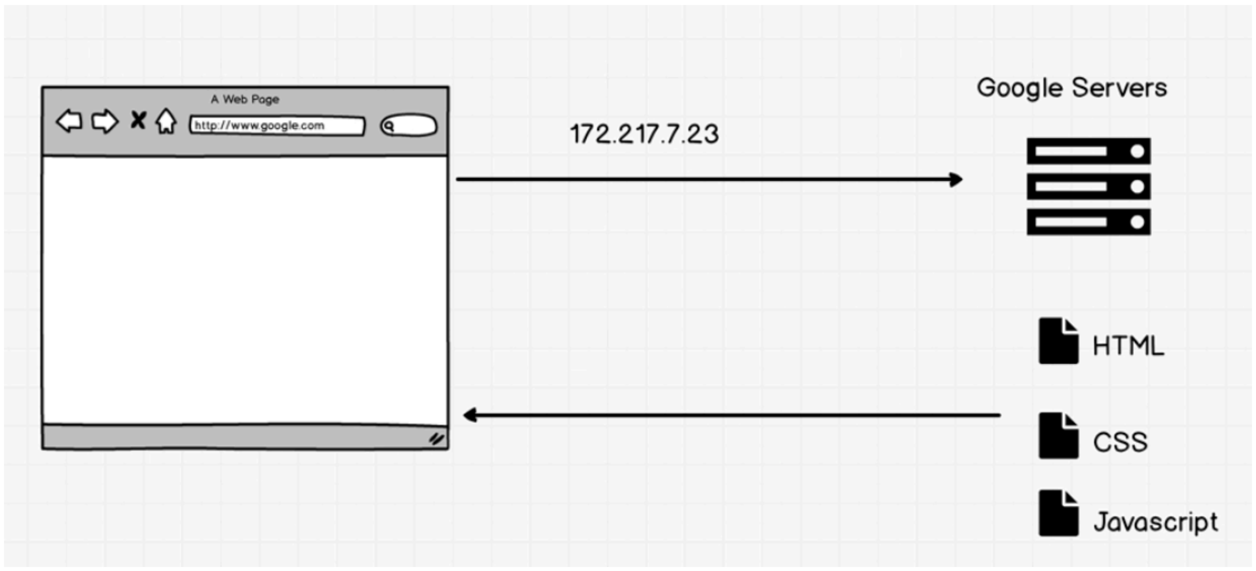
JavaScript: Functions

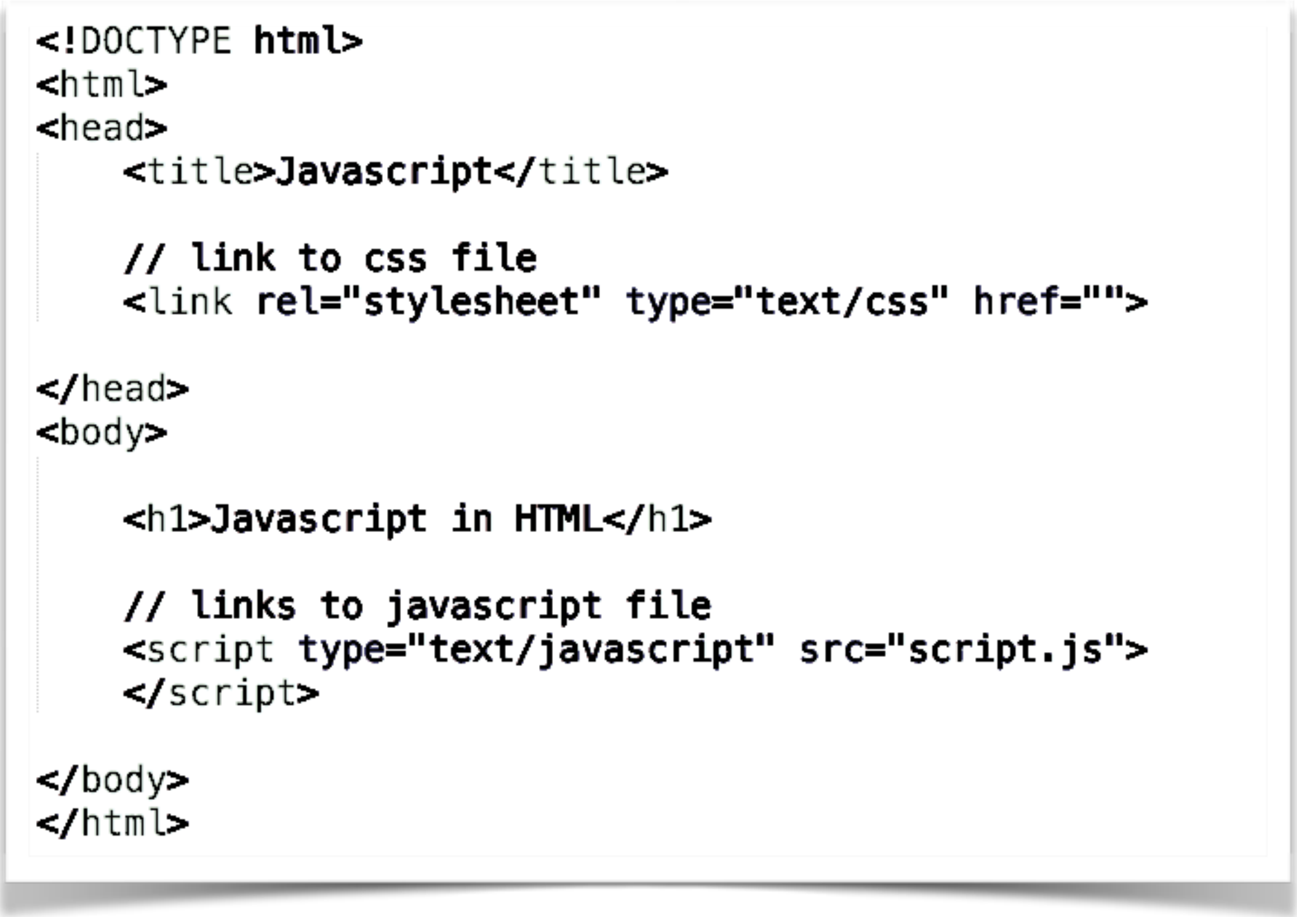
The Complete Web Developer in 2018

The Complete Web Developer in 2018
Zero to Mastery
Andrei Neagoie
Lecture Notes by Stephanie

Add Javascript to Webpage

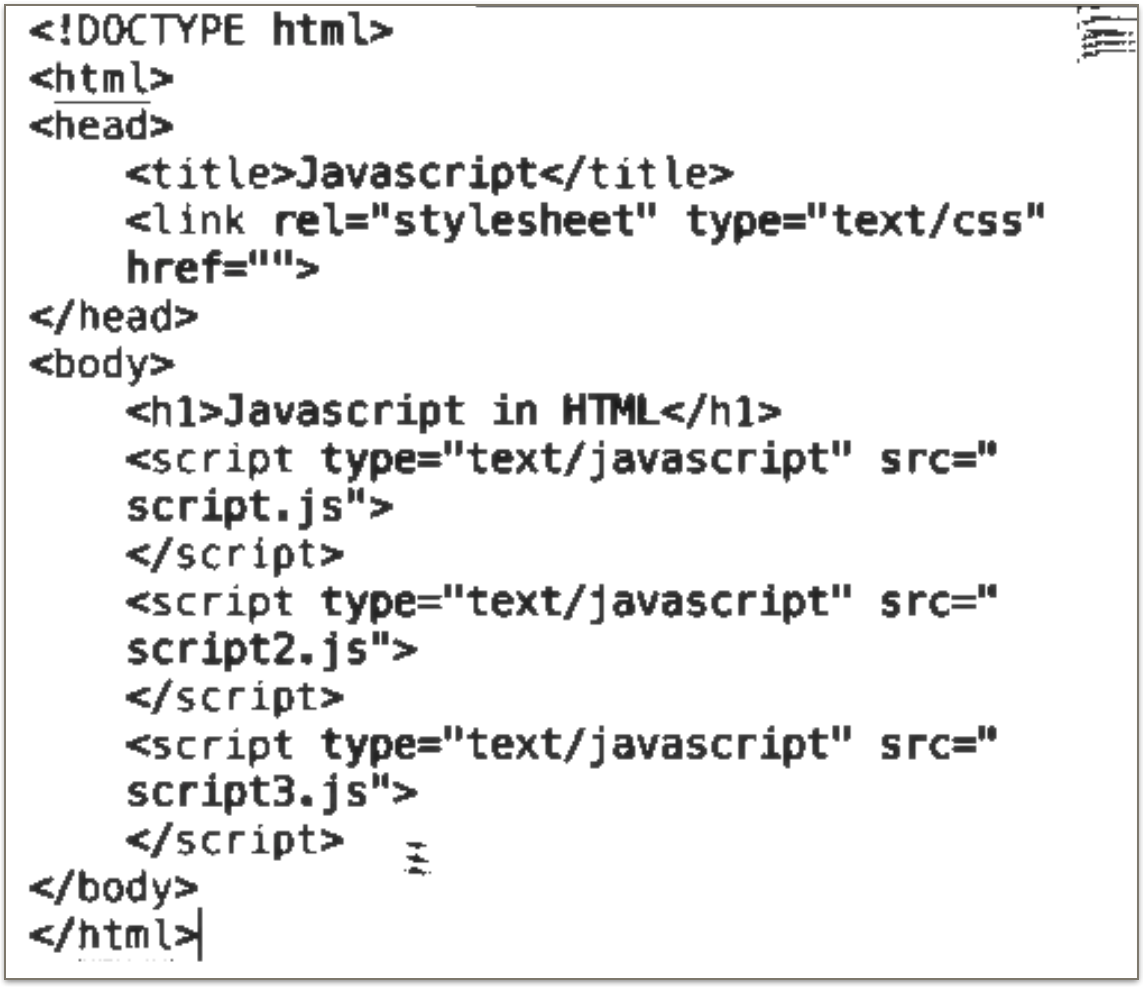


HTML file (index.html)



```
<!DOCTYPE html>
<html>
<head>
    <title>Javascript</title>

    // link to css file
    <link rel="stylesheet" type="text/css" href="">

</head>
<body>

    <h1>Javascript in HTML</h1>

    // links to javascript file
    <script type="text/javascript" src="script.js">
    </script>

</body>
</html>
```

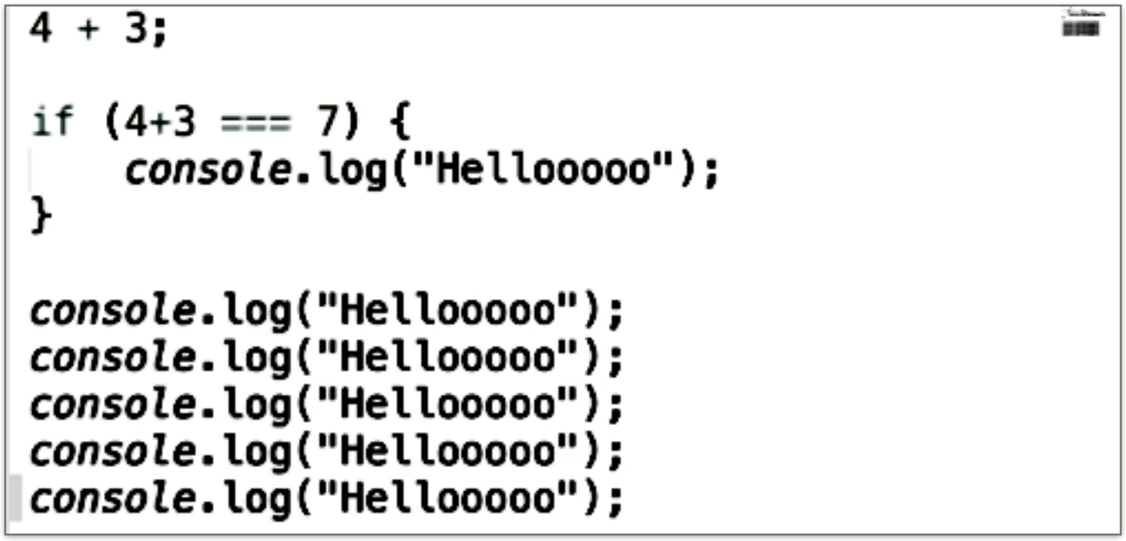
Can also have multiple javascript links as follows:



```
<!DOCTYPE html>
<html>
<head>
    <title>Javascript</title>
    <link rel="stylesheet" type="text/css"
    href="">
</head>
<body>
    <h1>Javascript in HTML</h1>
    <script type="text/javascript" src="
    script.js">
    </script>
    <script type="text/javascript" src="
    script2.js">
    </script>
    <script type="text/javascript" src="
    script3.js">
    </script>
</body>
</html>
```

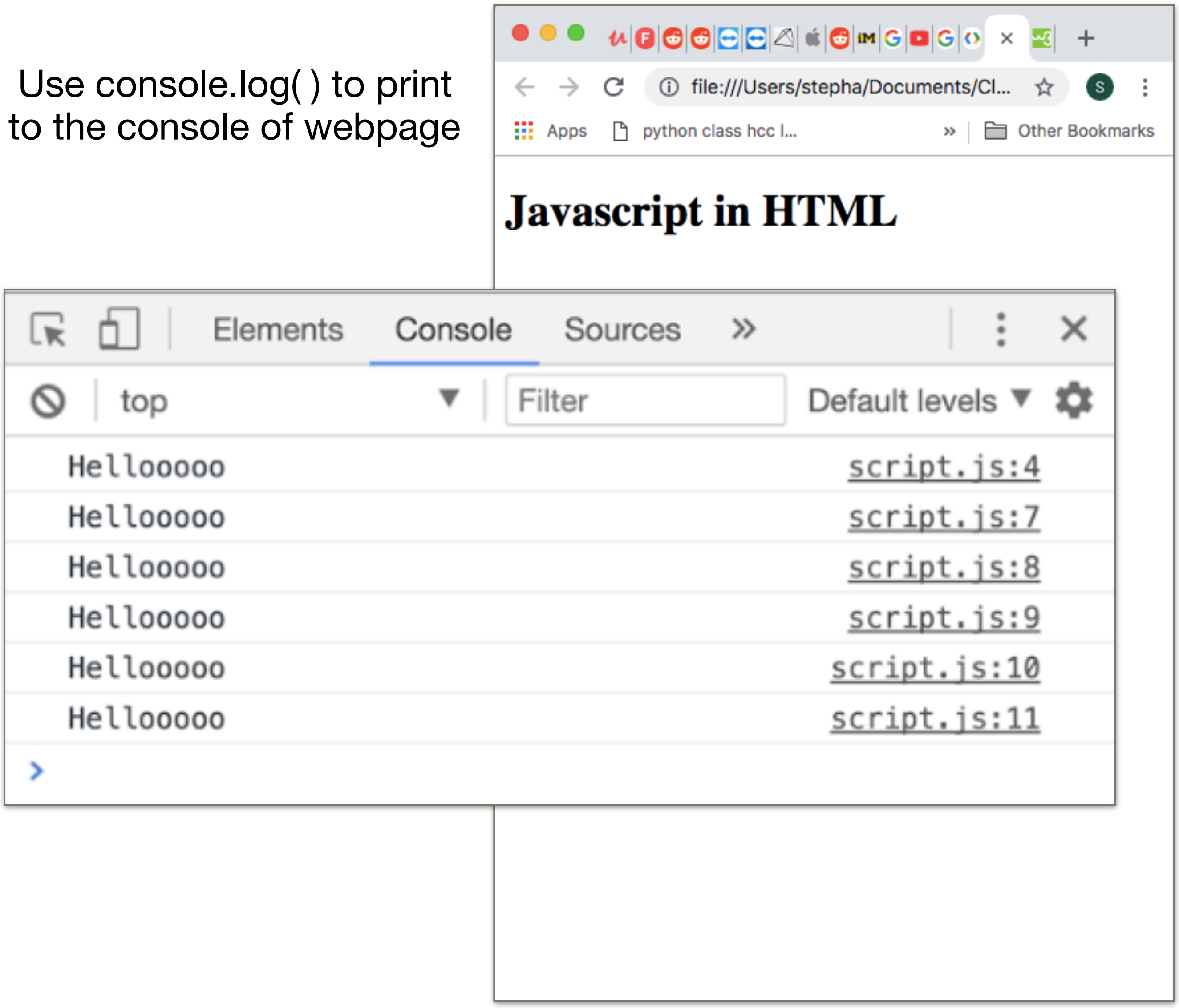
We put the javascript link in the end of <body> so that the page renders before pulling the javascript file

Javascript file (script.js)



```
4 + 3;

if (4+3 === 7) {
    console.log("Hellooooo");
}

console.log("Hellooooo");
console.log("Hellooooo");
console.log("Hellooooo");
console.log("Hellooooo");
console.log("Hellooooo");
```

Use console.log( ) to print to the console of webpage



Javascript in HTML

```
Hellooooo                                  script.js:4
Hellooooo                                  script.js:7
Hellooooo                                  script.js:8
Hellooooo                                  script.js:9
Hellooooo                                  script.js:10
Hellooooo                                  script.js:11
>
```

Javascript Functions

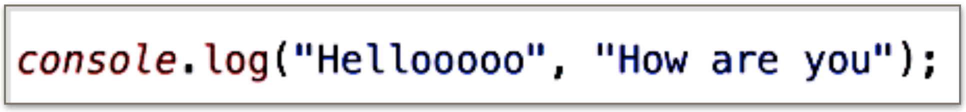
examples:

alert( )
prompt( )
console.log( )

These functions come with javascript

Use parentheses ( ) to call the function
Arguments given to funxn in parentheses

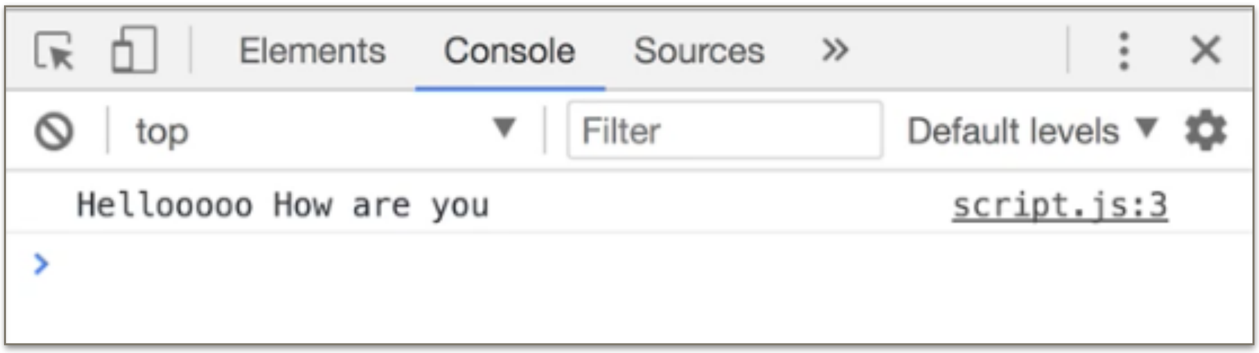
Javascript file (script.js)



```
console.log("Hellooooo", "How are you");
```

1st argument 2nd argument

Console when webpage loads

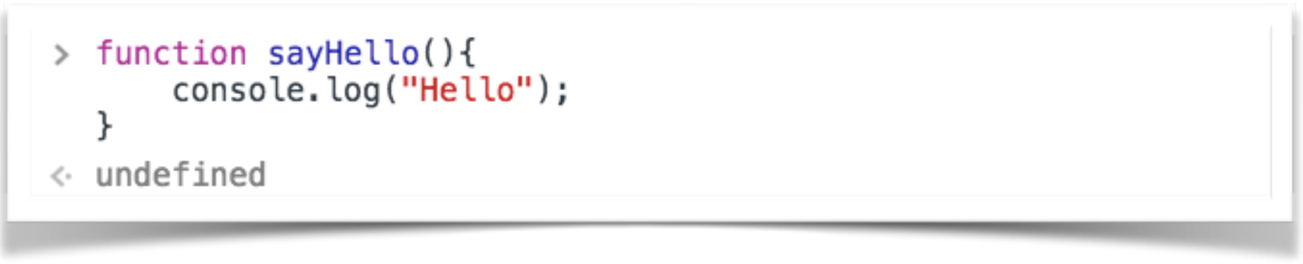


```
Hellooooo How are you                         script.js:3
>
```

Javascript Functions

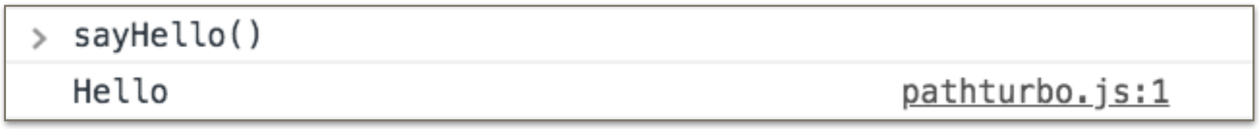
function name( ) { }
var a = function name( ) { }
return
( ) => (new in ECMAScript 6)

Function Declaration: function name( ) { }



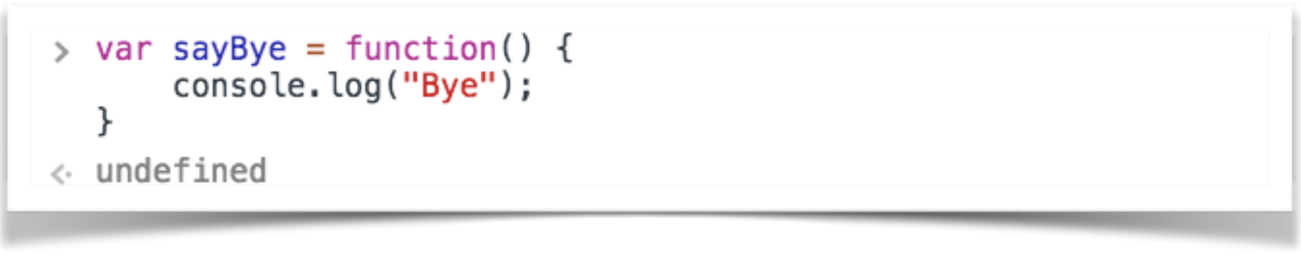
```
> function sayHello(){
      console.log("Hello");
  }
<· undefined
```

Call the function



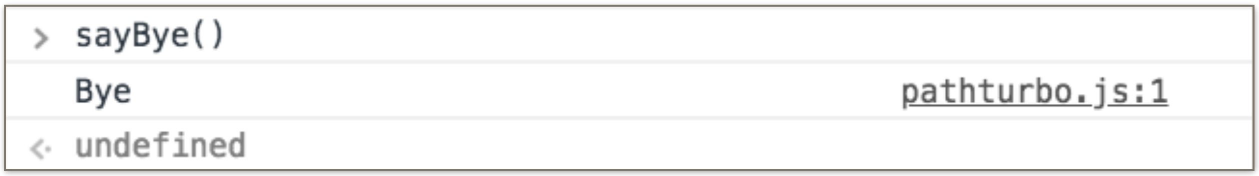
```
> sayHello()
  Hello                                          pathturbo.js:1
```

Function Expression: var a = function name( ) { }
"anonymous function": funxn assigned to var but has no name



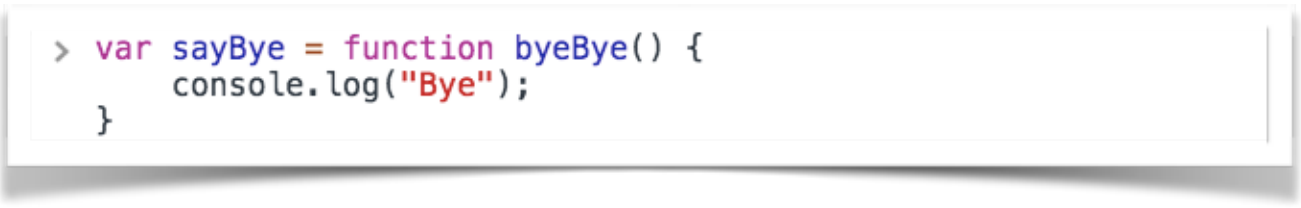
```
> var sayBye = function() {
      console.log("Bye");
  }
<· undefined
```

Call the function

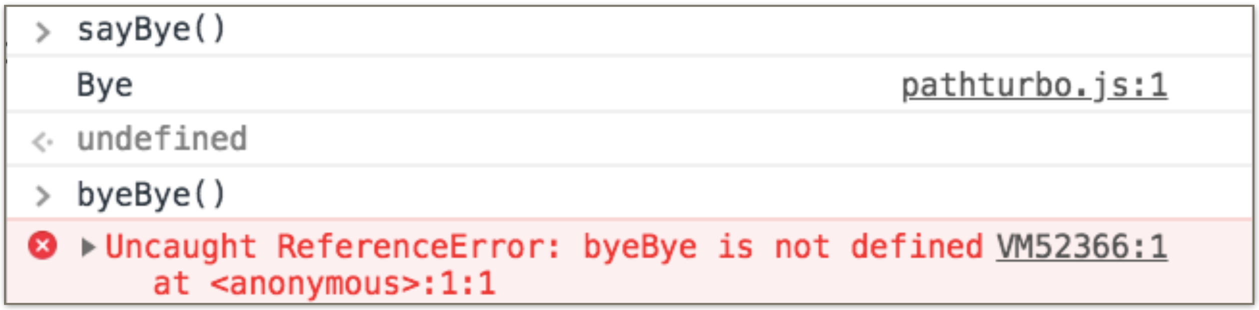


```
> sayBye()
  Bye                                              pathturbo.js:1
<· undefined
```

Can name the function byeBye( ), but it has limited use
Call the function



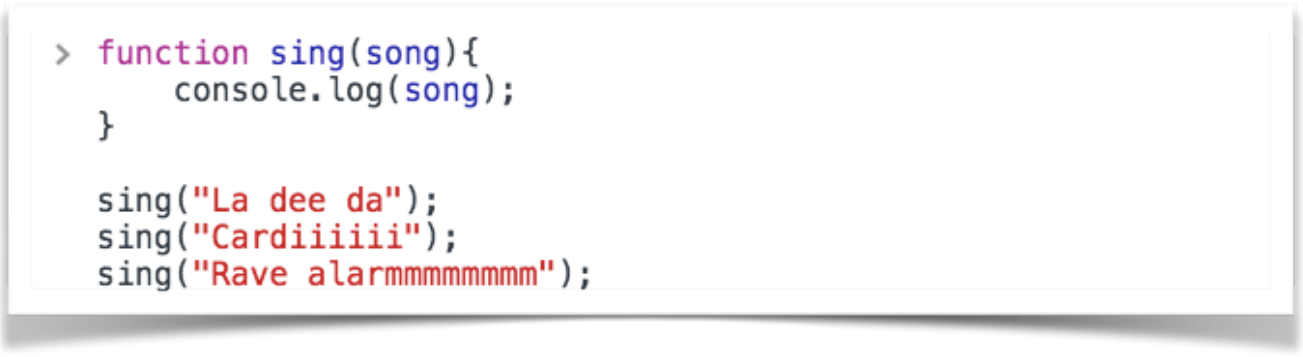
```
> var sayBye = function byeBye() {
      console.log("Bye");
  }
```



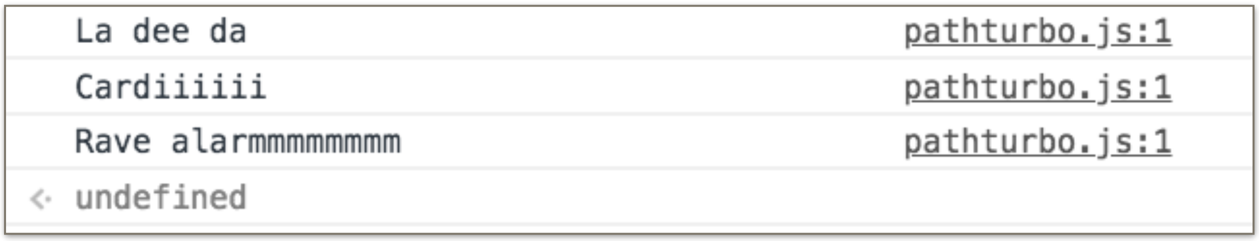
```
> sayBye()
  Bye                                              pathturbo.js:1
<· undefined
> byeBye()
✖ ▶Uncaught ReferenceError: byeBye is not defined VM52366:1
      at <anonymous>:1:1
```

DRY - “Don’t Repeat Yourself”

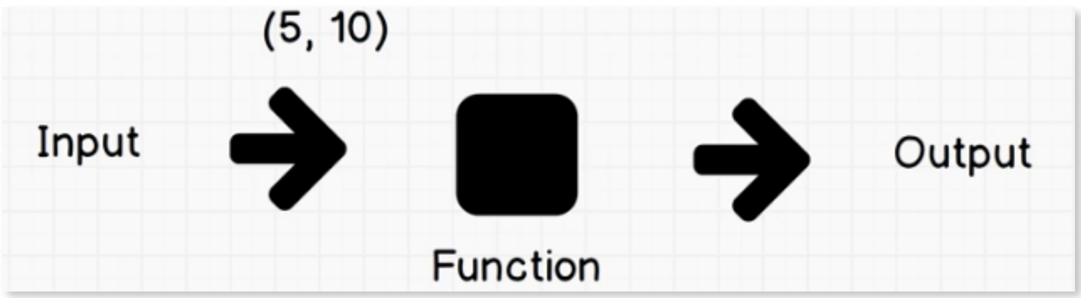
Use arguments to make functions more “extensible”



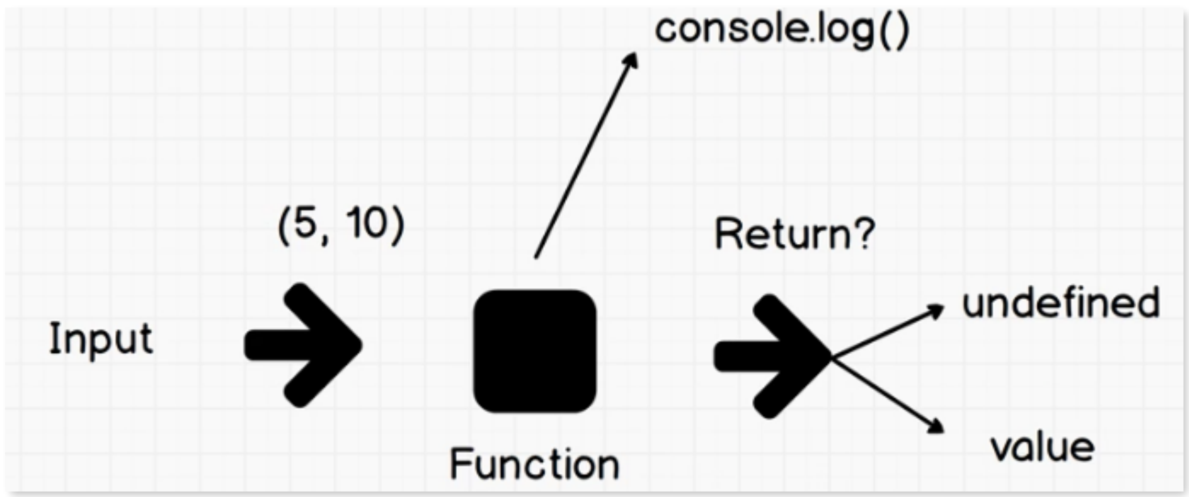
```
> function sing(song){
      console.log(song);
  }

  sing("La dee da");
  sing("Cardiiiiii");
  sing("Rave alarmmmmmmmm");
```



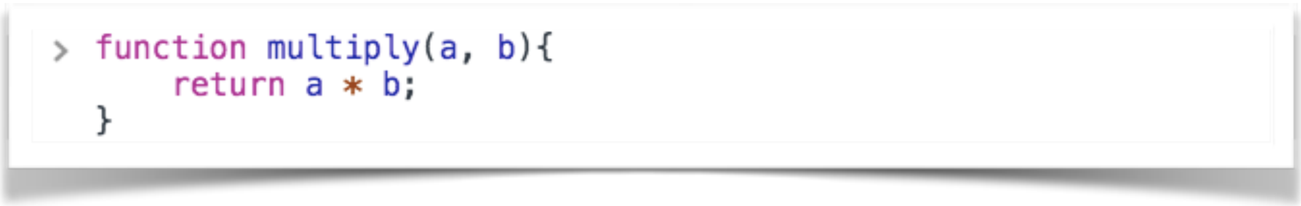
```
  La dee da                                  pathturbo.js:1
  Cardiiiiii                                 pathturbo.js:1
  Rave alarmmmmmmmm                          pathturbo.js:1
<· undefined
```



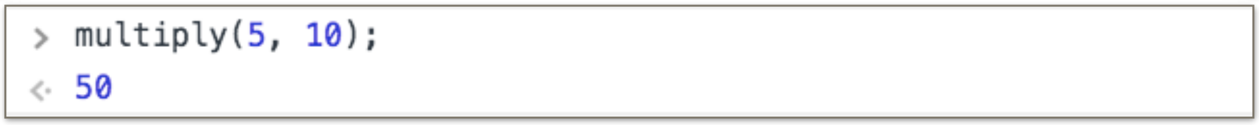
Function can create console log, return a value, or not return anything (undefined)



Use **return** to have function return value

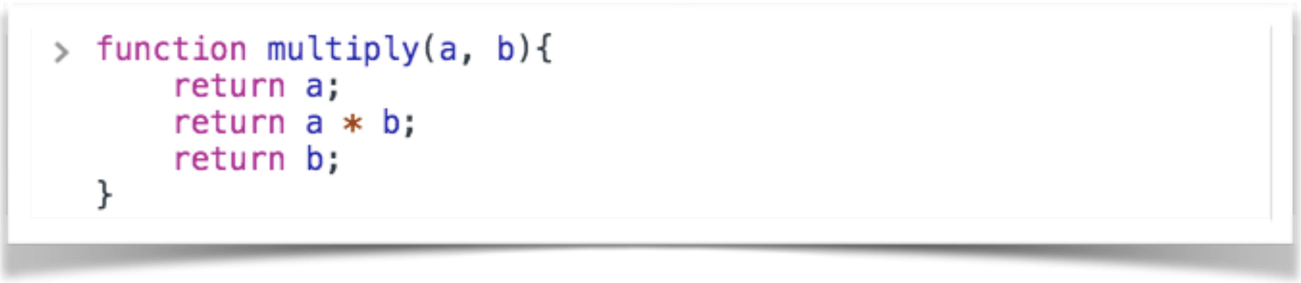


```
> function multiply(a, b){
      return a * b;
  }
```

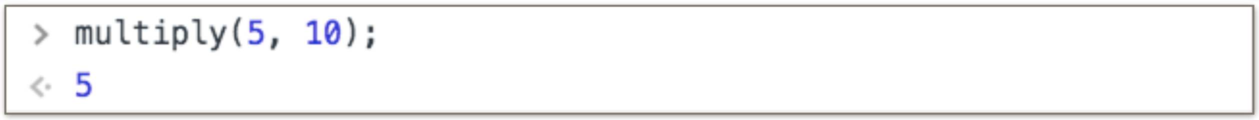


```
> multiply(5, 10);
<· 50
```

Functions always exit after first return.
Returns a and then exits. Other two lines of code not read at all.

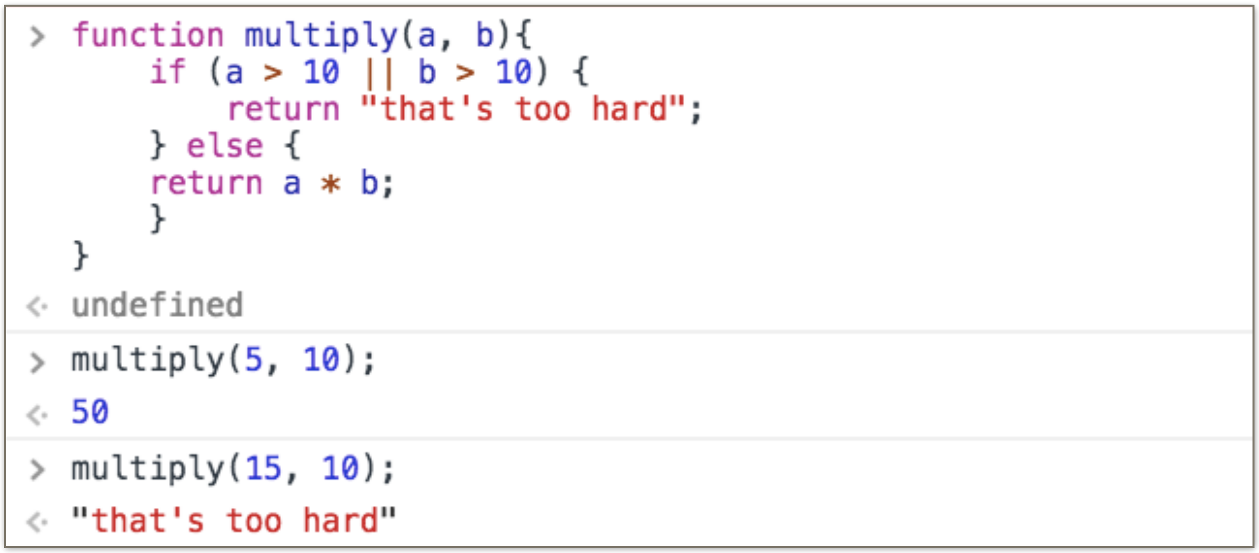


```
> function multiply(a, b){
      return a;
      return a * b;
      return b;
  }
```

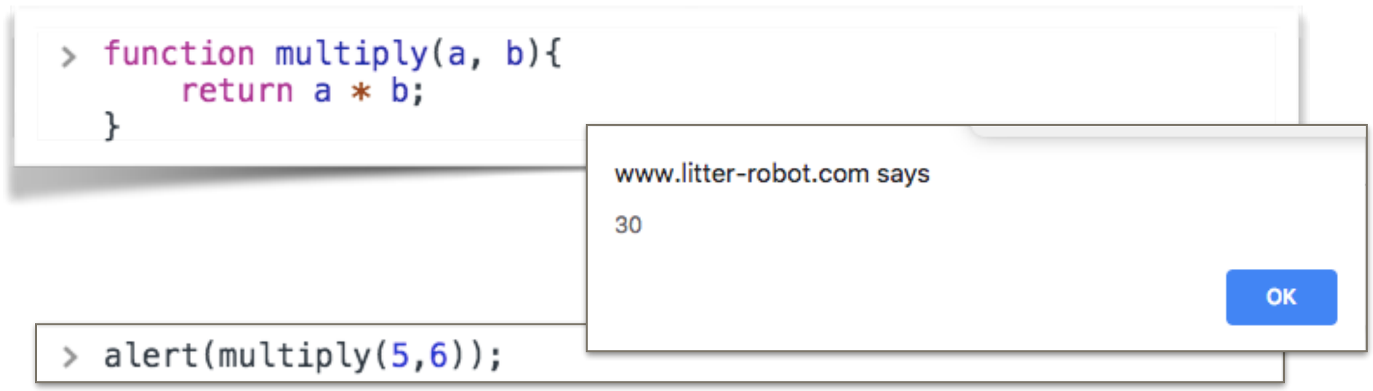


```
> multiply(5, 10);
<· 5
```

Use multiple returns in situations like:



```
> function multiply(a, b){
      if (a > 10 || b > 10) {
          return "that's too hard";
      } else {
      return a * b;
      }
  }
<· undefined
> multiply(5, 10);
<· 50
> multiply(15, 10);
<· "that's too hard"
```



```
> function multiply(a, b){
      return a * b;
  }
```

www.litter-robot.com says

30

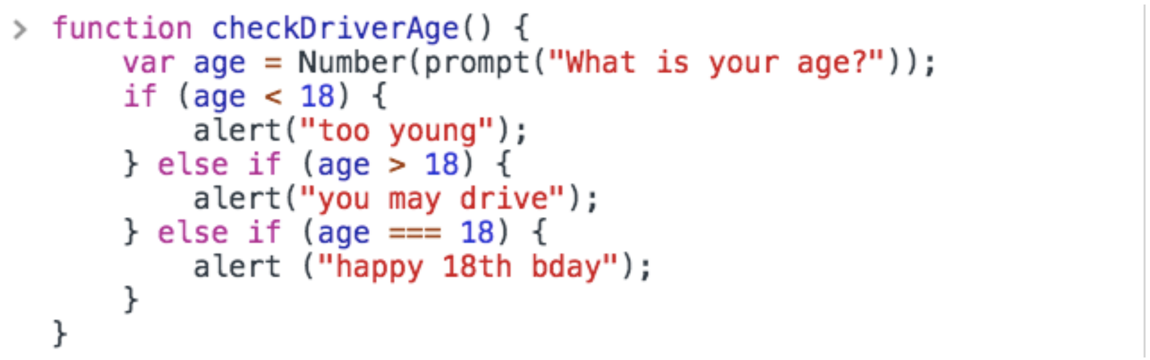
```
> alert(multiply(5,6));
```

Inner functions - nest functions, alert(multiply( ))

Parameters vs arguments:
Parameters - a, b
Arguments - 5, 6

```
// Exercise 5
// Make a keyless car EVEN BETTER!
// We are improving our car from previous exercise now.

var age = prompt("What is your age?");

if (Number(age) < 18) {
	alert("Sorry, you are too young to drive this car. Powering off");
} else if (Number(age) > 18) {
	alert("Powering On. Enjoy the ride!");
} else if (Number(age) === 18) {
	alert("Congratulations on your first year of driving. Enjoy the ride!");
}
```

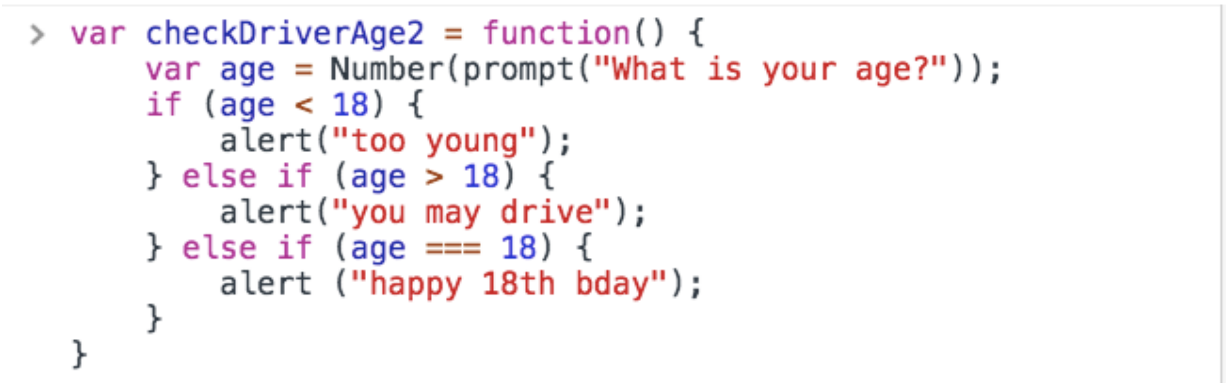
//1. Make the above code have a function called checkDriverAge(). Whenever you call this function, you will get prompted for age. Use **Function Declaration** to create this function.



```
> function checkDriverAge() {
      var age = Number(prompt("What is your age?"));
      if (age < 18) {
          alert("too young");
      } else if (age > 18) {
          alert("you may drive");
      } else if (age === 18) {
          alert ("happy 18th bday");
      }
  }
```

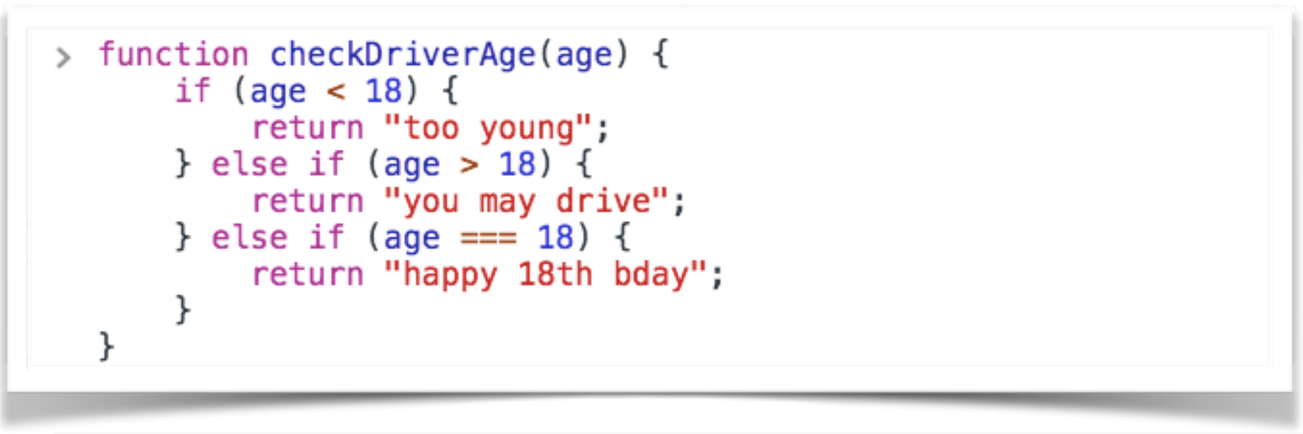
Notice the benefit in having checkDriverAge() instead of copying and pasting the function everytime?

//2. Create another function that does the same thing, called checkDriverAge2() using Function Expression.

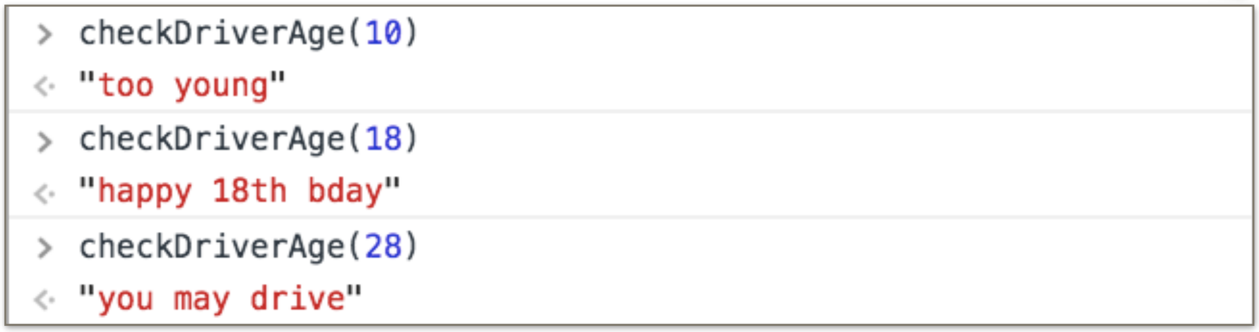


```
> var checkDriverAge2 = function() {
      var age = Number(prompt("What is your age?"));
      if (age < 18) {
          alert("too young");
      } else if (age > 18) {
          alert("you may drive");
      } else if (age === 18) {
          alert ("happy 18th bday");
      }
  }
```

//BONUS: Instead of using the prompt. Now, only use the return keyword and make the checkDriverAge() function accept an argument of age, so that if you enter:
checkDriverAge(92);
it returns "Powering On. Enjoy the ride!"



```
> function checkDriverAge(age) {
      if (age < 18) {
          return "too young";
      } else if (age > 18) {
          return "you may drive";
      } else if (age === 18) {
          return "happy 18th bday";
      }
  }
```



```
> checkDriverAge(10)
<· "too young"
> checkDriverAge(18)
<· "happy 18th bday"
> checkDriverAge(28)
<· "you may drive"
```