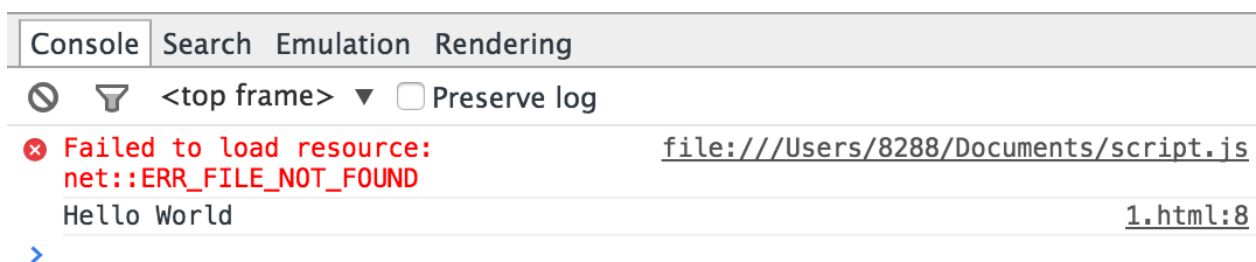
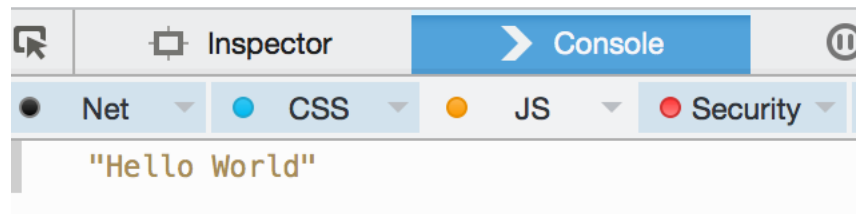
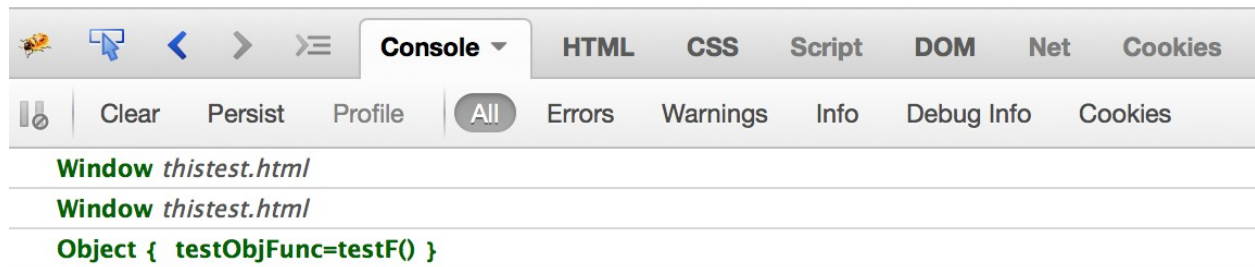


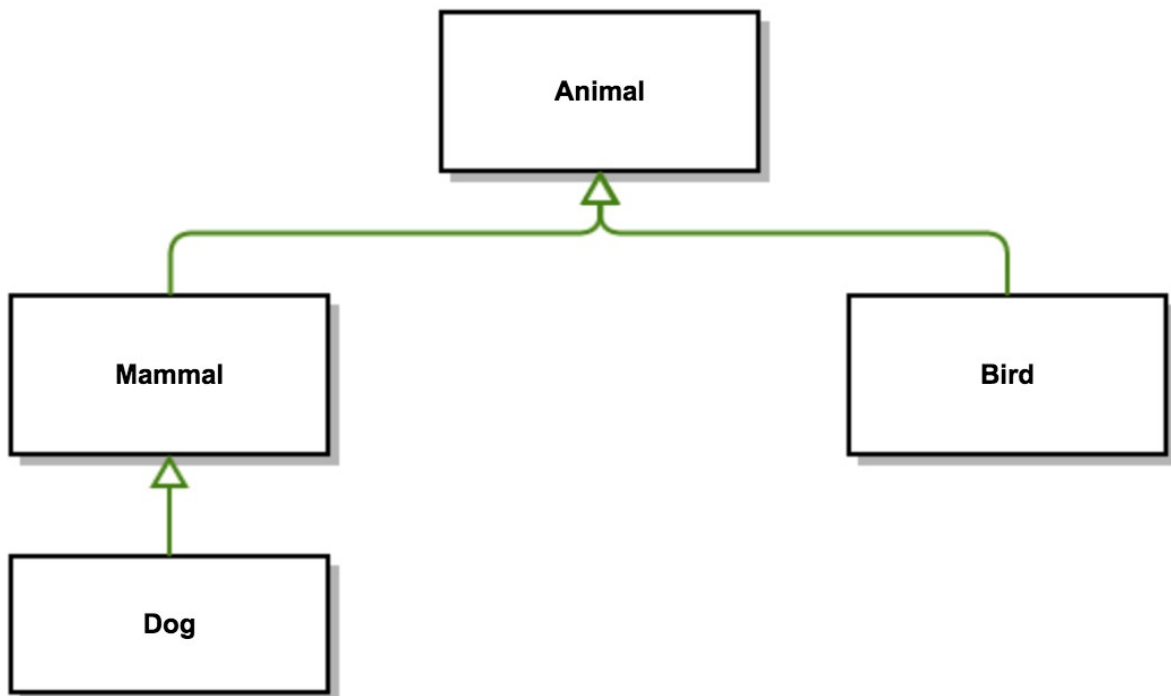
Chapter 1: JavaScript Primer

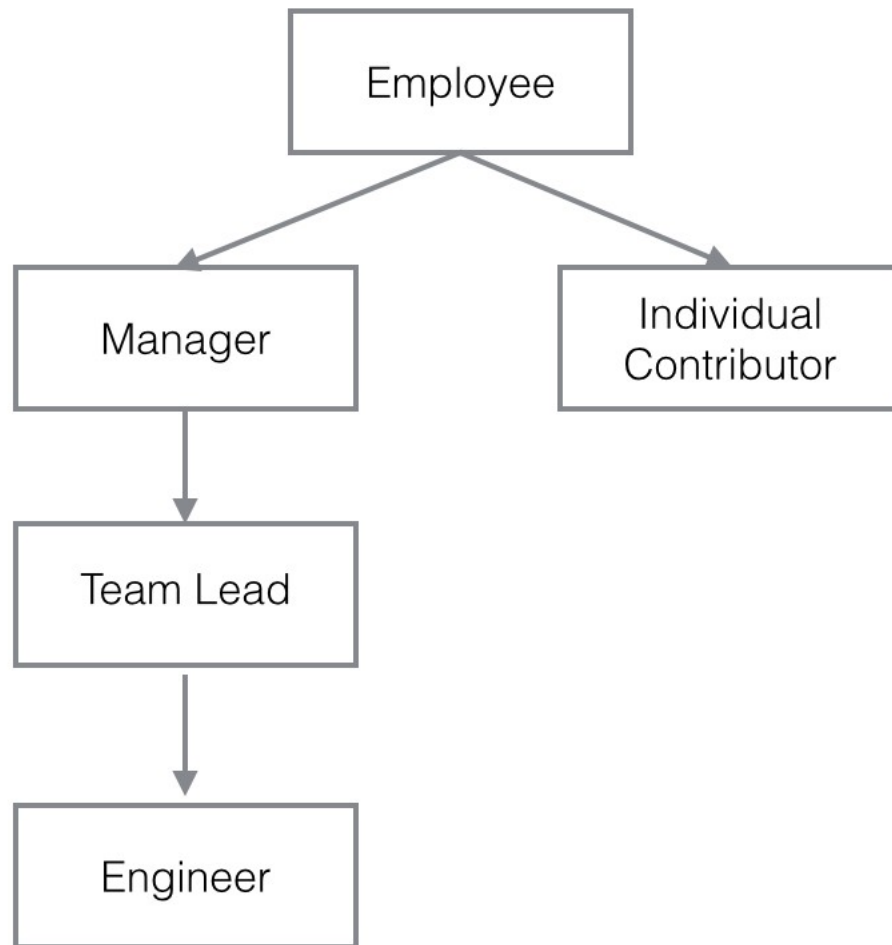


Chapter 2: Functions, Closures, and Modules

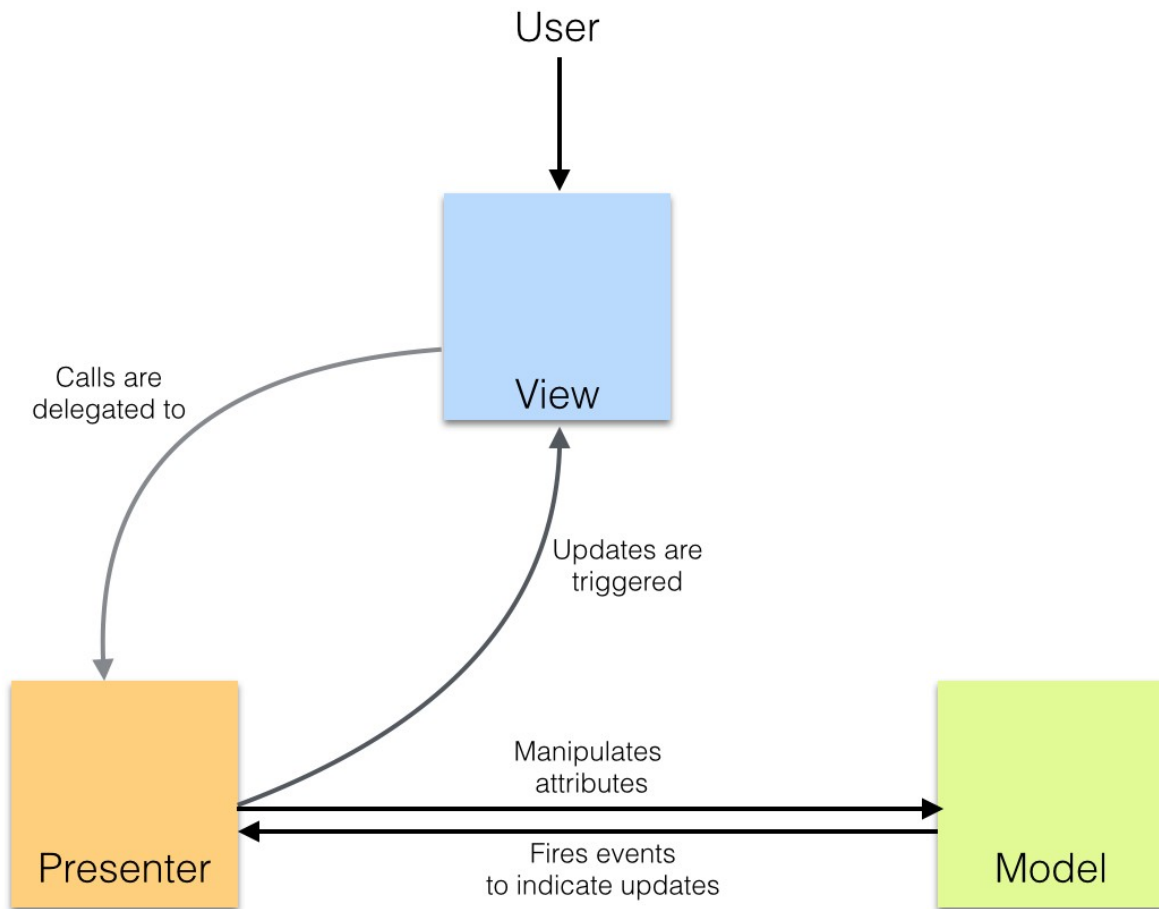


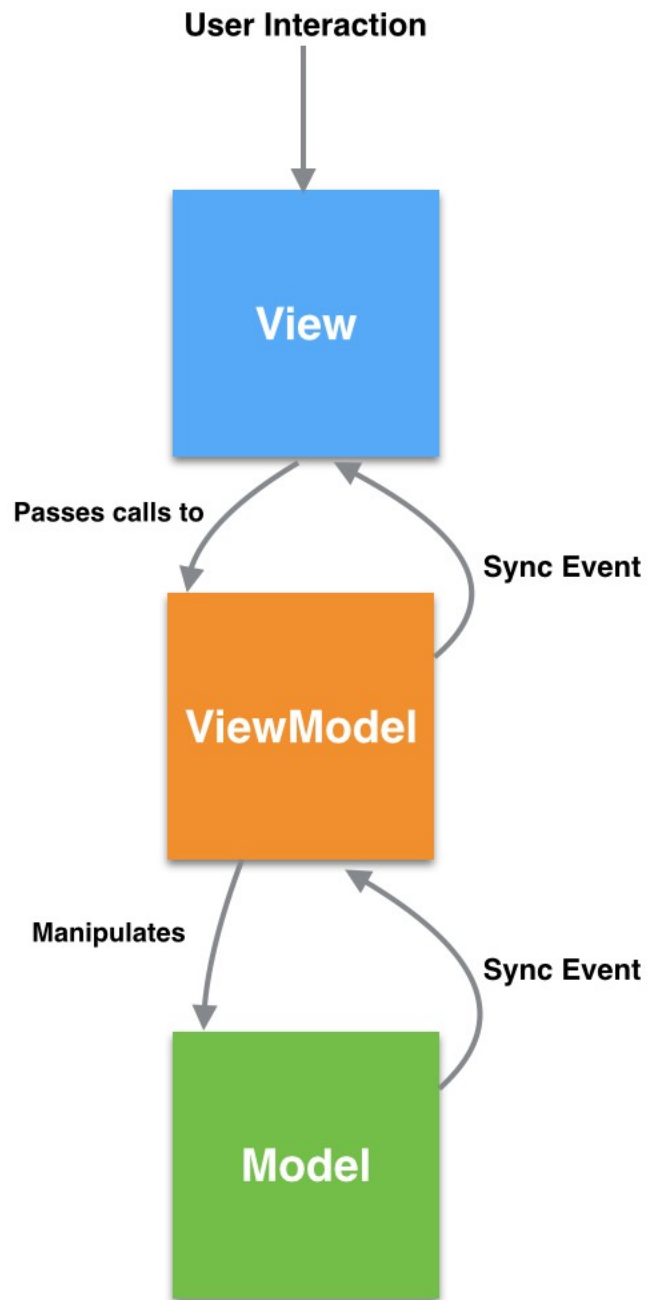
Chapter 4: Object-Oriented JavaScript



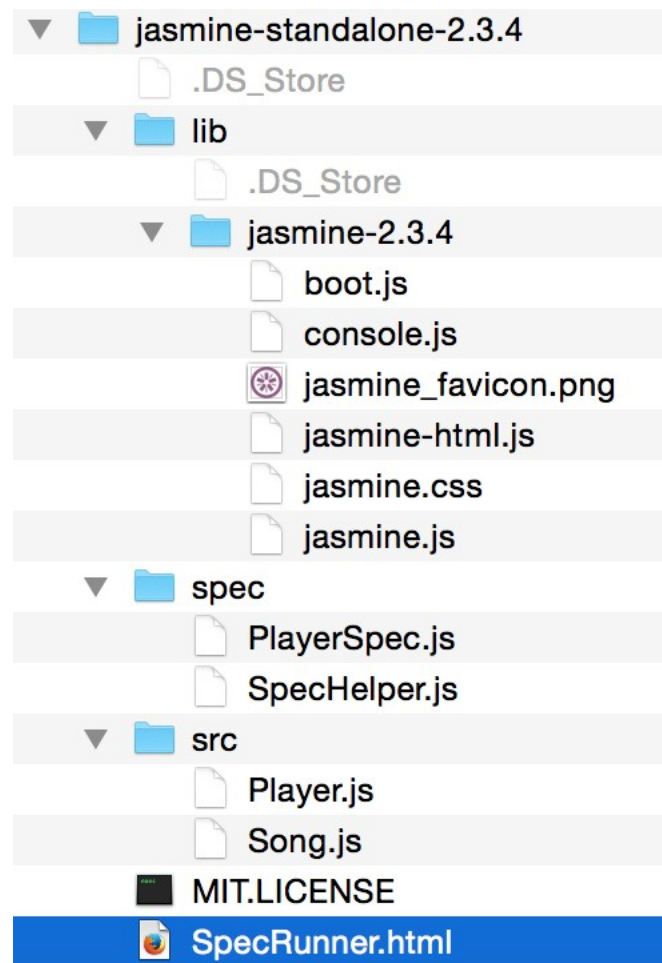


Chapter 5: JavaScript Patterns





Chapter 6: Testing and Debugging



 Jasmine 2.3.4

● ✖

2 specs, 1 failure

Spec List | Failures

TestStringUtilities can handle undefined

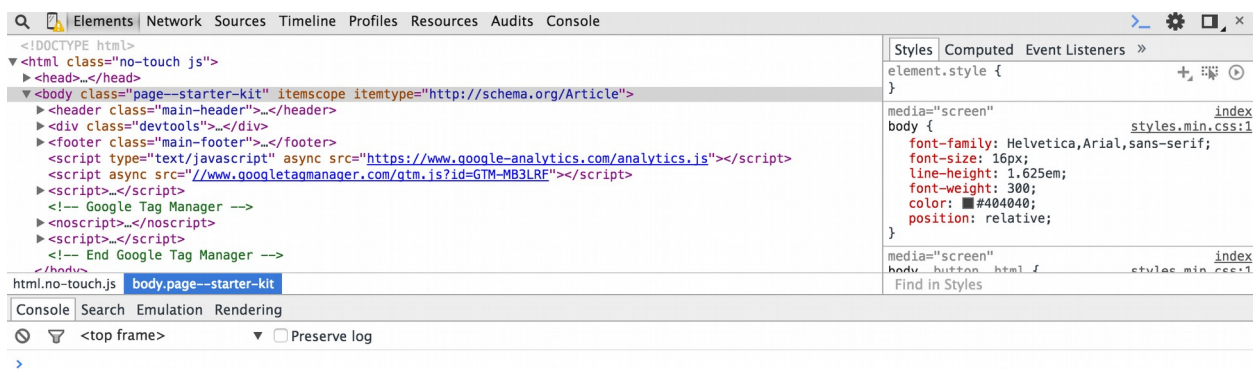
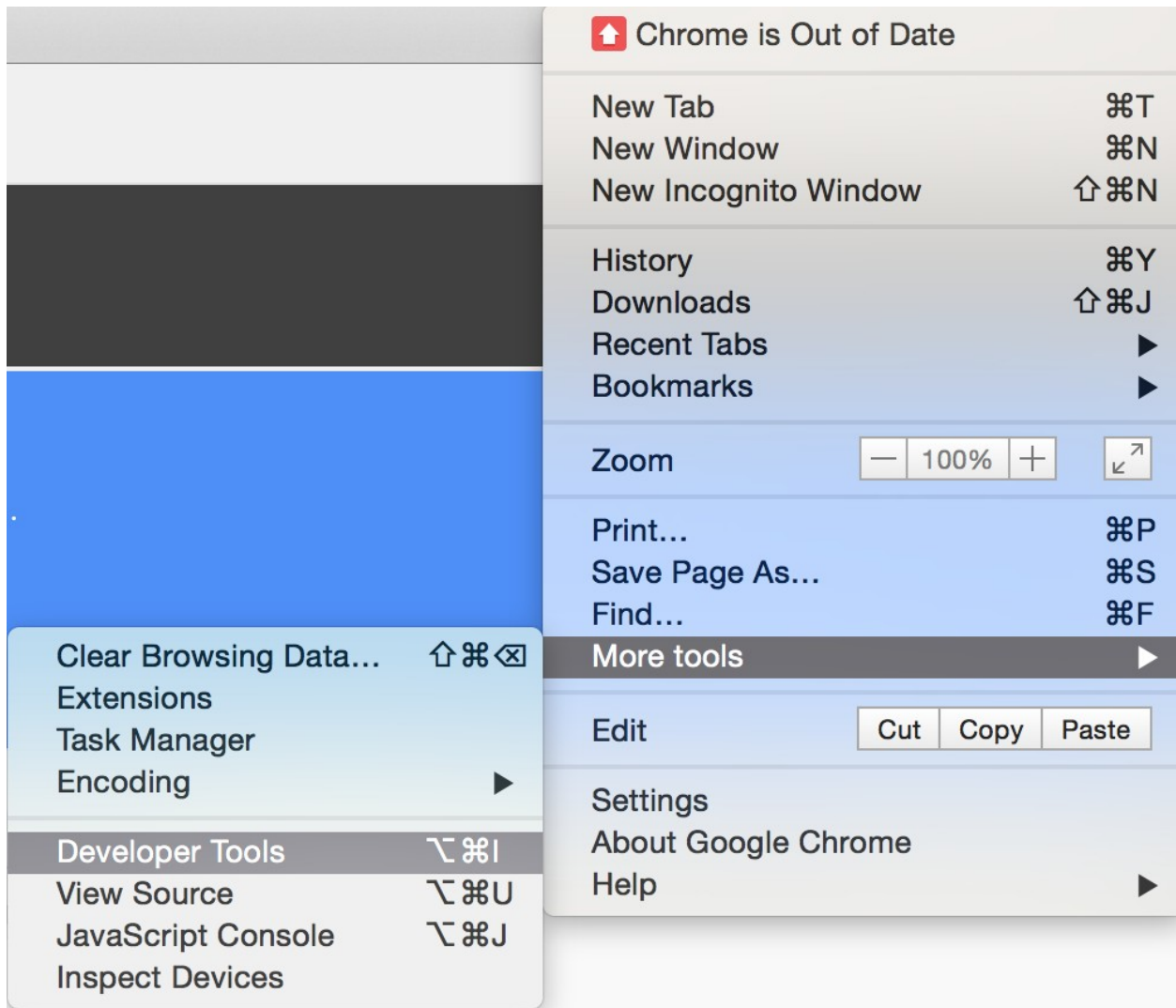
TypeError: name is undefined in file:///Users/8288/Downloads/jasmine-standalone-2.3.4/src/bigfatjavascriptcode.js (line 2)

 Jasmine 2.3.4

● ●

2 specs, 0 failures

TestStringUtilities
converts to capital
can handle undefined



🔍📄

ElementsNetworkSourcesTimelineProfilesResourcesAuditsConsole

SourcesContent scriptsSnippets

thistest.html ×

▼🌐 file://
▼📁 Users/8288/Documents
📄 thistest.html

3<head>
4<meta charset="utf-8">
5<title>This test</title>
6<script type="text/javascript">
7function engageGear(gear){
8 if(gear==="R"){ console.log ("Reversing");}
9 if(gear==="D"){ console.log ("Driving");}
10 if(gear==="N"){ console.log ("Neutral/Parking");}
11 throw new Error("Invalid Gear State");
12 }
13 try
14 {
15 engageGear("R"); //Reversing
16 engageGear("P"); //Invalid Gear State
17 }
18 catch(e){
19 console.log(e.message);
20 }
21 </script>

{ } Line 1, Column 1

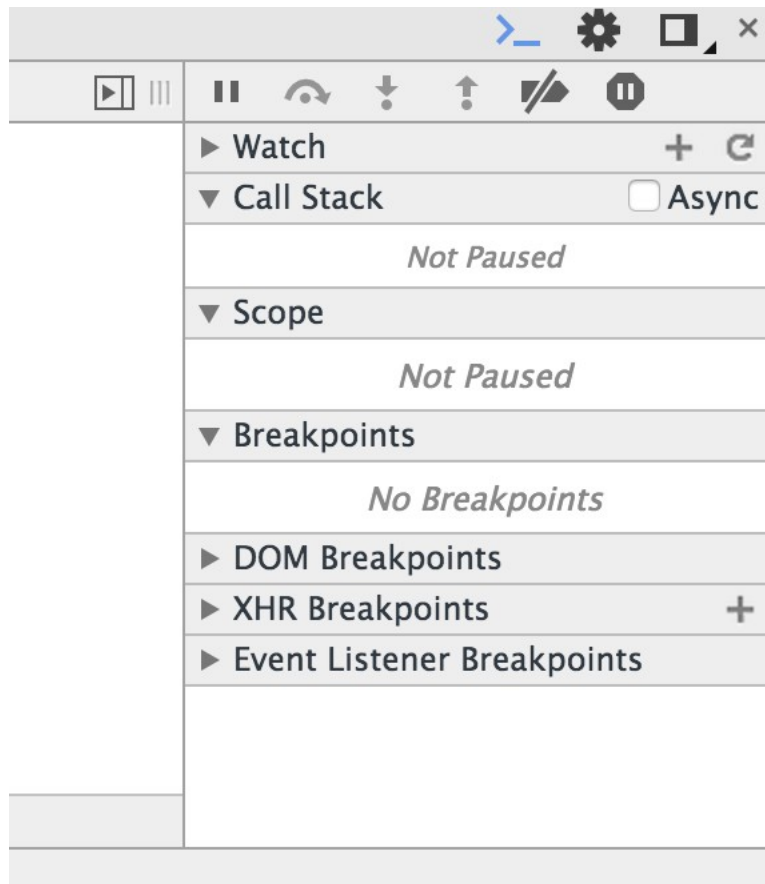
ConsoleSearchEmulationRendering

🔇🔍 <top frame> ▼ ☐ Preserve log

Reversing

Invalid Gear State

>



```
4 <meta charset="utf-8">
5 <title>This test</title>
6 <script type="text/javascript">
7   function engageGear(gear){
8     if(gear==="R"){ console.log ("Reversi")
9     if(gear==="D"){ console.log ("Driving")
10    if(gear==="N"){ console.log ("Neutral")
11    throw new Error("Invalid Gear State")
12  }
13  try
14  {
15    engageGear("R"); //Reversing
16    engageGear("P"); //Invalid Gear Stat
17  }
```

▼ Breakpoints

- ✓ thistest.html:8
if(gear=="R"){ console.log (...
- ✓ thistest.html:15
engageGear("R"); //Reversing

thistest.html x

1 <!DOCTYPE html>

2 <html>

3 <head>

4 <meta charset="utf-8">

5 <title>This test</title>

6 <script type="text/javascript">

7 function engageGear(gear){

8 if(gear=="R"){ console.log ("Reversing");

9 if(gear=="D"){ console.log ("Driving");}

10 if(gear=="N"){ console.log ("Neutral/Park

11 throw new Error("Invalid Gear State");

12 }

13 try

14 {

15 engageGear("R"); //Reversing

16 engageGear("P"); //Invalid Gear State

17 }

18 catch(e){

▶ Watch + ↻

▼ Call Stack ☐ Async

(anonymous function) thistest.html:15

Paused on a JavaScript breakpoint.

▼ Scope

▶ Global Window

▼ Breakpoints

✓ thistest.html:8
if(gear=="R"){ console.log (..

✓ thistest.html:15
engageGear("R"); //Reversing



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>This test</title>
6   <script type="text/javascript">
7     function engageGear(gear){ gear = "R"
8     if(gear==="R"){ console.log ("Reversing");
9     if(gear==="D"){ console.log ("Driving");}
10    if(gear==="N"){ console.log ("Neutral/Park");
11    throw new Error("Invalid Gear State");
12    }
13    try
14    {
15      engageGear("R"); //Reversing
16      engageGear("P"); //Invalid Gear State
17    }
18    catch(e){
19      console.log(e.message);
20    }
21  </script>
22 </head>
23 <body>
24 </body>
25 </html>

```

▼ Watch

+

↺

No Watch Expressions

▼ Call Stack

engageGear

thisistest.html:8

(anonymous function)

thisistest.html:15

Paused on a JavaScript breakpoint.

▼ Scope

▼ Local

gear: "R"

▶ this: Window

▶ Global

Window

▼ Breakpoints

✓

thisistest.html:8

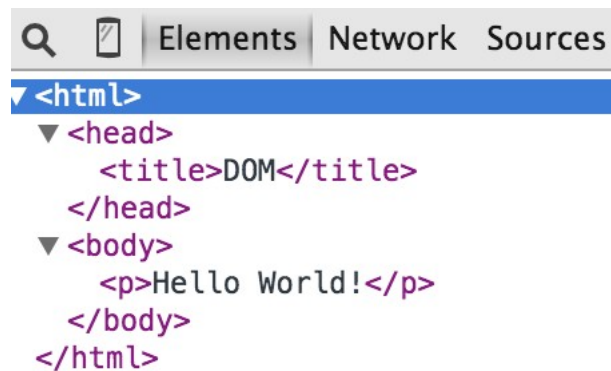
if(gear==="R"){ console.log (...

✓

thisistest.html:15

engageGear("R"); //Reversing

Chapter 8: DOM Manipulation and Events

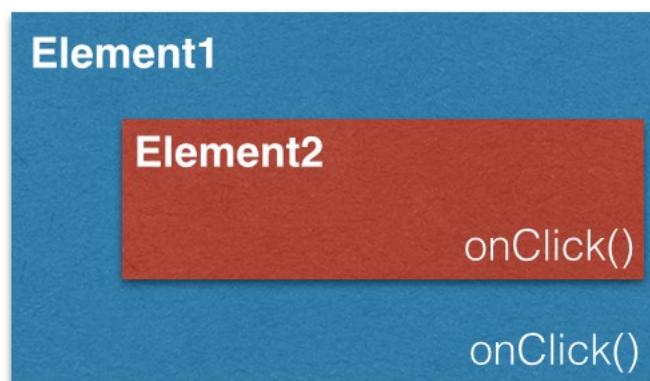


Are we there yet ? Are we there yet ?

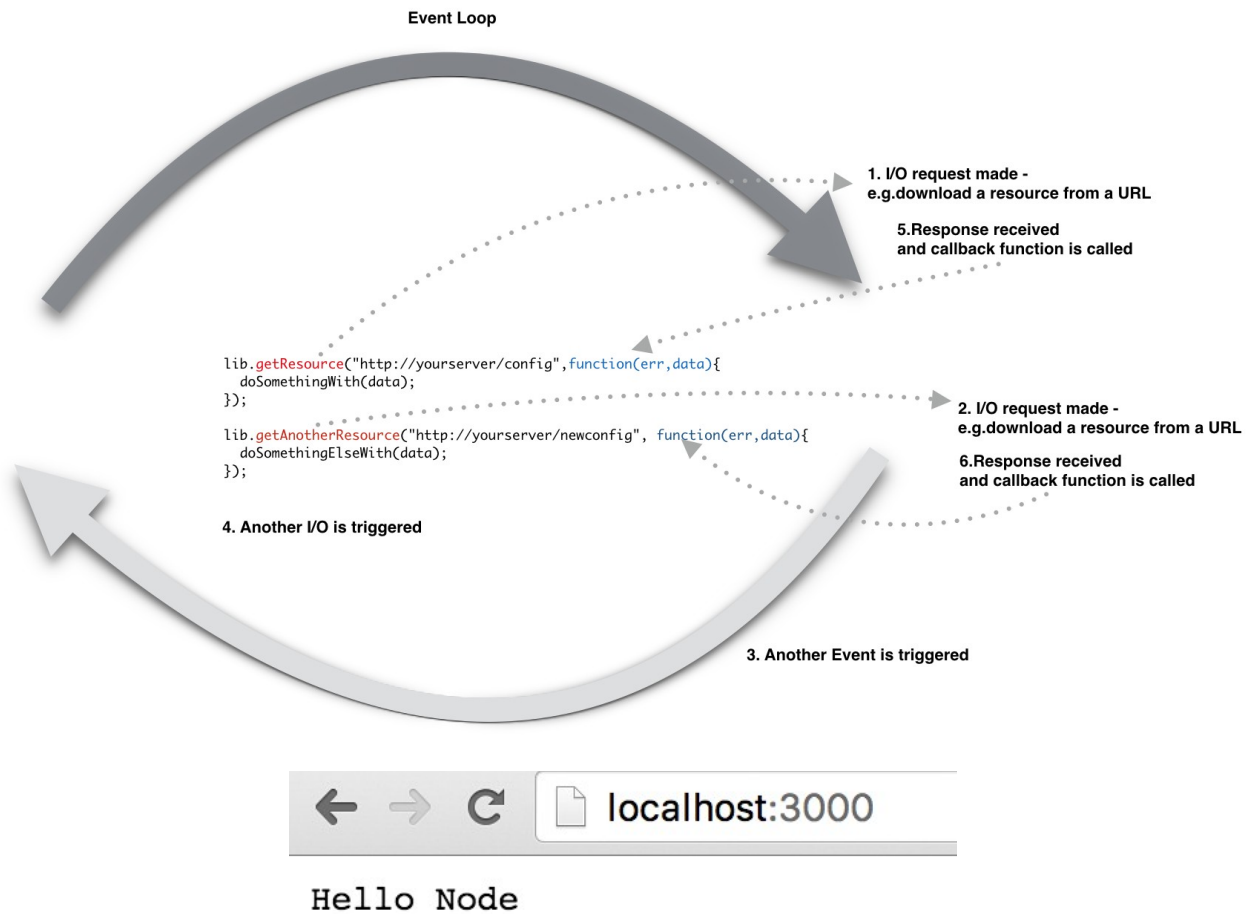
Journey to Mars

- First
- Second
- Third

Id	First name	Last Name
1	Albert	Einstein
2	Issac	Newton
3	Enrico	Fermi
4	Richard	Feynman



Chapter 9: Server-Side JavaScript



S

Work for the best startups

Q

📄

Elements

Network

Sources

Timeline

Profiles

Resources

Audits

Console

●

🔒

Profiles

Select profiling type

☒

Collect JavaScript CPU Profile

CPU profiles show where the execution time is spent

☐

Take Heap Snapshot

Heap snapshot profiles show memory distribution at the time the snapshot is taken

☐

Record JavaScript object allocations over time. Use the heap snapshot tool to view the heap at any point in time.

Start

Load

Q

📄

Elements

Network

Sources

Timeline

Profiles

Resources

Audits

Console

●

🔒

Heavy (Bottom Up) ▾ 🔍 ✕ ↺

Profiles

CPU PROFILES

📊

Profile 1

Save

	Self		Total		Function
6923.0 ms			6923.0 ms		(idle)
2381.5 ms	7.09%		2381.5 ms	7.09%	▼ lin_solve
2380.5 ms	7.09%		2380.5 ms	7.09%	▼ project
2380.5 ms	7.09%		2380.5 ms	7.09%	▼ vel_step
2380.5 ms	7.09%		2380.5 ms	7.09%	▼ FluidField.update
2380.5 ms	7.09%		2380.5 ms	7.09%	▼ runNavierStokes
2380.5 ms	7.09%		2380.5 ms	7.09%	▼ Measure
2380.5 ms	7.09%		2380.5 ms	7.09%	▼ BenchmarkSuite.RunSingleBenchmark
2380.5 ms	7.09%		2380.5 ms	7.09%	▶ ⚠ RunNextBenchmark
1.0 ms	0.00%		1.0 ms	0.00%	▶ diffuse
2245.9 ms	6.69%		2249.8 ms	6.70%	▶ montReduce
1915.1 ms	5.70%		1915.1 ms	5.70%	(garbage collector)
1068.3 ms	3.18%		1068.3 ms	3.18%	▶ bnpSquareTo
897.6 ms	2.67%		897.6 ms	2.67%	▶ GeneratePayloadTree

