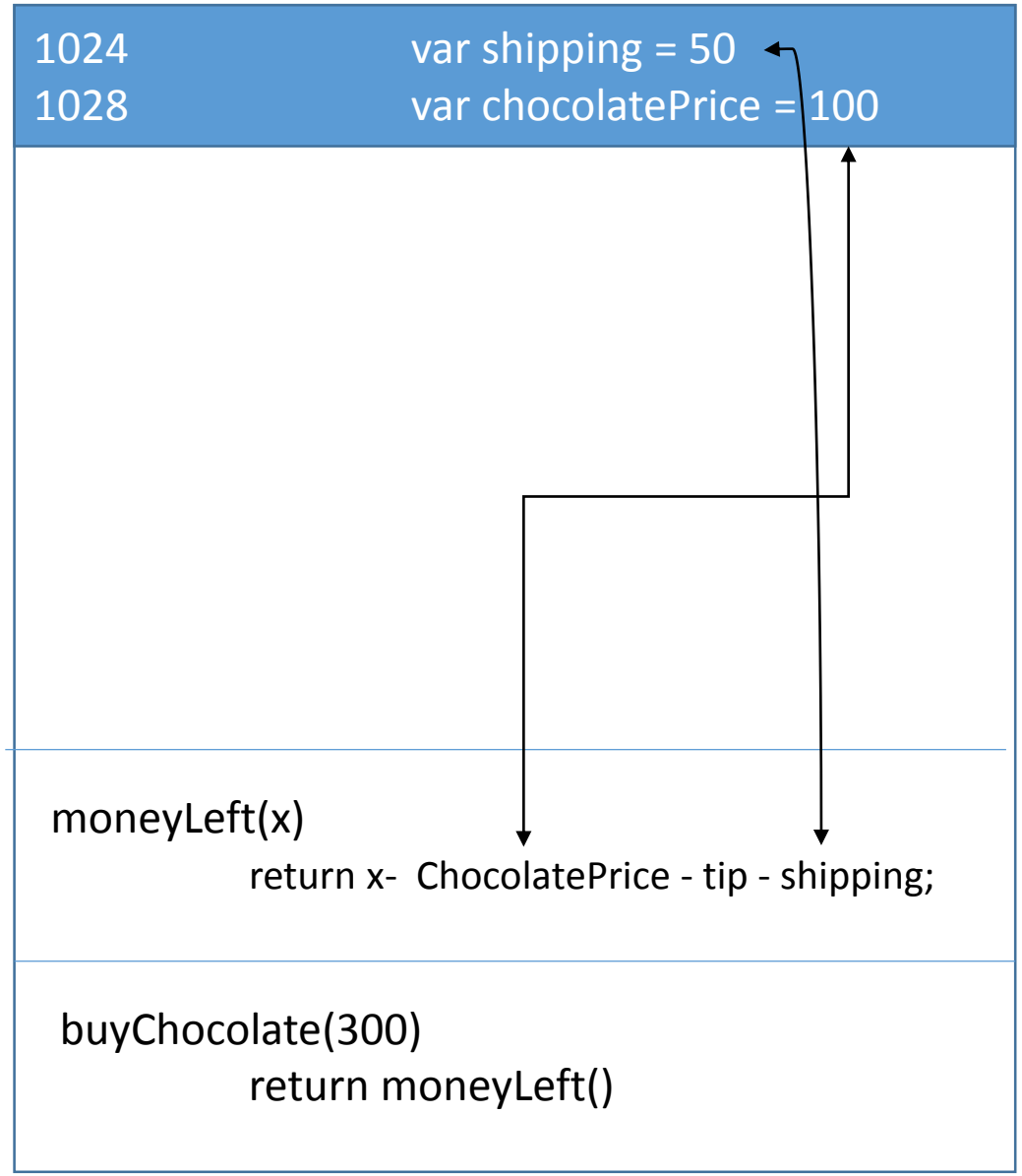
- buyChocolate (is called and placed on the stack)



shipping was already a global variable
chocolatePrice is placed on the heap when buyChocolate is called

```
function buyChocolate(x) {
  var ChocolatePrice = 100;

  function moneyLeft(x) {
     var tip = 10;
     return x-  ChocolatePrice - tip - shipping;
  }

  return moneyLeft(x);
}
```

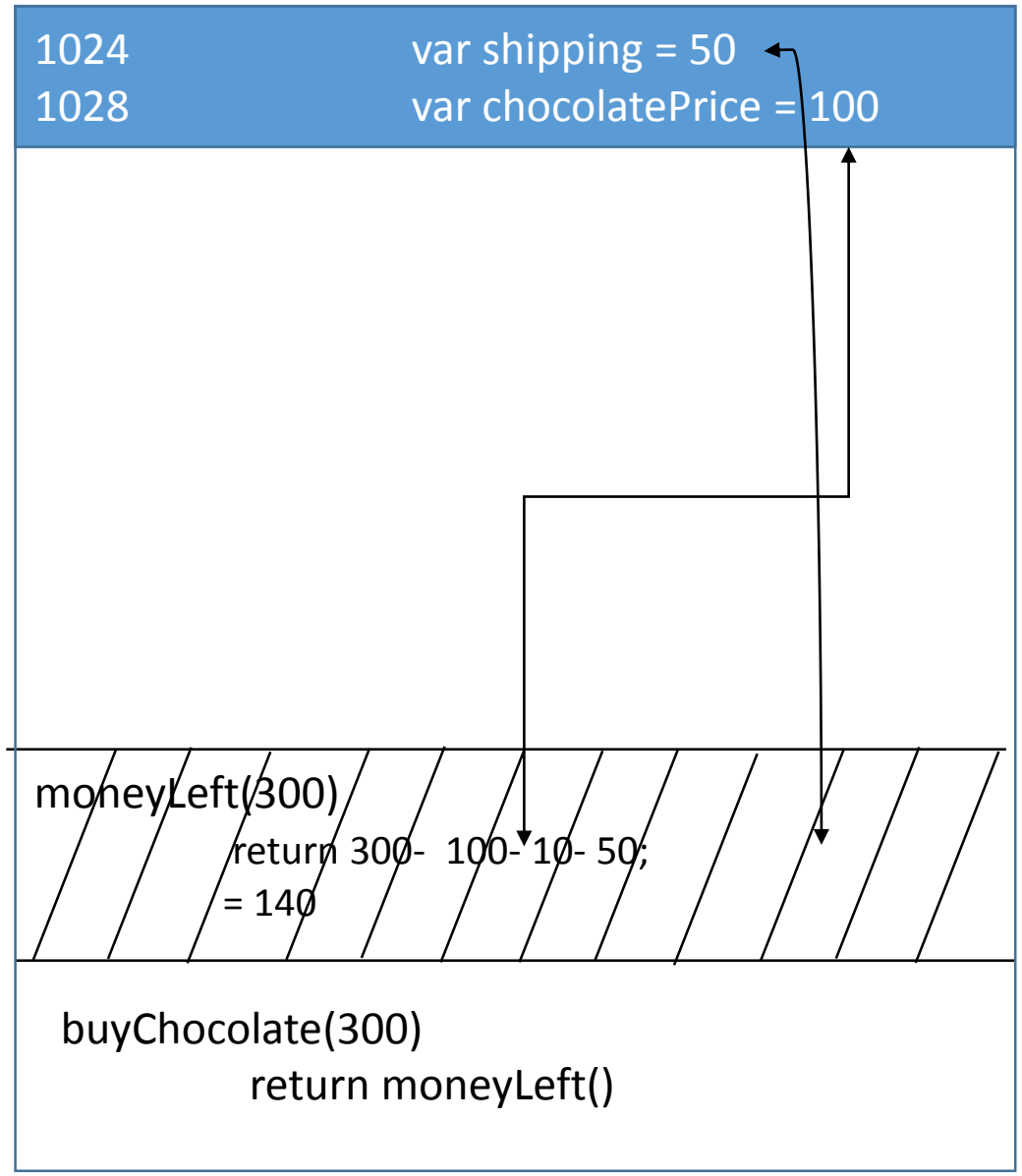
moneyLeft(x) needs to be computed to calculate buyChocolate(300), so it is placed on the stack



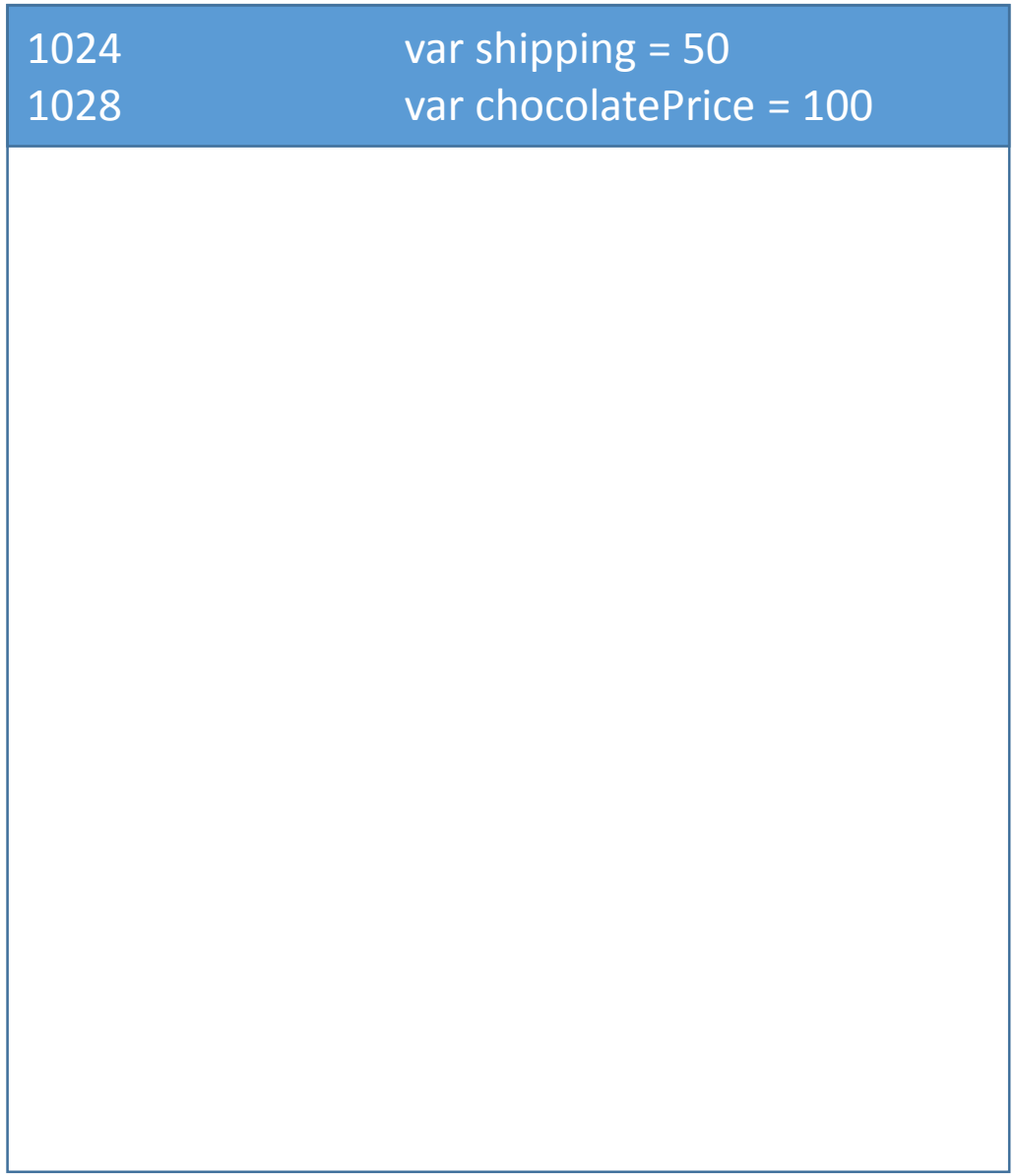
shipping was already a global variable
chocolatePrice is placed on the heap when buyChocolate is called

```
function buyChocolate(x) {
  var ChocolatePrice = 100;

  function moneyLeft(x) {
     var tip = 10;
     return x-  ChocolatePrice - tip - shipping;
  }

  return moneyLeft(x);
}
```

moneyLeft(x) needs to be computed to calculate buyChocolate(300), so it is placed on the stack, it is able to access variables outside its local scope because those variables are stored on the heap.



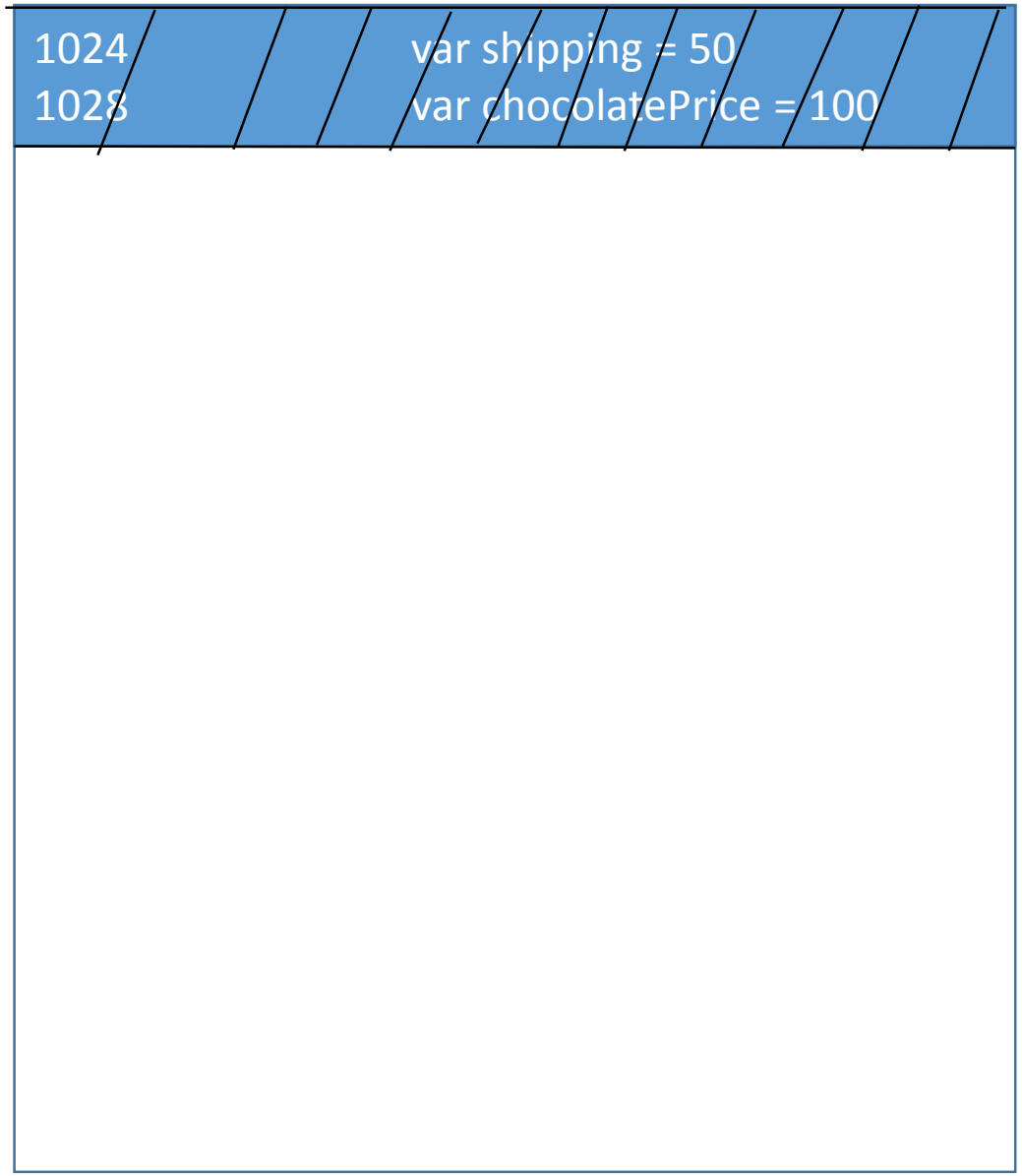
moneyLeft(300) evaluates to 140, now that it has executed, moneyLeft(x) can be popped off the stack



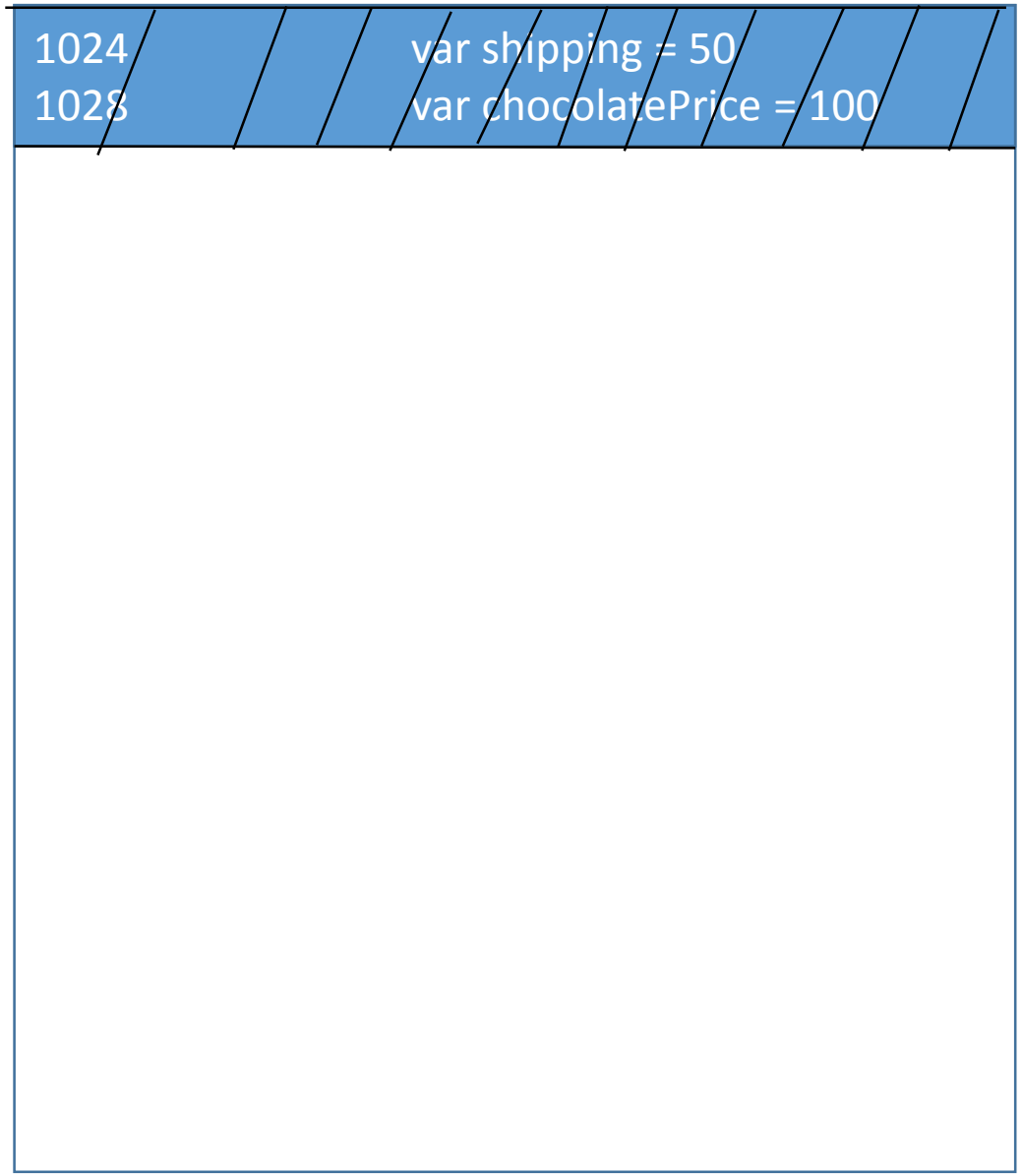
buyChocolate(300)

return moneyLeft() = 140

buyChocolate (300) returns 140, and is also popped off the stack, leaving it empty.



Since there are no references to these variables on the heap anymore, they are freed by the garbage collector



Since there are no references to these variables on the heap anymore, they are freed by the garbage collector

R2.3)

- Describe what modifications to the stack and heap model provided, if any, are necessary to convey how recursion works with javascript functions. That is, can you illustrate your recursion using the model provided or does it fall short and require modifications or new features?

- In order to be able to access variables outside a function's scope from within that function, I had to modify the stack/heap to store variables in the heap. If I had stored these variables in the stack, in order to reach them again, I would have to pop off everything above, the function whose variables I want, which does not make sense in this example since the function at the top of the stack needs variables from a function deeper down in the stack.
  Therefore, in order to be able to access such variables, these variables had to be stored in heap memory.