Chapter 6: Javascript: Debugging

CS 80: Internet Programming

Instructor: Mark Edmonds

Developer Tools

- No program is perfect at first
- We need a way to examine program state as the program is running
 - This will enable us to understand and fix problems in our code
- Debugging will help fix any error, but it's particularly good for identifying edge cases you may have not considered when writing your program
 - Common edge cases: empty string "", zero 0, negative numbers, etc

Developer Tools

- The debugger is present in any modern browser and has two main elements for javascript: console and debugger
- Since we are editing the program as it executes, we need to remember where we are in execution
 - Variables may not exist yet
- How to write information to the console to inspect state:
 - Use a console.log() statement in your javascript
 - The value/text arguments will be printed to the Console section of the Developer Tools

Breakpoints

- Setting a breakpoint
 - Breakpoints say "when you hit this line of code, pause the program for me"
 - They allow you to inspect variable/program state during execution
 - Enable a breakpoint by clicking the line number
 - Important semantic note: The line of the breakpoint has **not** executed yet. It is about to execute
 - * E.g. if we set a breakpoint on line 9, line 9 hasn't executed when the breakpoint triggers (line 8 has, however)
 - * Keep this in mind!

Breakpoints

- Stepping through a program
 - Once we are at a breakpoint we have multiple options to control the program:
 - * Step Over: step over the current line of code. This means if we are at a function call, **do not** move the debugger into the function. Instead, the function will execute, and the program will pause after function completes
 - * Step Into: If the program is about to call a function, move the debugger into that function and pause execution
 - * Step Out: Finish the current function call and pause execution at the calling function
 - * Continue: Continue the program's execution; basically unpauses the program

Console Execution

- We can also call functions or inspect variables while paused using the console
 - Move to the console, and type a javascript statement to execute
 - * Can also just type variable names to get their value
 - If the javascript statement we called doesn't have a return value (e.g. statement doesn't yield a value), then the console will report undefined.

Example

- Each browser has a slightly different interface, but the developer tools work the same way for all
- Let's examine the developer tools within Chrome, but this will mostly work the same for every browser
- We'll use leap_year_checker.html for the example

Opening Developer Tools

- First, open the HTML file you wish to debug. Here, we'll open leap_year_checker.html in chrome
- Open the Developer Tools and go to the "Sources" tab:

Opening Developer Tools

- $ ightarrow$ X 💿 File /Users/mark/Dropbox/Developer/mark-rMBP-repos/personal/mjedmonds.github.io/CS80/examples/ch6_is/leap_year_checker.html	± €
This page says Input a year: Cancel OK	
The Elements Console Sources Network Performance Memory Application Security Audits	
Age Filesystem » :	
	► Watch
쁐 P Open file	▼ Call Stack
策 企 P Run command	
Dron in a folder to add to workspace	▼ Scope
<u>Learn more</u>	
	▶ XHR/fetch Breakpoints
	DOM Breakpoints
	Global Listeners
	Event Listener Breakp

Figure 1: Launch of dev tools

Opening a file

- Next, we'll press cmd+p (or ctrl+p) to open a file
- If your . html file does not appear, refresh the page and try again:

Opening a file

→ X ① File /Users/mark/Dropbox/Develope	/mark-rMBP-repos/personal/mjedmonds.github.io/CS80/examples/ch6_js/leap_year_checker.html This page says	± €
	Input a year:	
Elements Console Source	s Network Performance Memory Application Security Audits	:
age Filesystem » :		
top	<pre>leap_year_checker.html /Users/mark/Drop_s.github.io/CS80/examples/ch6_js/leap_year_checker.html</pre>	► Watch
v Users/mark/Dropbox/f leap_year_checker.t	.localized Downloads/.localized	Not paused
	baseline.png	▼ Scope
	Screen Shot 2019-10-19 at 10.53.15 PM.png	Not paused
	Descript Installer, app.zip	No breakpoints
	download_rips.sh	► XHR/fetch Breakpoint
		DOM Breakpoints
	rips.txt Downloads/Rivendell Rips/rips.txt	► Global Listeners
		Event Listener Break
	playlist_sync_genre.sh Downloads/Rivendell Rips/playlist_sync_genre.sh	

Figure 2: Loading HTML

Opening a file

- Now, we should see our file loaded
- You will need to enter into the prompt to see the file in the area below:

Opening a file



Figure 3: Loading HTML

Setting a breakpoint

• Next, we'll set a breakpoint on line 10 by clicking on the line (the green marker indicates a breakpoint):

Setting a breakpoint

🗕 🔍 💿 Leap Year Checker 🛛 🗙	+	
$m{\epsilon} \ ightarrow m{C} \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$	x/Developer/mark-rMBP-repos/personal/mjedmonds.github.io/CS80/examples/ch6_js/leap_year_checker.html	☆ 🖰 ፤
99 is not a leap year		
🕞 🚹 Elements Console	Sources Network Performance Memory Application Security Audits	: ×
Page Filesystem » :	leap_year_checker.html ×	П 🗠 🕇 🔶 🗌
▼ 🗖 top 2	<html></html>	▶ Watch
▼ file:// 3	<head></head>	▼ Call Stack
Users/mark/Dropbox/I 6	<pre><meta charset="utf-8"/> <title>Leap Year Checker</title></pre>	
leap_year_checker.h	<pre>leap_year_checker.h 7 <script type="text/javascript"></script></pre>	

Figure 4: Setting a breakpoint

Pausing at a breakpoint

- Refresh the page. Now the page should be paused, before the prompt appears
- Note that since we paused at line 10, line 10 is *about* to execute, but has not executed:

Pausing at a breakpoint

🔍 🔍 🔹 Leap Year Checker 🛛 🗙 -	+	
$igstarrow$ $igstarrow$ $igstarrow$ File \mid /Users/mark/Dropbox/D	Developer/mark-rMBP-repos/personal/mjedmonds.github.io/CS80/examples/ch6_js/leap_year_checker.html	☆ 🖰 🗄
🕞 🕞 Elements Console S	Sources Network Performance Memory Application Security Audits	: ×
Page Filesystem » : I	eap_year_checker.html ×	
▼ □ top 1 < 2 <	:DOCTYPE html>	i Paused on breakpoint
▼ file:// 3	head	▶ Watch
▼ Users/mark/Dropbox/I 5	<meta charset="utf-8"/>	▼ Call Stack
leap_year_checker.h 6 7	<title>Leap Year Checker</title> <script type="text/javascript"></td><td>(anonymous)</td></tr><tr><td>8</td><td><pre>var year, leapYear = false, leapYearText = "":</pre></td><td></td></tr><tr><td>10</td><td><pre>pyear = parseInt(window. prompt("Input a year: ")); if (year % 100 0) {</pre></td><td>- 50000</td></tr><tr><td>12</td><td>if (year % 400 === 0) {</td><td>▼ Scope ▶ Global Windo</td></tr><tr><td>13</td><td>leapyear = true; }</td><td>Breakpoints</td></tr><tr><td>15 16 17 18</td><td><pre>} else if (year % 4 === 0) { leapYear = true; } if (lleapYear) {</pre></td><td><pre>leap_year_checker.h year = parseInt(w</pre></td></tr><tr><td>19</td><td><pre>leapYearText = " not";</pre></td><td>XHR/fetch Breakpoints</td></tr><tr><td>20 21</td><td><pre>document.writeln("<h2>" + year + " is" + leapYearText + " a leap year</h2>")</pre></td><td>DOM Breakpoints</td></tr><tr><td>22 23 <,</td><td></script> /head>	Global Listeners
24		▶ Event Listener Breakpoin
	Line 10, Column 5	

Figure 5: Pausing at a breakpoint

Stepping over

- Next, hit the Step Over icon (above the yellow "Debugger paused"; arrow hopping over a dot)
- The prompt will appear. Enter 99 and hit OK. We should now be paused at the next line:

Stepping over

● ● ● ⊃ Leap Year Checker × + ← → X ③ File /Users/mark/Dropbox/Developer/mark-rMBP-repos/personal/mjedmonds.github.io/CS80/examples/ch6_js/leap_year_checker.html	☆ 9 :
🕞 🔂 Elements Console Sources Network Performance Memory Application Security Audits	: ×
Page Filesystem » : I leap_year_checker.html ×	
▼ □ top 1 html	i Debugger paused
▼ file:// 3	▶ Watch
	▼ Call Stack
leap_year_checker.r 6 <title>Leap Year Checker</title> 7 <script type="text/javascript"></td><td>(anonymous)</td></tr><tr><td><pre>8 var year, leapYear = false, 9 leapYearText = "":</pre></td><td></td></tr><tr><td><pre>10</td><td>- 80000</td></tr><tr><td>$\begin{array}{c ccccccccccccccccccccccccccccccccccc$</td><td>► Global Window</td></tr><tr><td>13 teaprear = true; 14 }</td><td>Breakpoints</td></tr><tr><td>15 } else if (year % 4 === 0) { 16 leapYear = true;</td><td></td></tr><tr><td>17 } 18 if (!leapYear) {</td><td>year = parseInt(w</td></tr><tr><td>19 leapYearText = " not";</td><td>XHR/fetch Breakpoints</td></tr><tr><td><pre>20 21 document.writeln("<h2>" + year + " is" + leapYearText + " a leap year</h2>" </pre></td><td>) ► DOM Breakpoints</td></tr><tr><td>22 </script> 23	Global Listeners
24	▶ Event Listener Breakpoint
{} Line 11, Column 5	

Figure 6: Stepping over

Checking variables

- Click the arrow next to Global under Scope on the rightside panel.
- Scroll down until you see the variable "year"
- Notice this variable has the value 99, which is what we inputted into the prompt:
- We can check and verify variables as our program runs, so we can spot incorrect results as we execute

Checking variables

🔍 🔍 🔹 Leap Year Checker 🛛 🗙	+	
$igstarrow$ $igstarrow$ $igstarrow$ $igstarrow$ File \mid /Users/mark/Dropb	ox/Developer/mark-rMBP-repos/personal/mjedmonds.github.io/CS80/examples/ch6_js/leap_year_checker.html	☆ 🔒 :
🕞 🗍 Elements Console	Sources Network Performance Memory Application Security Audits	: ×
Page Filesystem > . ▼ top . . ▼ file:// . . ▼ Users/mark/Dropbox/I . . Ieap_year_checker.r . . 1 	<pre>leap_year_checker.html × </pre> <pre> leap_year_checker.html × </pre> <pre> leapCorrect = "" </pre> <pre> leapYear_checker </pre> <pre> ctitle>LeapYear = false, leapYear = false, leapYearText = ""; leapYearText = ""; leapYearText = "0 { leapYear = fue; } leapYear = true; } leapYearText = " not"; document.writeln("<h2>" + year + " is" + leapYearText + " a leap year</h2>") </pre>	<pre>> styleMedia: Styl > toolbar: BarProp > toolbar: BarProp > top: Window (par > visualViewport: > webkitCancelAnin > webkitCancelAnin > webkitRequestFil > webkitStorageInf > webkitStorageInf > webkitStorageInf > window: Window { year: 99 > _LD: {00MEditHan Infinity: Infini > AbortController: > AbortSignal: f A > Accelerometer: f > AnalyserNode: f ini</pre>
{}	Line 11, Column 5	▶ AnimationEffect: ▶ AnimationEvent:

Figure 7: Stepping over

Leap year example

- Continue to Step Over or press Continue (play button) until your program finishes.
- For this example, I would recommend inputting different types of leap years (2000, 1996, 1975, etc) to see how the control flow of your program changes with different inputs.

Additional resources

- Here are a couple of external introductions that may also be useful:
- Chrome introduction: https://developers.google.com/web/tools/chrome-devtools/javascript
- Firefox introduction: https://developer.mozilla.org/en-US/docs/Tools/Debugger