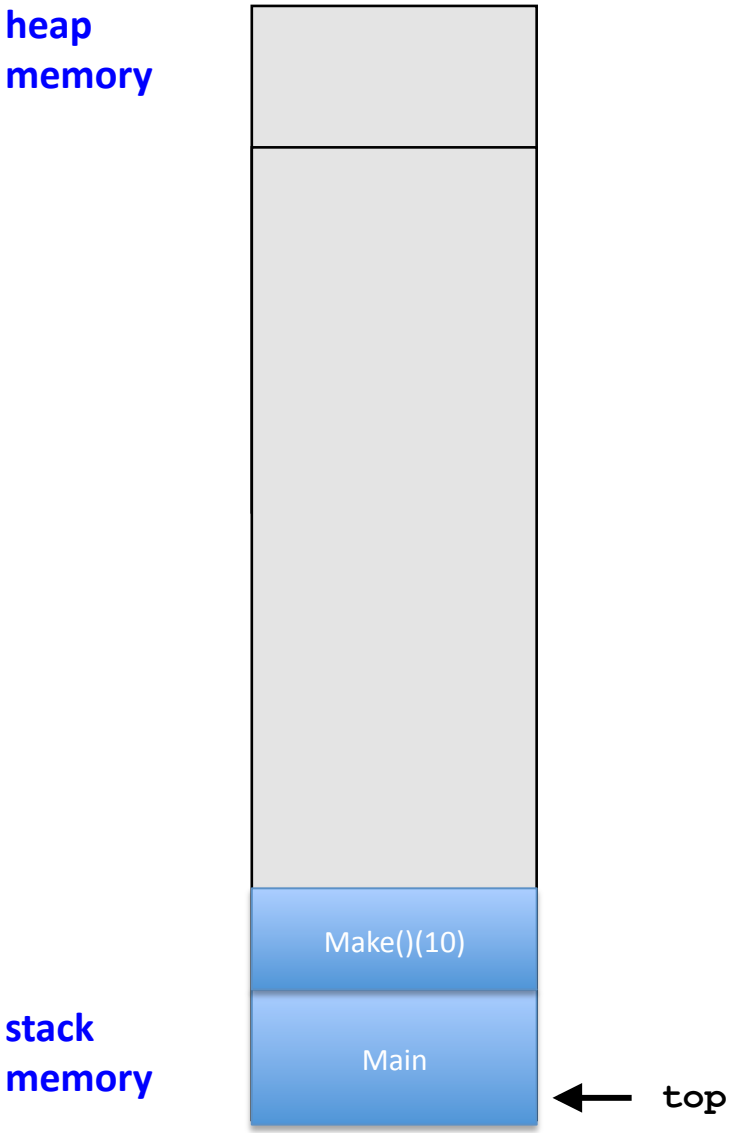
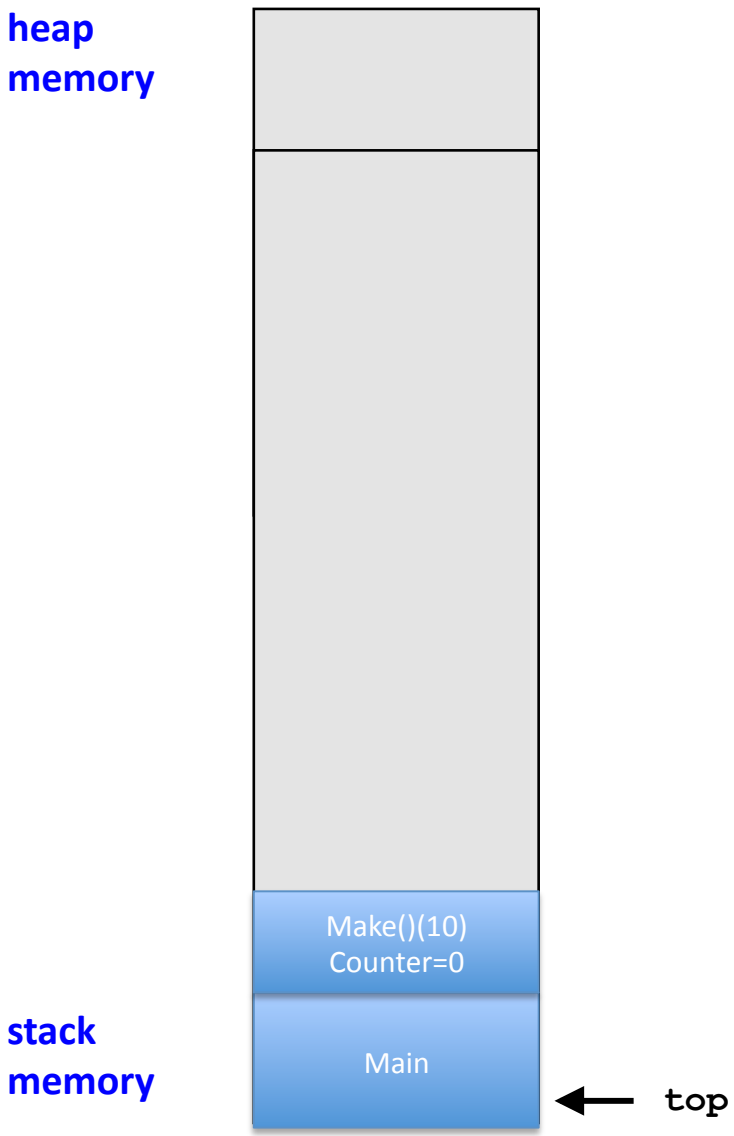
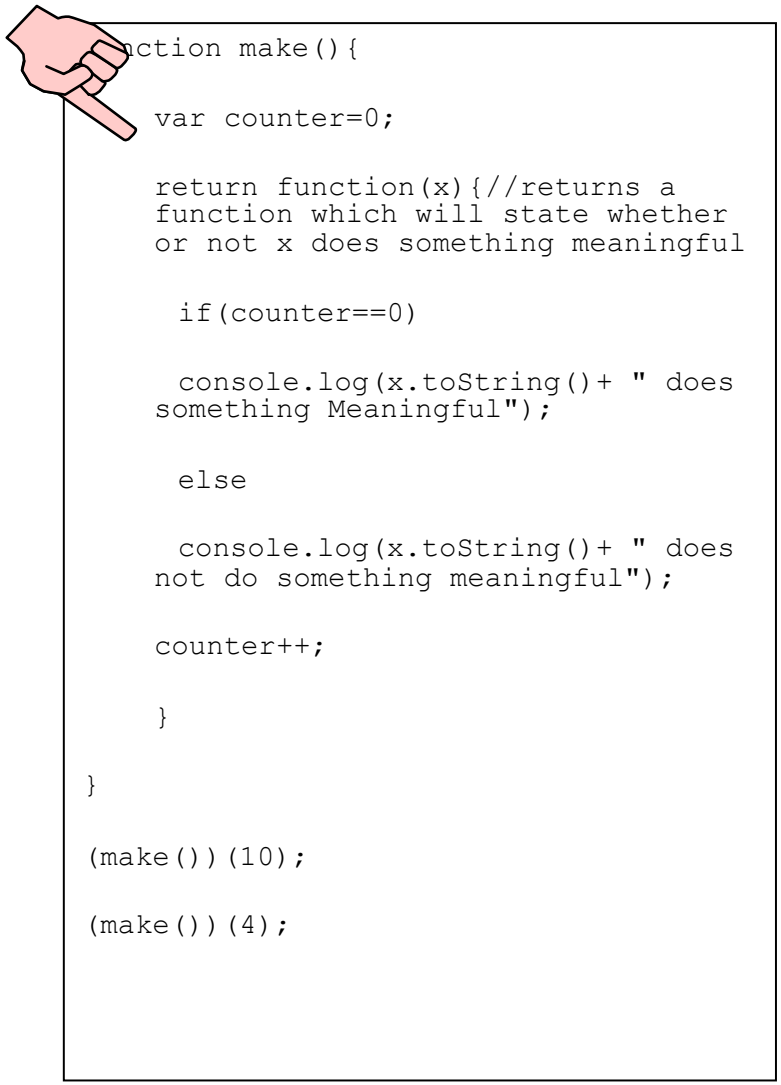
Javascript Closures

R3.2/3.3

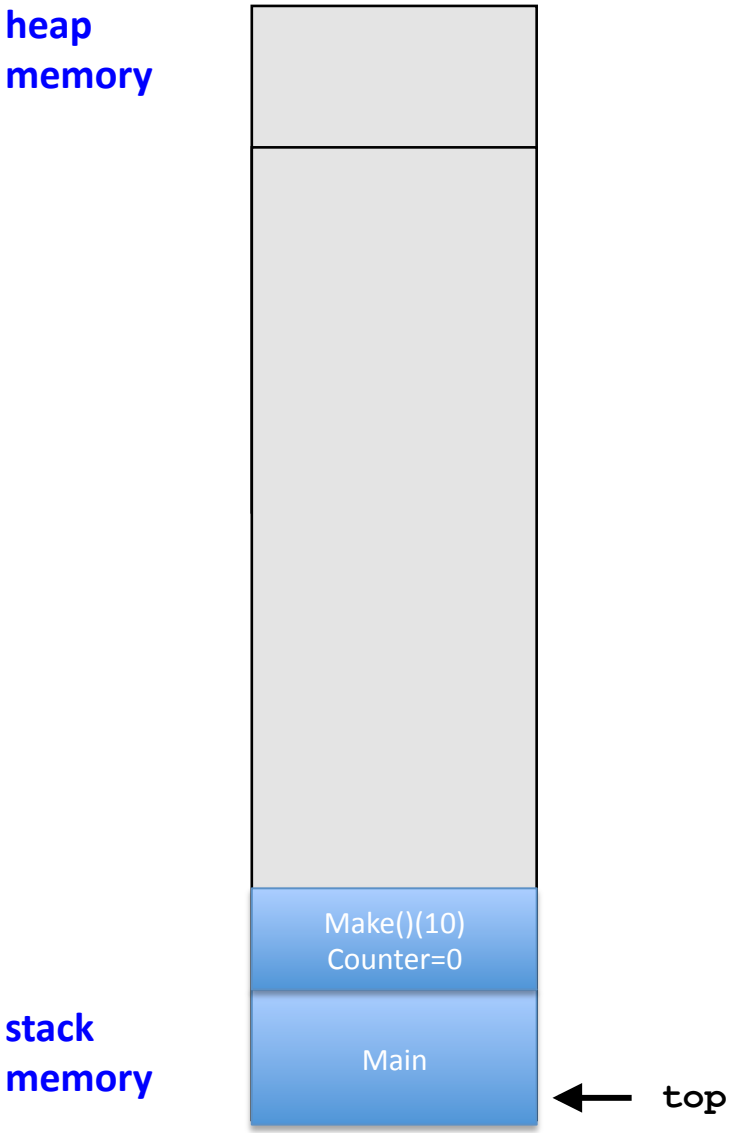



```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

      if(counter==0)

      console.log(x.toString()+ " does
    something Meaningful");

      else

      console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```

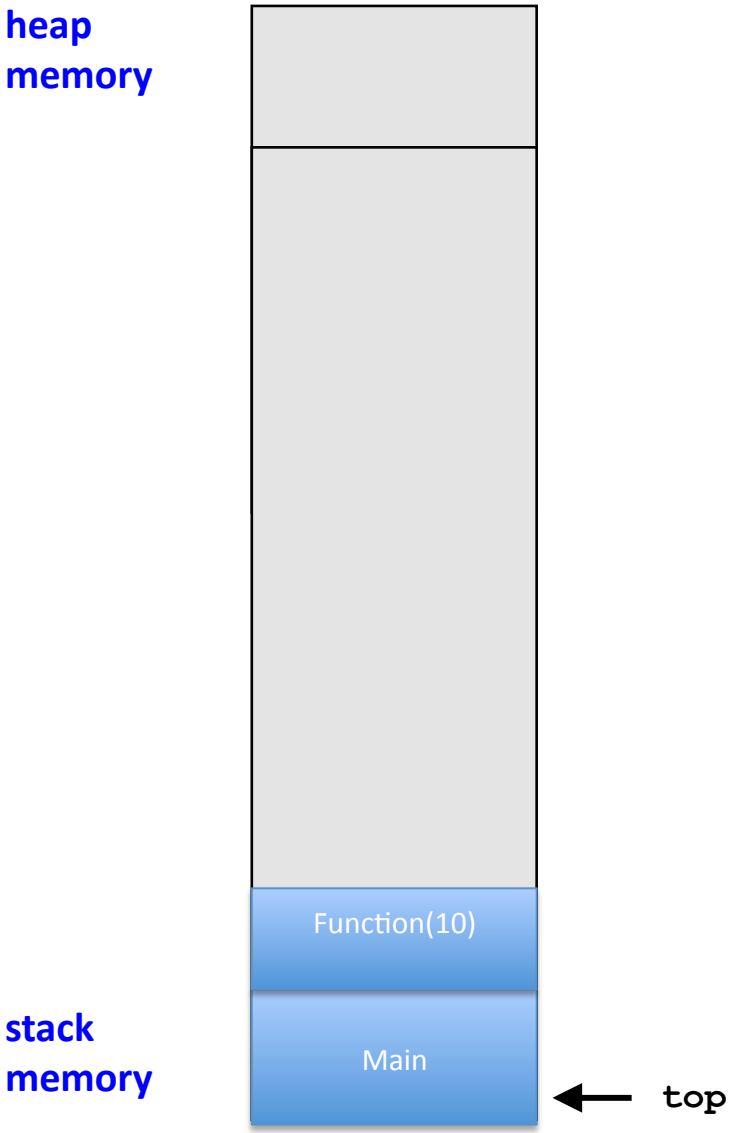




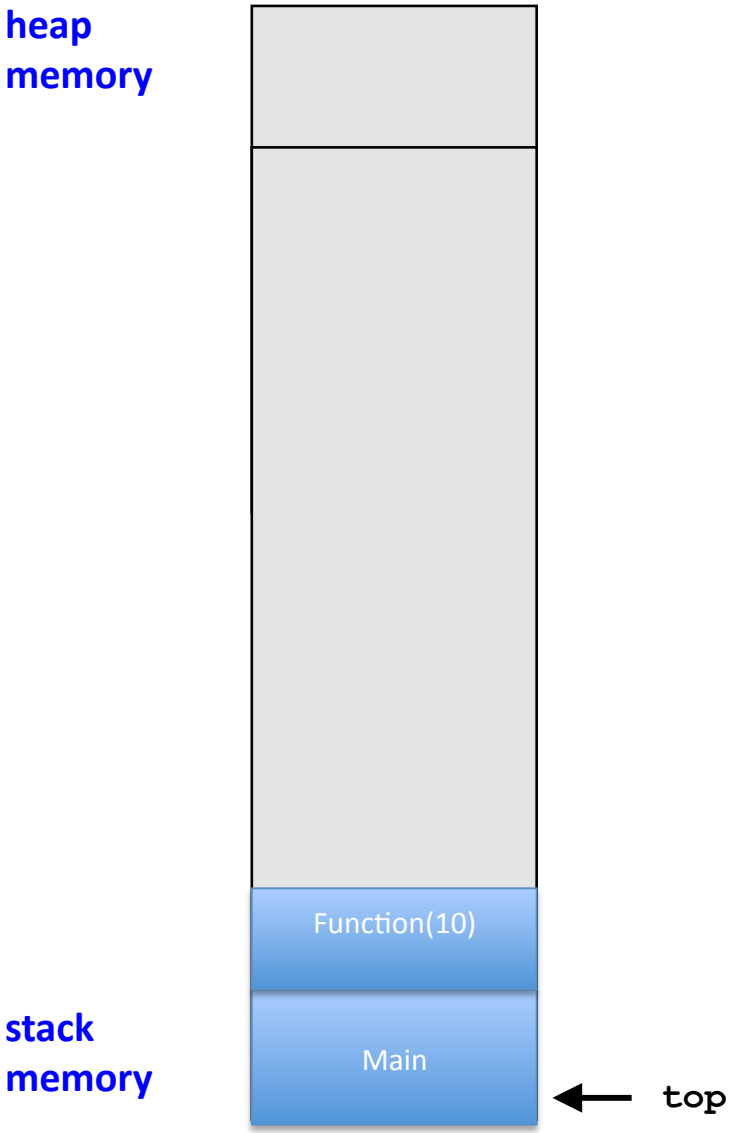
```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

      if(counter==0)

      console.log(x.toString()+ " does
    something Meaningful");

      else

      console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```



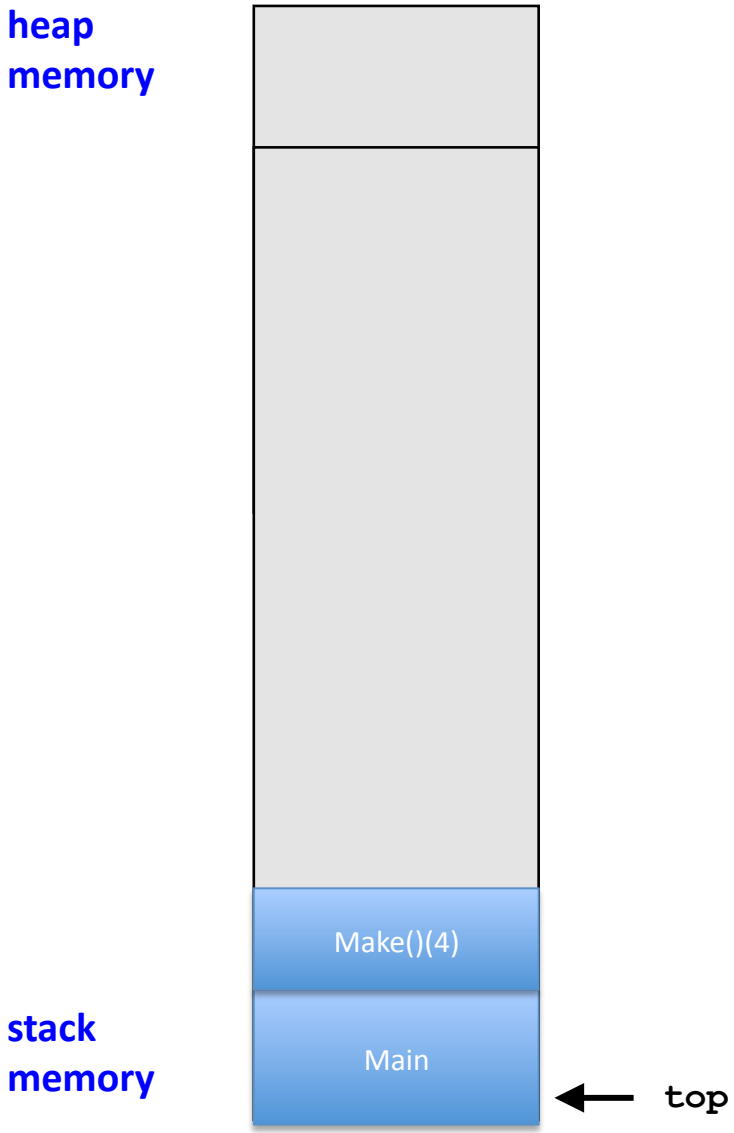

```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

      if(counter==0)

      console.log(x.toString()+ " does
    something Meaningful");

      else

      console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```



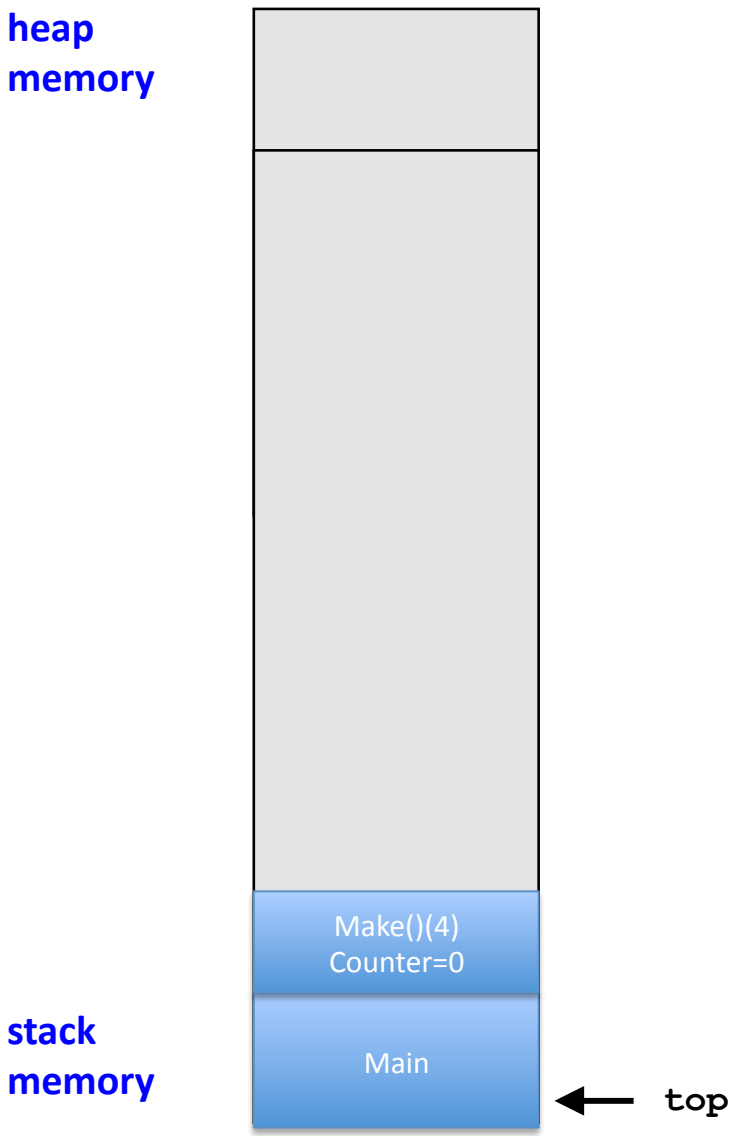

```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

     if(counter==0)

     console.log(x.toString()+ " does
    something Meaningful");

     else

     console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```



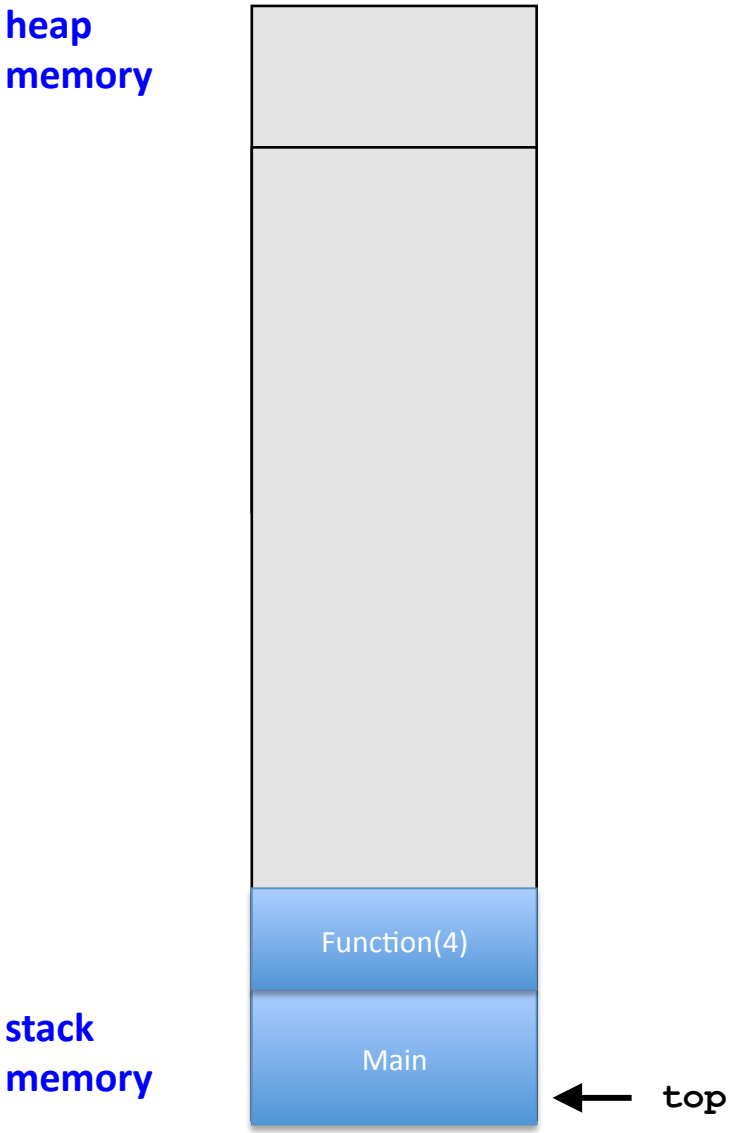

```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

      if(counter==0)

      console.log(x.toString()+ " does
    something Meaningful");

      else

      console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```



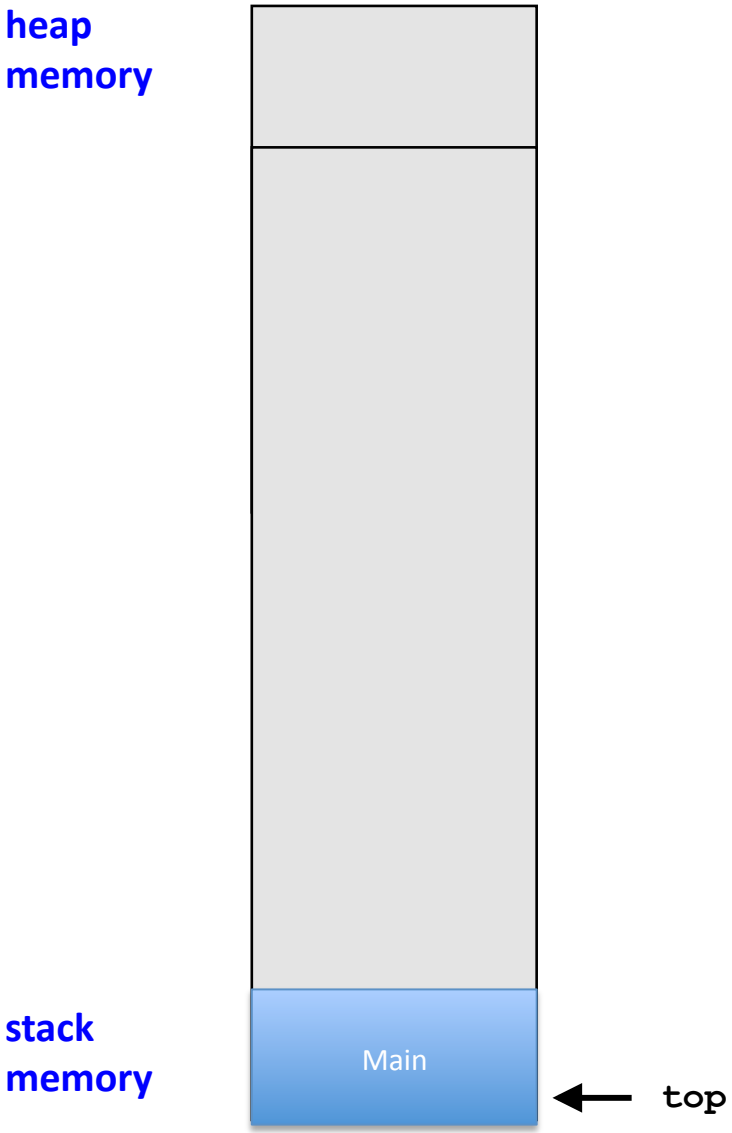

```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

     if(counter==0)

     console.log(x.toString()+ " does
    something Meaningful");

     else

     console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```




```
function make(

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

      if(counter==0)

      console.log(x.toString()+ " does
    something Meaningful");

      else

      console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```




```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

     if(counter==0)

     console.log(x.toString()+ " does
    something Meaningful");

     else

     console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```




```
function make(){

    var counter=0;

    return function(x){//returns a
    function which will state whether
    or not x does something meaningful

     if(counter==0)

     console.log(x.toString()+ " does
    something Meaningful");

     else

     console.log(x.toString()+ " does
    not do something meaningful");

    counter++;

    }

}

(make())(10);

(make())(4);
```

R3.3

- I found that our example didn’t really have a way of using functions that were returned as values. I just used Function(x) which isn’t exactly true but I found it did a reasonable job of explaining what was going on.
- I personally had trouble with this question because I was unsure of what simple useful purposes this serves.