Debugging JavaScript and CSS Using Firebug

Harman Goei

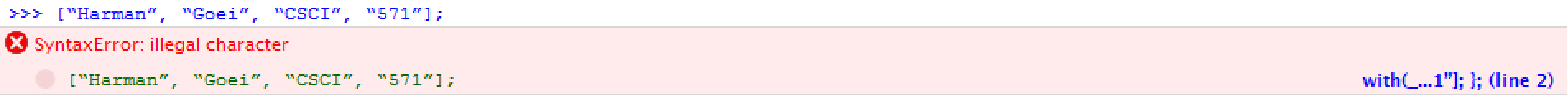
CSCI 571

1/27/13

USC Viterbi
School of Engineering

Notice for Copying JavaScript Code from these Slides

- When copying any JavaScript code from these slides, the console might return the following:



```
>>> [“Harman”, “Goei”, “CSCI”, “571”];
SyntaxError: illegal character
    [“Harman”, “Goei”, “CSCI”, “571”];                    with(_...1”]; }; (line 2)
```

- To fix this, **retype** each quote.

Outline

- **What is Firebug?** Why do web developers use Firebug?
- Installation of Firebug for **Mozilla Firefox** browser
- Launching Firebug for the First Time
- The Panels of Firebug
- Firebug Tutorials

What is Firebug?

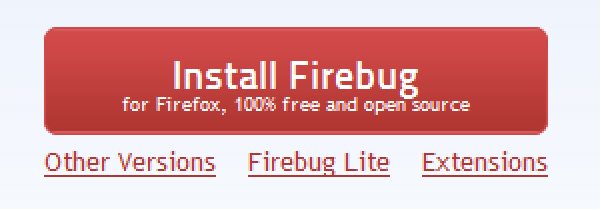
- Firebug is an extension for the Mozilla Firefox browser that allows you to debug and inspect HTML, CSS, the Document Object Model (DOM) and JavaScript.

Why do Web Developers use Firebug?

- Inspect the behavior of HTML/CSS, and modify style & layout with true WYSIWYG
- Debug JavaScript
- Detect performance of website
- Track Cookies & Sessions

Installation of Firebug for Mozilla Firefox Browser

- To install Firebug for Firefox, go to **http://www.getfirebug.com** (click on Install Firebug)



Install Firebug
for Firefox, 100% free and open source

Other Versions Firebug Lite Extensions

- *Don't have Firefox? **Firebug** has a **lite** version which can be saved as a bookmark or embedded into your web page in JavaScript.*

2 Ways to Launch Firebug

With the Mozilla Firefox browser open…

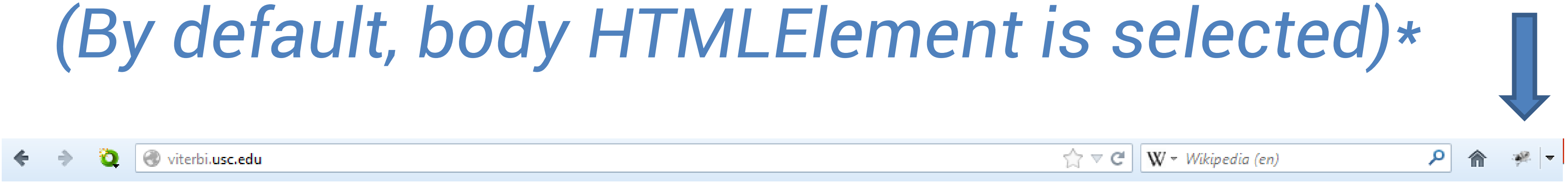
1) Press **F12** on the keyboard

*(By default, body HTMLElement is selected)**

OR

2) Press the Firebug button on the toolbar

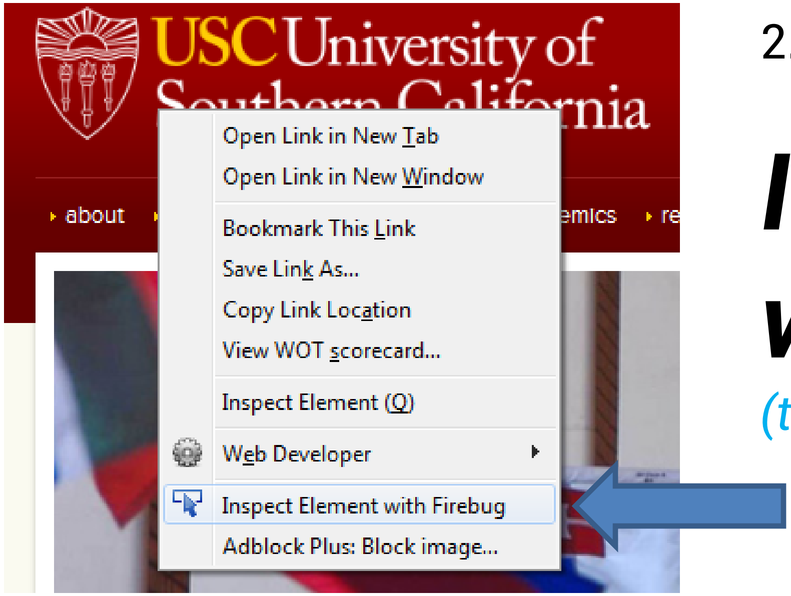
*(By default, body HTMLElement is selected)**



***Note:** Firebug may continue from a last saved session if it is still running

The 3rd Way to Launch Firebug

1) In the current webpage, **<u>right click</u>** on an element (an image, text, background, etc).



Open Link in New Tab
Open Link in New Window
Bookmark This Link
Save Link As...
Copy Link Location
View WOT scorecard...
Inspect Element (Q)
Web Developer
Inspect Element with Firebug
Adblock Plus: Block image...

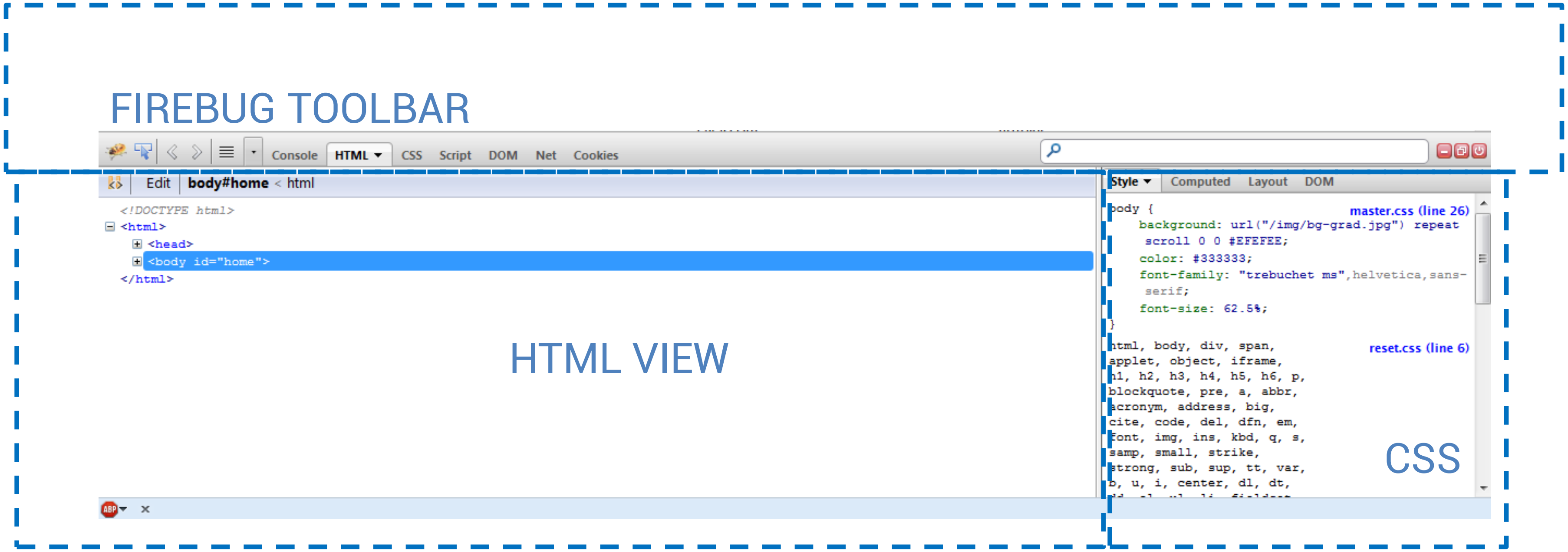
2. In the dropdown menu, click on

Inspect Element with Firebug...

(the element you right clicked on is selected.)

USC Viterbi School of Engineering

Hello, Firebug!



```
<!DOCTYPE html>
<html>
  <head>
  <body id="home">
</html>
```

master.css (line 26)

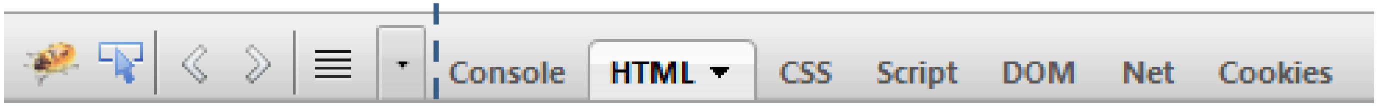
```
body {
    background: url("/img/bg-grad.jpg") repeat
     scroll 0 0 #EFEFEE;
    color: #333333;
    font-family: "trebuchet ms",helvetica,sans-
     serif;
    font-size: 62.5%;
}
```

reset.css (line 6)

```
html, body, div, span,
applet, object, iframe,
h1, h2, h3, h4, h5, h6, p,
blockquote, pre, a, abbr,
acronym, address, big,
cite, code, del, dfn, em,
font, img, ins, kbd, q, s,
samp, small, strike,
strong, sub, sup, tt, var,
b, u, i, center, dl, dt,
dd, ol, ul, li, fieldset
```

*Let's take a look at the **Firebug Toolbar** first, as we will use this throughout the tutorial.*

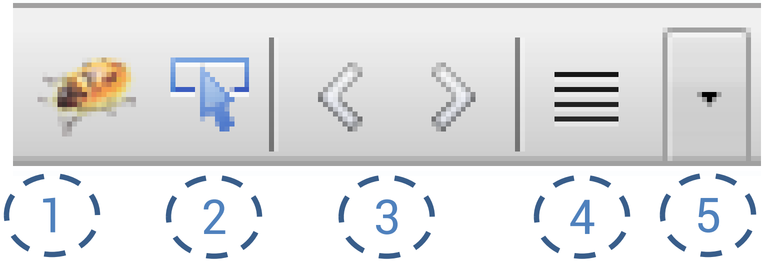
Firebug Toolbar



PANELS

Console: Brings up a Interactive JavaScript Console
HTML: Brings up the HTML View (see previous)
CSS: Brings up the CSS View
Script: Brings up the JavaScript Debugger (used later)
DOM: A list of all the DOM Properties *(defaults to window object)*
Net: Displays requests made from the browser
Cookies: Displays sessions & cookies from the browser

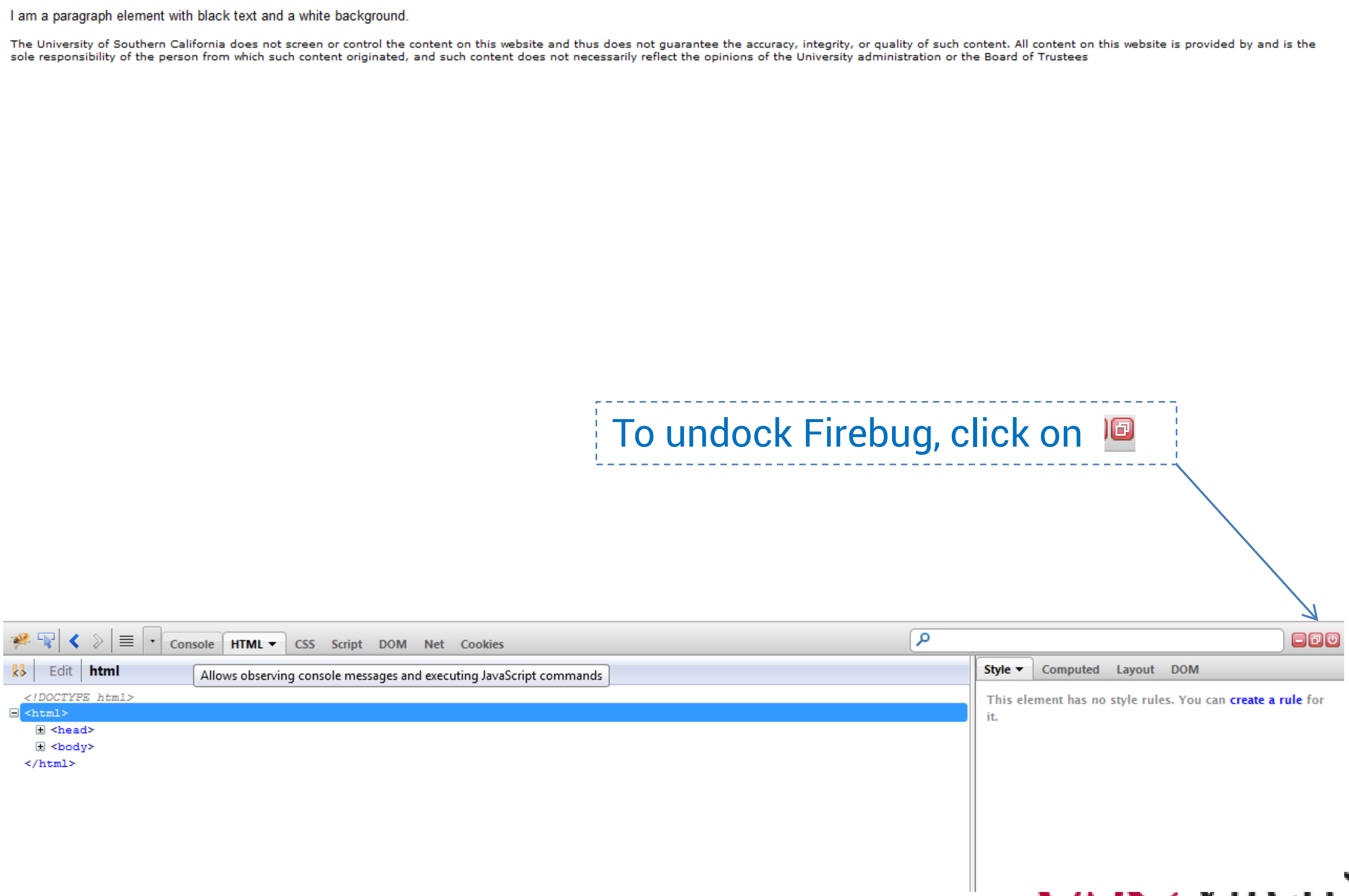
Firebug Toolbar – Firebug Button



1. **The Firebug Button**
    a. Hide Firebug (hides the panel)
    b. Deactivate Firebug (turns off Firebug)
    c. Firebug UI Location…
        a. Detached
        b. Left/Right
        c. Top/Bottom
    d. Open with Editor
    e. Options
    f. Firebug Online
    g. Customize Shortcuts
    h. About

Bold: Useful things in Firebug

Firebug UI Locations - Docked



I am a paragraph element with black text and a white background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

Allows observing console messages and executing JavaScript commands

```
<!DOCTYPE html>
⊟<html>
  ⊞<head>
  ⊞<body>
</html>
```

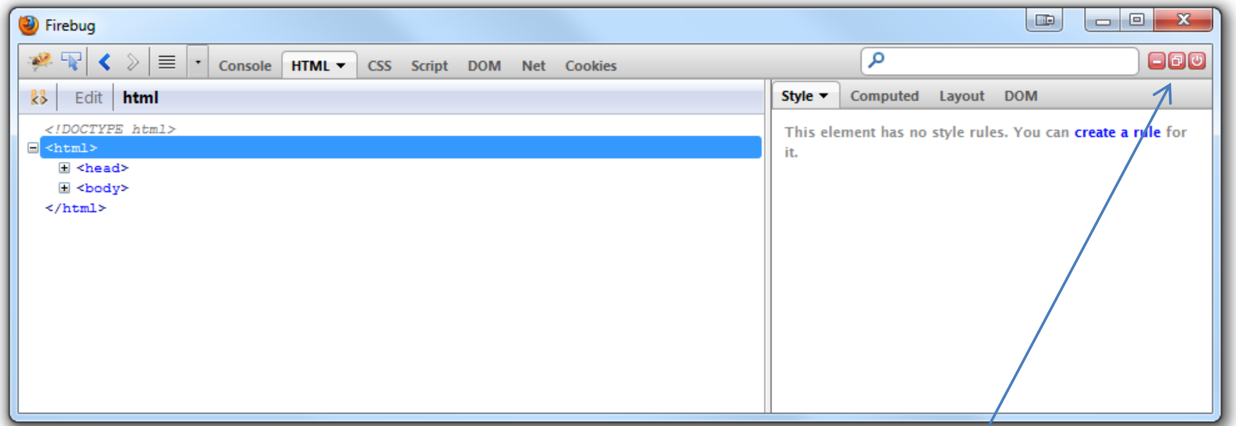
This element has no style rules. You can create a rule for it.

To undock Firebug, click on

Firebug UI Locations - UnDocked

I am a paragraph element with black text and a white background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

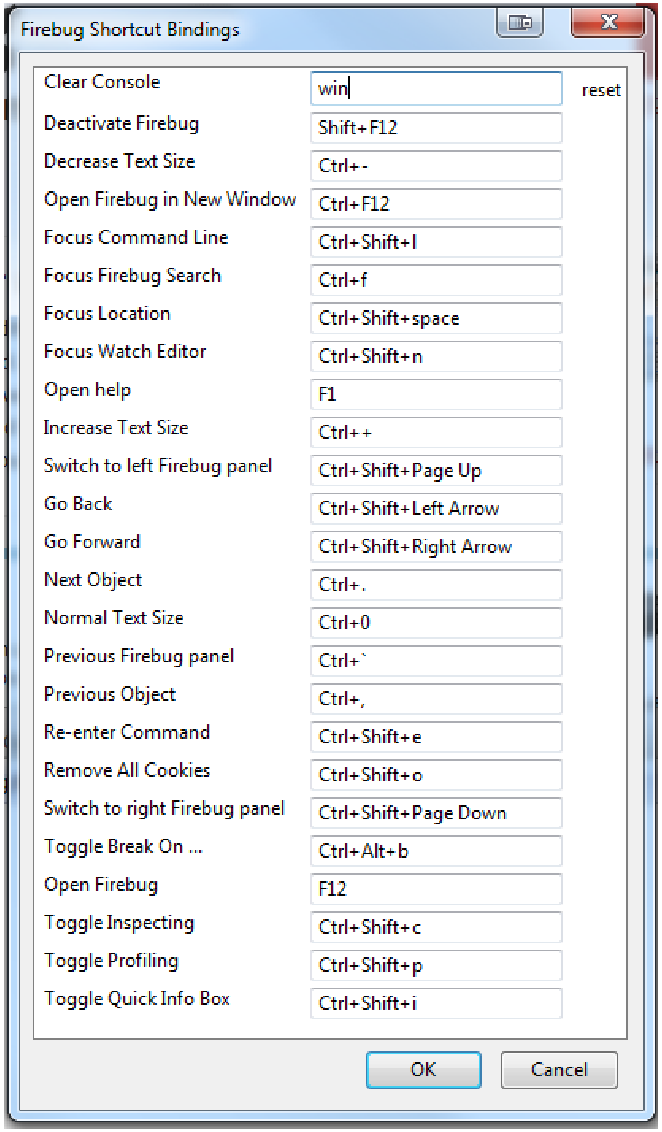


```
<!DOCTYPE html>
<html>
  <head>
  <body>
</html>
```

This element has no style rules. You can create a rule for it.

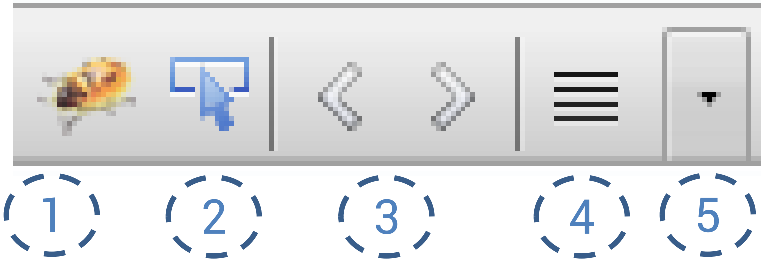
To **dock** Firebug, click on

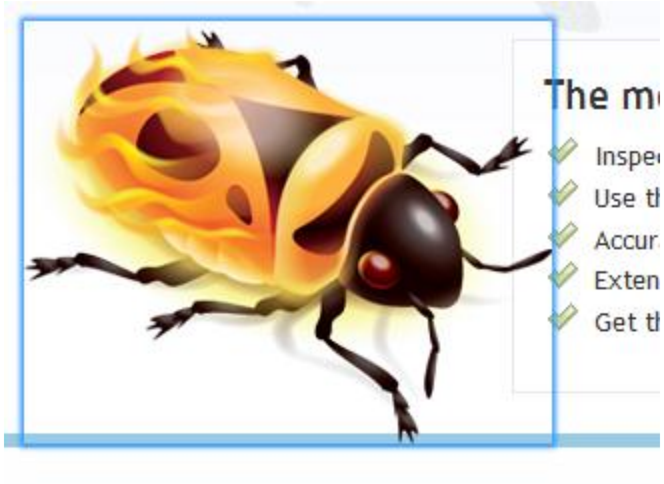
List of Useful Firebug Shortcuts



Clear Console	win
Deactivate Firebug	Shift+F12
Decrease Text Size	Ctrl+-
Open Firebug in New Window	Ctrl+F12
Focus Command Line	Ctrl+Shift+l
Focus Firebug Search	Ctrl+f
Focus Location	Ctrl+Shift+space
Focus Watch Editor	Ctrl+Shift+n
Open help	F1
Increase Text Size	Ctrl++
Switch to left Firebug panel	Ctrl+Shift+Page Up
Go Back	Ctrl+Shift+Left Arrow
Go Forward	Ctrl+Shift+Right Arrow
Next Object	Ctrl+.
Normal Text Size	Ctrl+0
Previous Firebug panel	Ctrl+`
Previous Object	Ctrl+,
Re-enter Command	Ctrl+Shift+e
Remove All Cookies	Ctrl+Shift+o
Switch to right Firebug panel	Ctrl+Shift+Page Down
Toggle Break On ...	Ctrl+Alt+b
Open Firebug	F12
Toggle Inspecting	Ctrl+Shift+c
Toggle Profiling	Ctrl+Shift+p
Toggle Quick Info Box	Ctrl+Shift+i

Firebug Toolbar – Inspect Element Button



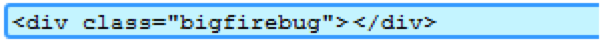


A Hovered Element using Inspect Element Button

2. **Inspect Element** (this is similar to Right click & Inspect Element with Firebug)

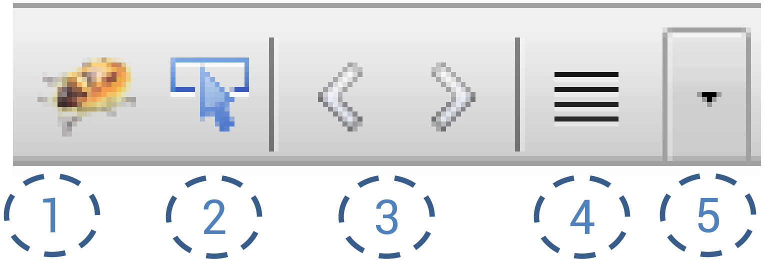
The difference: When hovering over elements in the page, the element is highlighted.

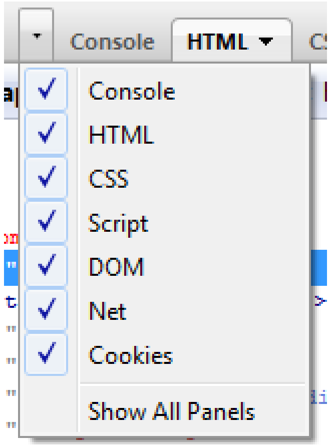
Also, the element is also highlighted in blue in the HTML View

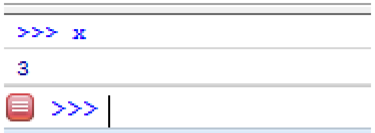


```
<div class="bigfirebug"></div>
```

Firebug Toolbar – Arrows, Quick Console, Show Hide Panels







```
>>> x
3
>>> |
```

3. Back/Forward – Switches between Panels

4. Quick Console – Interactive JavaScript console

5. Show or Hide Panels – Show or hide all panels

Firebug Tutorials Outline

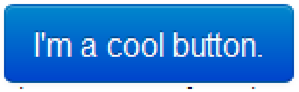
1. Inspecting HTML/CSS elements and their properties
2. Modifying HTML/CSS elements in real time
3. Debugging JavaScript & Analyzing Behavior of JavaScript code
4. Web Performance
5. Cookies and Sessions

Inspecting HTML/CSS Elements and their Properties

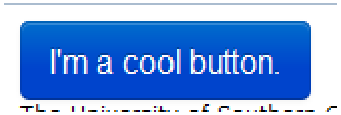
1) Click here for the tutorial: http://www-scf.usc.edu/~goei/571-firebug/lesson1.html

TUTORIAL OBJECTIVES

We are given the following element:



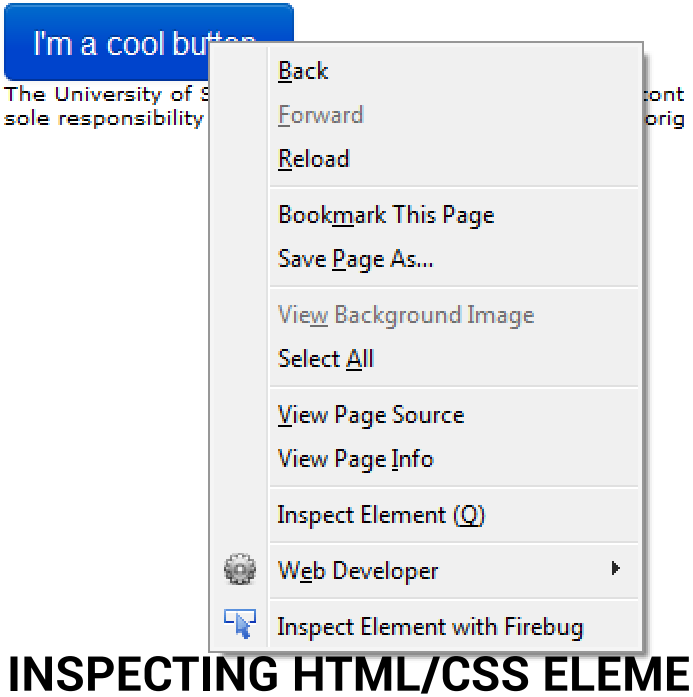
1) Find out the attributes & DOM properties of the element
2) Find out the computed CSS properties
3) Draw the box model for the element.
4) When hovering the element, determine the CSS properties.

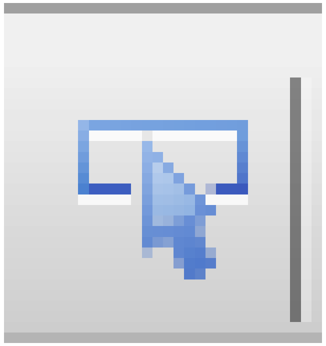


Hovered

Finding the Attributes & DOM properties of the Element

1) Use the Inspect Element feature in Firebug. (**Right click** on the element, and click on **Inspect Element with Firebug**)

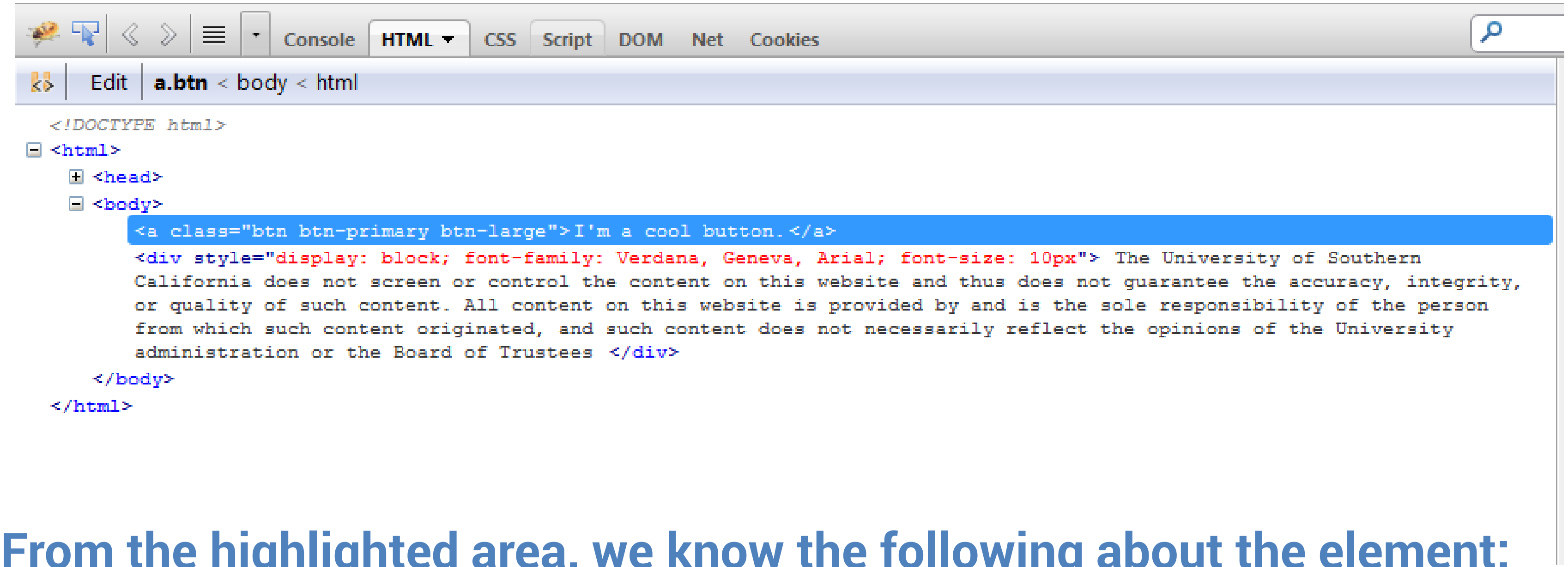




***Note**: We can also use this button and click on the element*

Finding the Attributes & DOM properties of the Element

2) The HTML Panel is displayed with the element selected.

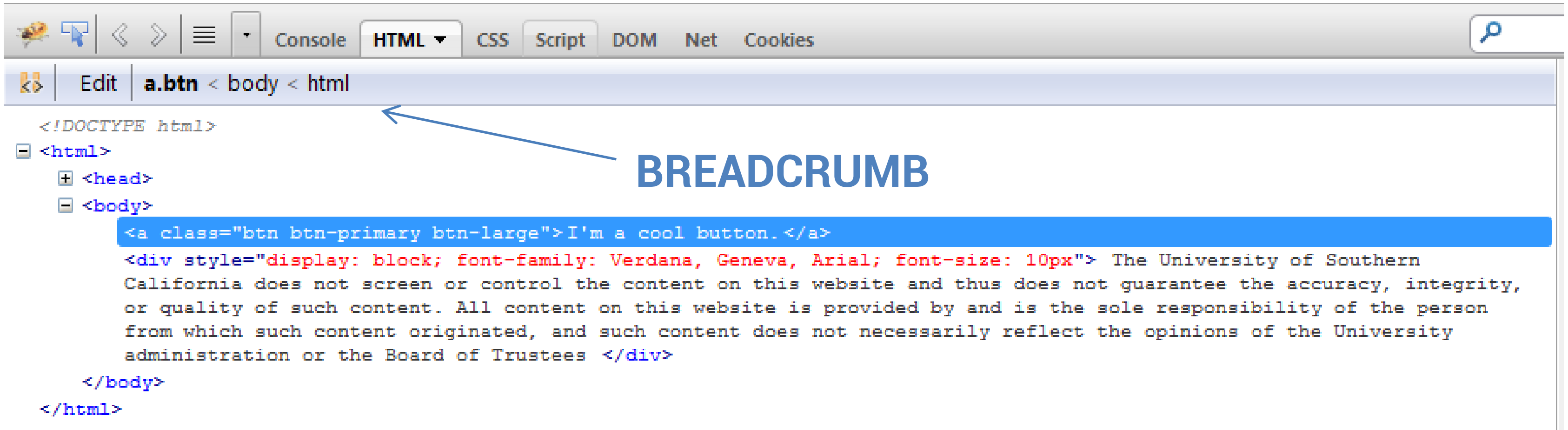


```
<!DOCTYPE html>
<html>
  <head>
  <body>
      <a class="btn btn-primary btn-large">I'm a cool button.</a>
      <div style="display: block; font-family: Verdana, Geneva, Arial; font-size: 10px"> The University of Southern
      California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity,
      or quality of such content. All content on this website is provided by and is the sole responsibility of the person
      from which such content originated, and such content does not necessarily reflect the opinions of the University
      administration or the Board of Trustees </div>
  </body>
</html>
```

From the highlighted area, we know the following about the element:

```
<a class="btn btn-primary btn-large">I'm a cool button.</a>
```

Finding the Attributes & DOM properties of the Element



BREADCRUMB

```
  <!DOCTYPE html>
⊟ <html>
    ⊞ <head>
    ⊟ <body>
          <a class="btn btn-primary btn-large">I'm a cool button.</a>
          <div style="display: block; font-family: Verdana, Geneva, Arial; font-size: 10px"> The University of Southern
          California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity,
          or quality of such content. All content on this website is provided by and is the sole responsibility of the person
          from which such content originated, and such content does not necessarily reflect the opinions of the University
          administration or the Board of Trustees </div>
      </body>
  </html>
```

From the highlighted area & the breadcrumb we know the following about the element:

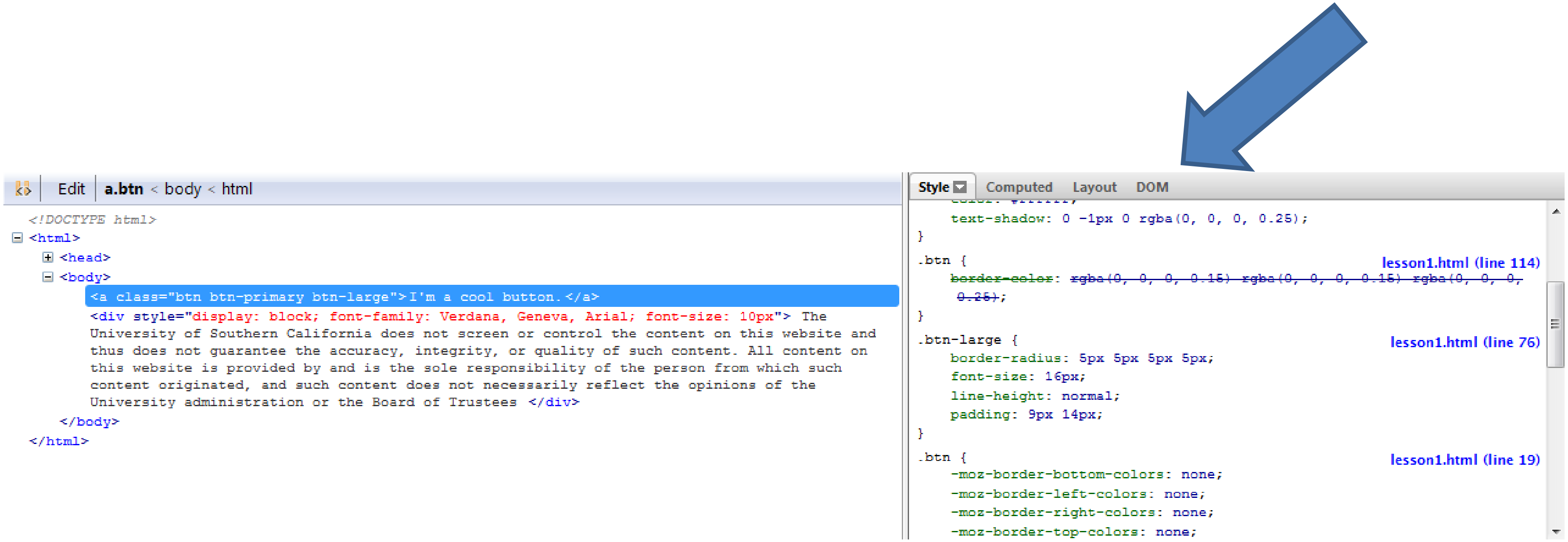
```
<a class="btn btn-primary btn-large">I'm a cool button.</a>
```

1. The element is an anchor
2. The element has a class attribute `btn btn-primary btn-large`
3. The element has a TextNode with TextContent "I'm a cool button"
4. The element's parent node is body (whose parent node is html)

There's a lot more we can say about the element though!

Finding the Attributes & DOM properties of the Element

4) Click on the DOM panel on the CSS sidebar on the HTML view.

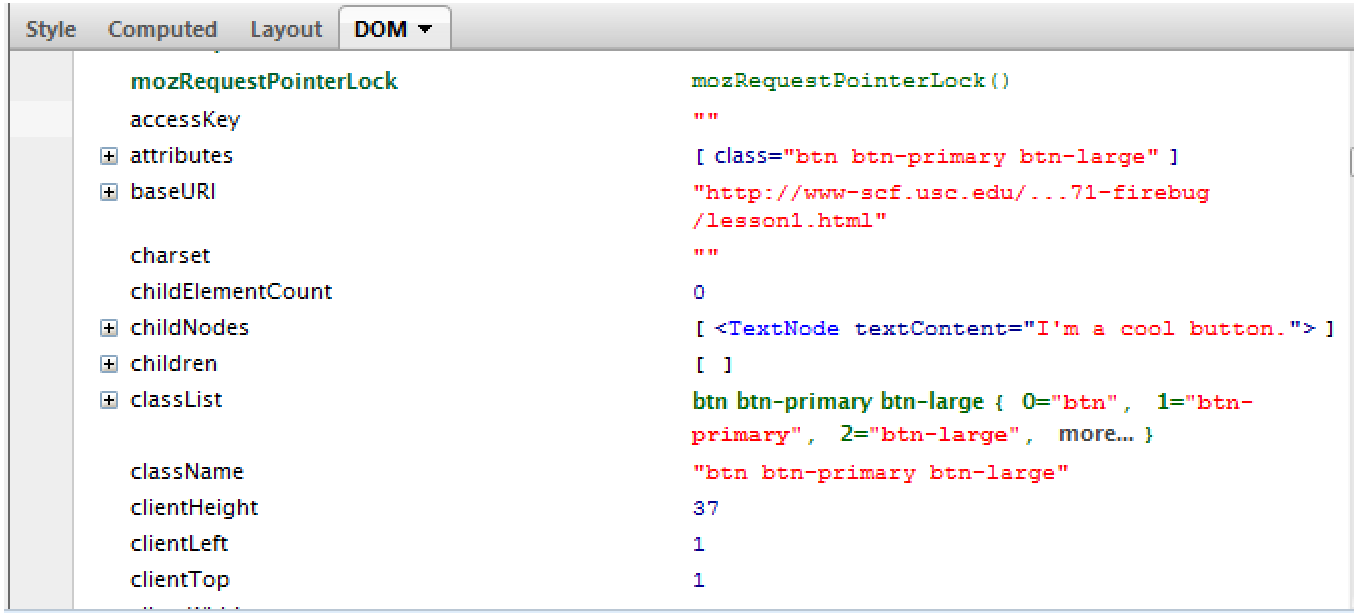


```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <a class="btn btn-primary btn-large">I'm a cool button.</a>
    <div style="display: block; font-family: Verdana, Geneva, Arial; font-size: 10px"> The
    University of Southern California does not screen or control the content on this website and
    thus does not guarantee the accuracy, integrity, or quality of such content. All content on
    this website is provided by and is the sole responsibility of the person from which such
    content originated, and such content does not necessarily reflect the opinions of the
    University administration or the Board of Trustees </div>
  </body>
</html>
```

```
    text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);
}
.btn {                                                        lesson1.html (line 114)
    ~~border-color: rgba(0, 0, 0, 0.15) rgba(0, 0, 0, 0.15) rgba(0, 0, 0,
     0.25);~~
}
.btn-large {                                                   lesson1.html (line 76)
    border-radius: 5px 5px 5px 5px;
    font-size: 16px;
    line-height: normal;
    padding: 9px 14px;
}
.btn {                                                         lesson1.html (line 19)
    -moz-border-bottom-colors: none;
    -moz-border-left-colors: none;
    -moz-border-right-colors: none;
    -moz-border-top-colors: none;
```

Finding the Attributes & DOM properties of the Element

A list of all DOM properties are displayed for the current element selected.



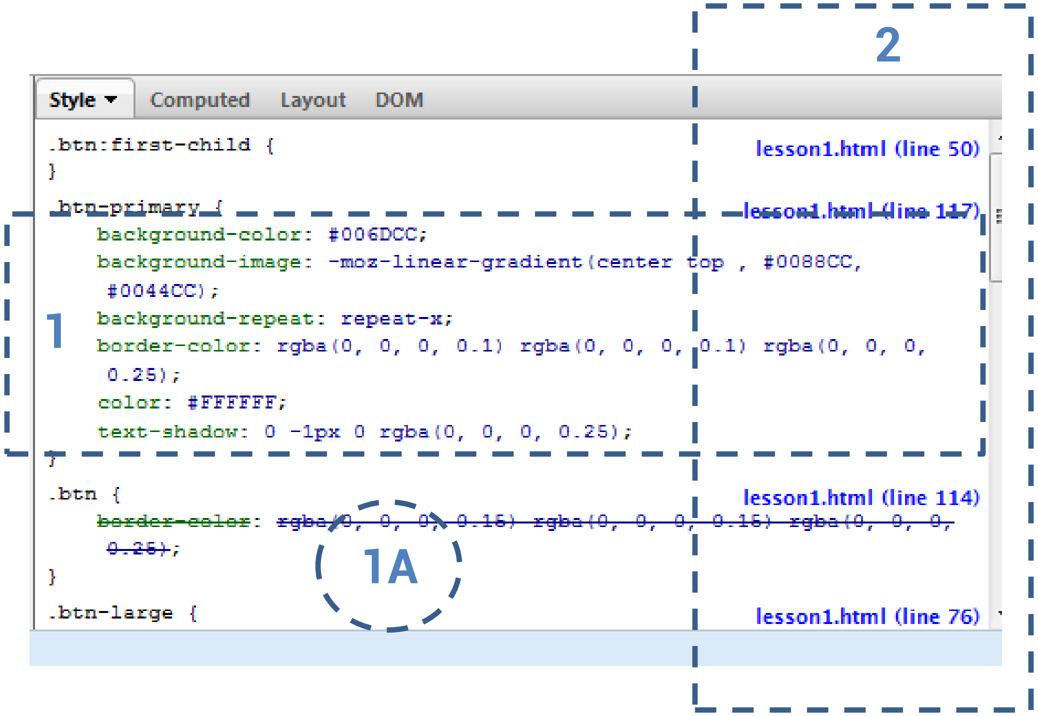
```
   mozRequestPointerLock      mozRequestPointerLock()
   accessKey                  ""
⊞  attributes                 [ class="btn btn-primary btn-large" ]
⊞  baseURI                    "http://www-scf.usc.edu/...71-firebug
                              /lesson1.html"
   charset                    ""
   childElementCount          0
⊞  childNodes                 [ <TextNode textContent="I'm a cool button."> ]
⊞  children                   [ ]
⊞  classList                  btn btn-primary btn-large { 0="btn", 1="btn-
                              primary", 2="btn-large", more... }
   className                  "btn btn-primary btn-large"
   clientHeight               37
   clientLeft                 1
   clientTop                  1
```

Notice: We have an attributes member variable, which has an array of attributes. This is another way how to determine the attributes for the selected element.

Find the Computed CSS Properties for that Element

1) Click on Style in the right hand column.

A list of CSS properties are shown, from most relevant to least.



```
.btn:first-child {                                          lesson1.html (line 50)
}
.btn-primary {                                              lesson1.html (line 117)
    background-color: #006DCC;
    background-image: -moz-linear-gradient(center top , #0088CC,
     #0044CC);
    background-repeat: repeat-x;
    border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0,
     0.25);
    color: #FFFFFF;
    text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);
}
.btn {                                                      lesson1.html (line 114)
    ~~border-color: rgba(0, 0, 0, 0.15) rgba(0, 0, 0, 0.15) rgba(0, 0, 0,
     0.25);~~
}
.btn-large {                                                lesson1.html (line 76)
```

What we know from the diagram:

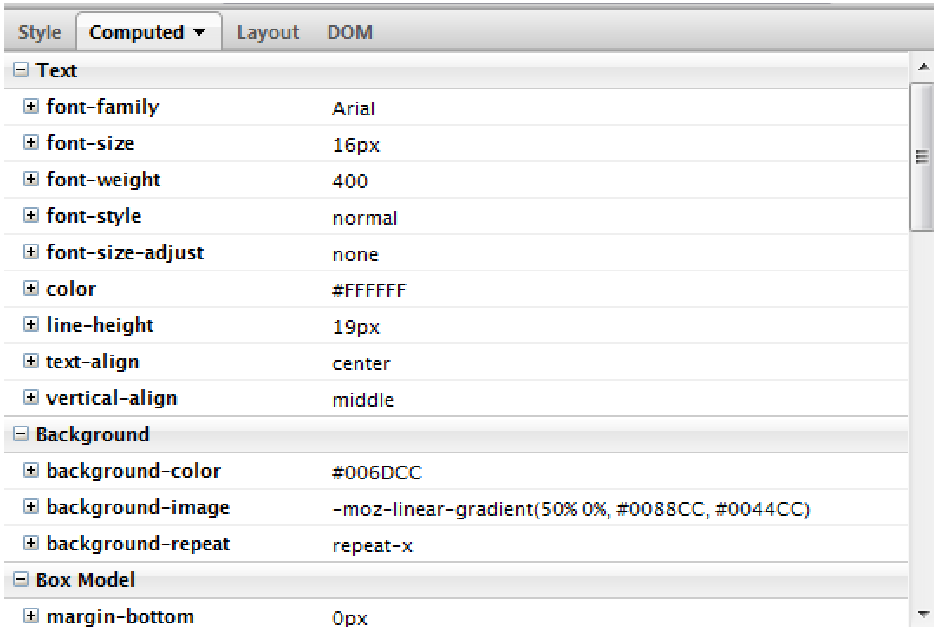
1) List of CSS properties
   a) **crossed out** – It means the style was cascaded
2) Where it is located in the server
   - Which file,
   - What line number does the CSS property start
   - Clicking on `lesson1.html (line 50)` brings up the file in the CSS panel.

Tip: If you did not mean for a css property to be cascaded, you can simply add an `!important` to the cascaded property, or you must rearrange the CSS properties (files) (bottom CSS cascades the top CSS)

Find the Computed CSS Properties for that Element

2) Click on Computed

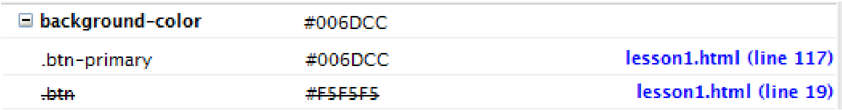
A list of Computed CSS properties are displayed.



```
Style   Computed ▾   Layout   DOM

⊟ Text
  ⊞ font-family          Arial
  ⊞ font-size            16px
  ⊞ font-weight          400
  ⊞ font-style           normal
  ⊞ font-size-adjust     none
  ⊞ color                #FFFFFF
  ⊞ line-height          19px
  ⊞ text-align           center
  ⊞ vertical-align       middle
⊟ Background
  ⊞ background-color     #006DCC
  ⊞ background-image     -moz-linear-gradient(50% 0%, #0088CC, #0044CC)
  ⊞ background-repeat    repeat-x
⊟ Box Model
  ⊞ margin-bottom        0px
```

Tip: Clicking on the +/- displays the computation for the CSS property

(Cascaded properties are crossed out)



⊟ background-color   #006DCC

.btn-primary   #006DCC   lesson1.html (line 117)

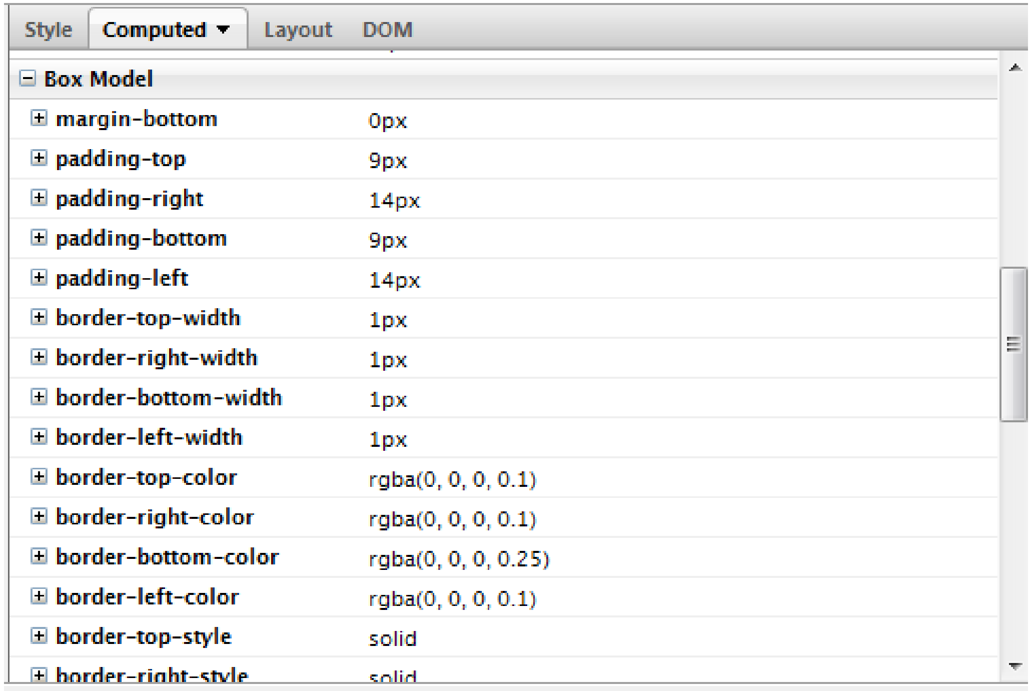
~~.btn~~   ~~#F5F5F5~~   lesson1.html (line 19)

Next objective, finding the box model of the element

Find the Computed CSS Properties for that Element

1) Stay on the Computed panel and scroll to box model.

A list of Box Model CSS properties are shown..



⊟ Box Model

⊞ margin-bottom	0px
⊞ padding-top	9px
⊞ padding-right	14px
⊞ padding-bottom	9px
⊞ padding-left	14px
⊞ border-top-width	1px
⊞ border-right-width	1px
⊞ border-bottom-width	1px
⊞ border-left-width	1px
⊞ border-top-color	rgba(0, 0, 0, 0.1)
⊞ border-right-color	rgba(0, 0, 0, 0.1)
⊞ border-bottom-color	rgba(0, 0, 0, 0.25)
⊞ border-left-color	rgba(0, 0, 0, 0.1)
⊞ border-top-style	solid
⊞ border-right-style	solid

We need to draw out the box model, however…

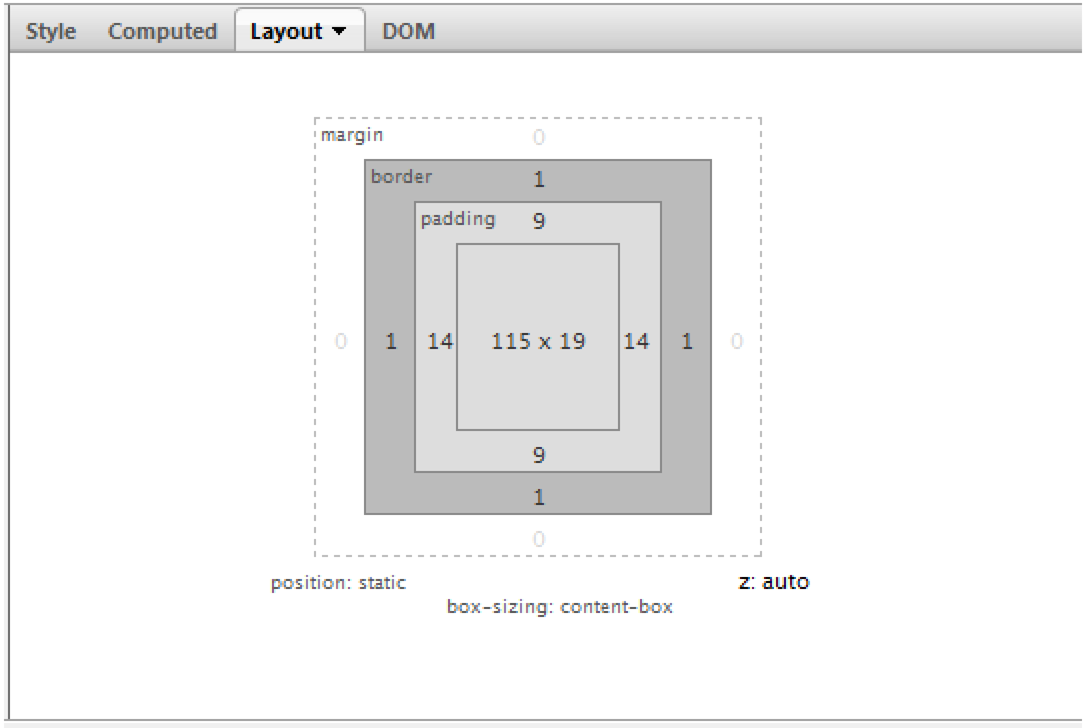
Find the Computed CSS Properties for that Element

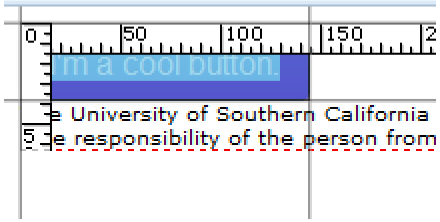
2) Click on Layout

The Box Model is drawn out. When highlighting over an area, the browser highlights the selected element's box property & displays a ruler

Purple highlight: Padding

Yellow highlight: Margin

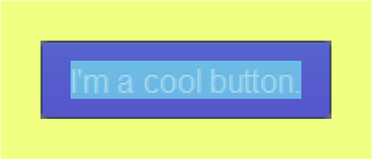




I'm a cool button.

e University of Southern California

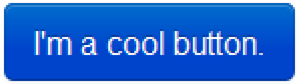
e responsibility of the person from



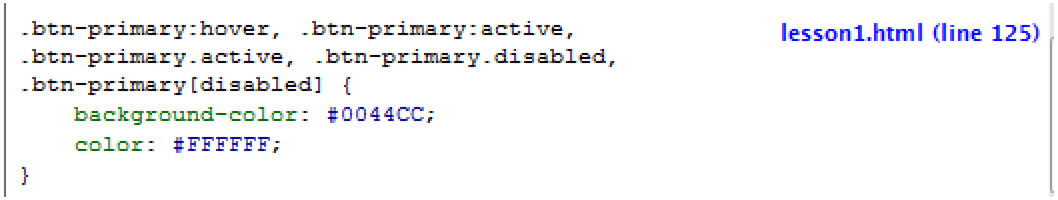
Determining Hover CSS Properties of the Element

1) **Hover over the element.**

The Style Panel will change accordingly.



Added CSS properties caused by the psuedo-class hover.



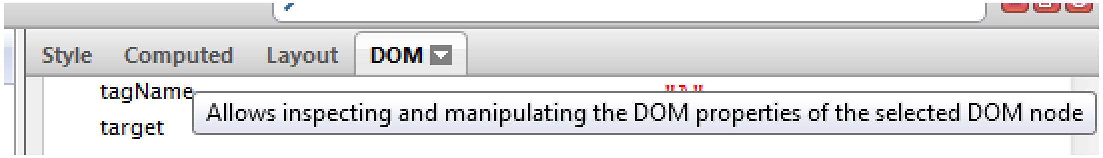
```
.btn-primary:hover, .btn-primary:active,                lesson1.html (line 125)
.btn-primary.active, .btn-primary.disabled,
.btn-primary[disabled] {
    background-color: #0044CC;
    color: #FFFFFF;
}
```

Tutorial 1 Summary

TUTORIAL OBJECTIVES

1) Find out the attributes & DOM properties of the element

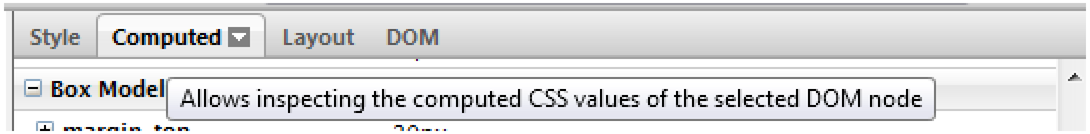
Solution: In the HTML Panel, click DOM on the right side pane.



Allows inspecting and manipulating the DOM properties of the selected DOM node

2) Find out the computed CSS properties

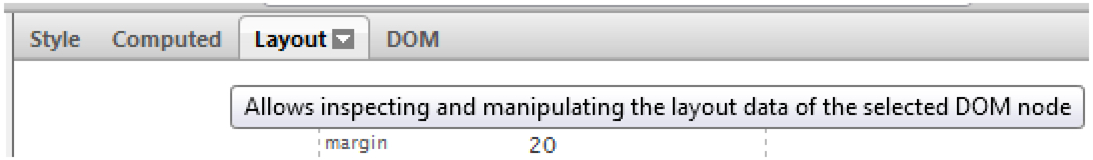
Solution: In the HTML Panel, click DOM on the right side pane



Allows inspecting the computed CSS values of the selected DOM node

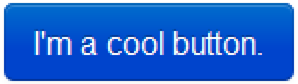
3) Draw the box model for the element.

Solution: In the HTML Panel, click on Layout



Allows inspecting and manipulating the layout data of the selected DOM node

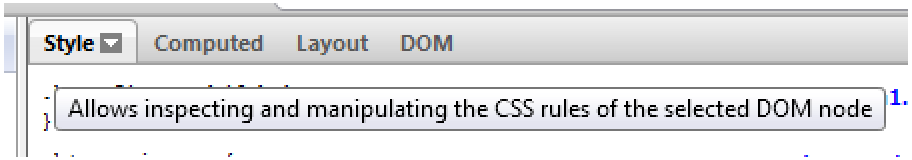
Tutorial 1 Summary



TUTORIAL OBJECTIVES

4) Determining CSS Hover Properties of the element

Solution: Hover over the element, watch the Style pane in HTML View



Allows inspecting and manipulating the CSS rules of the selected DOM node

HOW IS THIS USEFUL?

1) Can visualize how an element is formed in CSS/HTML/JavaScript

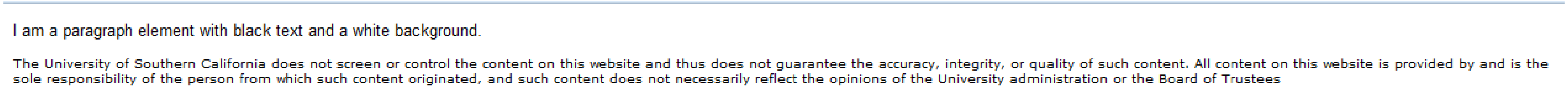
Example – A client of yours likes a button at Google+. The client wants that same exact button in their web application. **Time to use Firebug.**

Modifying the DOM of an Element in Real Time

Go to http://www-scf.usc.edu/~goei/571-firebug/lesson2.html

TUTORIAL OBJECTIVES

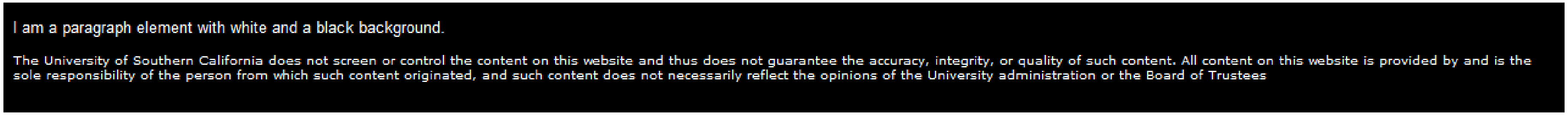
Transform this page:



I am a paragraph element with black text and a white background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

TO



I am a paragraph element with white and a black background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

WHAT WE WILL LEARN

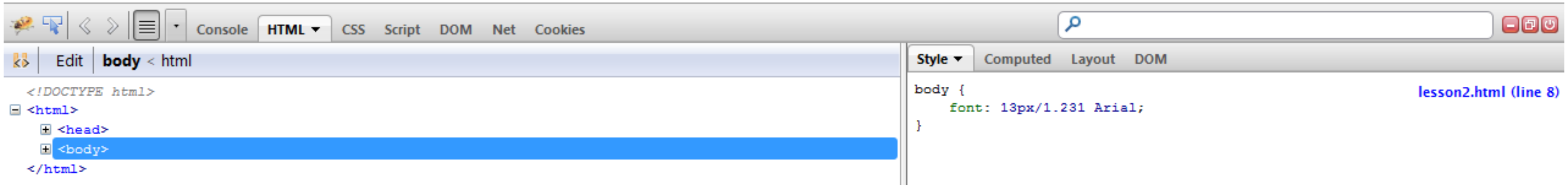
How to change the view of an element in real time in the browser, even though we don't have access rights to modifying the file.

Modifying the DOM of an Element in Real Time

Go to http://www-scf.usc.edu/~goei/571-firebug/lesson2.html

Method 1: Using the HTML View & Style Pane

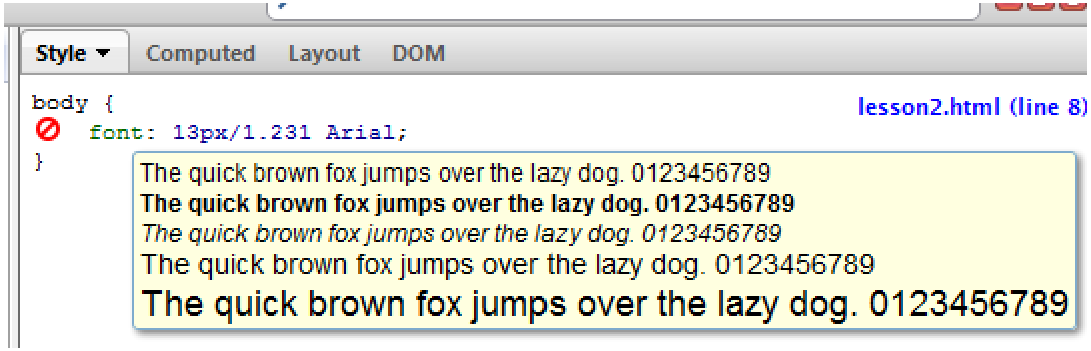
1) **Launch Firebug** and under the **HTML View** inspect the body element.
   - Easy way to do this: F12, click on the body tag. It is highlighted.



```
<!DOCTYPE html>
<html>
  <head>
  <body>
</html>
```

```
body {
    font: 13px/1.231 Arial;
}
```

1a) If we hover over: `font: 13px/1.231 Arial;`



```
body {
    font: 13px/1.231 Arial;
}
```

The quick brown fox jumps over the lazy dog. 0123456789

The quick brown fox jumps over the lazy dog. 0123456789

The quick brown fox jumps over the lazy dog. 0123456789

The quick brown fox jumps over the lazy dog. 0123456789

The quick brown fox jumps over the lazy dog. 0123456789

Hovering over the element does 2 things:

1) shows an info box about the property (if available)
2) Clicking on ⊘ will hide the selected property.

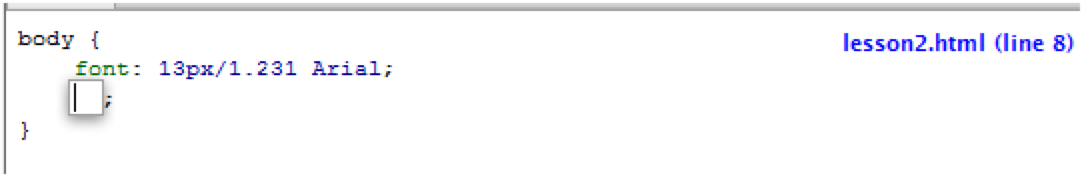
USC Viterbi
School of Engineering

Modifying the DOM of an Element in Real Time

Go to http://www-scf.usc.edu/~goei/571-firebug/lesson2.html

Method 1: Using the HTML View & Style Pane

2) Click on the braces { } of the CSS element declaration. Firebug will allow you to add a new css property.



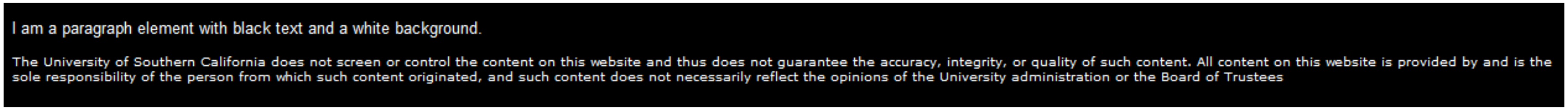
lesson2.html (line 8)

```
body {
    font: 13px/1.231 Arial;
    ;
}
```

3) Add the following properties:

```
color: white;
background: black;
```

Result… but we are not done yet



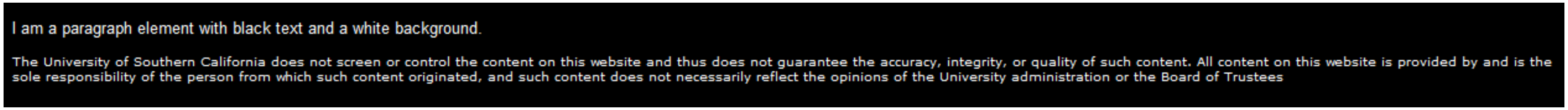
I am a paragraph element with black text and a white background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

Modifying the DOM of an Element in Real Time

Go to http://www-scf.usc.edu/~goei/571-firebug/lesson2.html

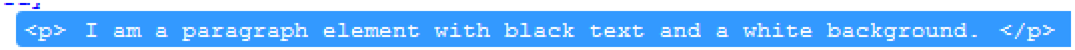
Method 1: Using the HTML View & Style Pane



I am a paragraph element with black text and a white background.

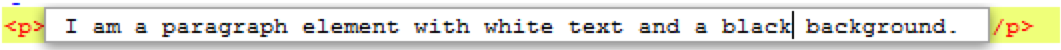
The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

4) Inspect the paragraph element, `I am a paragraph element with black text and a white background`

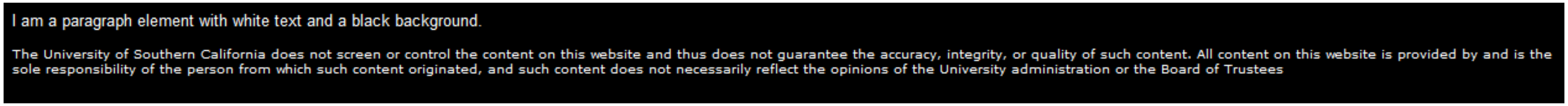


```
<p> I am a paragraph element with black text and a white background. </p>
```

5) Double click on the innerText of the element, and modify accordingly.



```
<p> I am a paragraph element with white text and a black background. /p>
```



I am a paragraph element with white text and a black background.

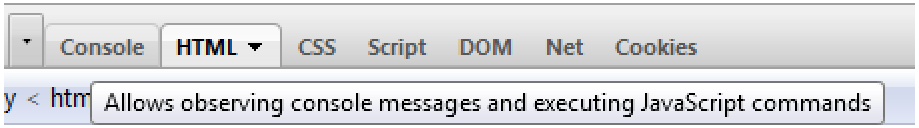
The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

Modifying the DOM of an Element in Real Time

Go to http://www-scf.usc.edu/~goei/571-firebug/lesson2.html

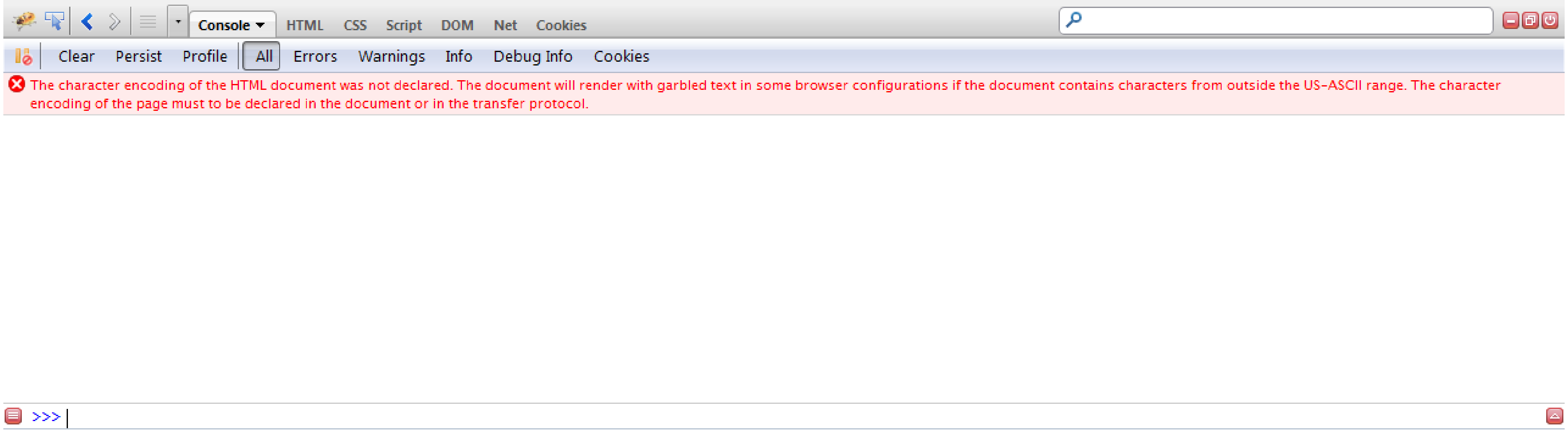
Method 2: Using the Interactive JavaScript Console

1) Click on Console.



Allows observing console messages and executing JavaScript commands

CONSOLE PANEL



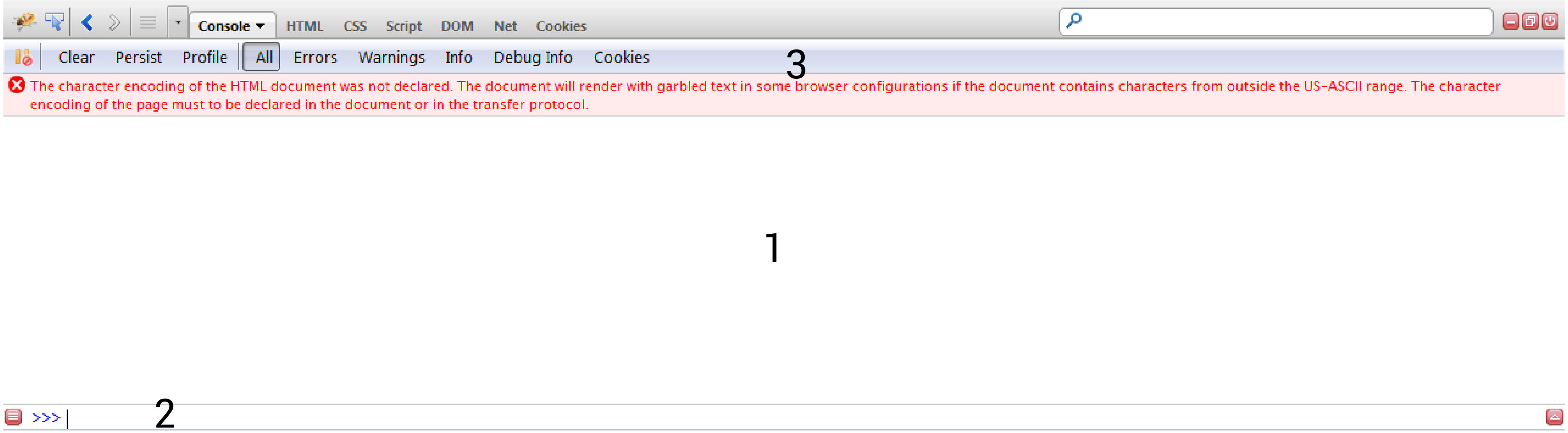
The character encoding of the HTML document was not declared. The document will render with garbled text in some browser configurations if the document contains characters from outside the US-ASCII range. The character encoding of the page must to be declared in the document or in the transfer protocol.

```
>>>
```

Modifying the DOM of an Element in Real Time

Method 2: Using the Interactive JavaScript Console

CONSOLE PANEL

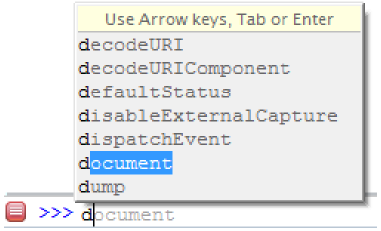


The character encoding of the HTML document was not declared. The document will render with garbled text in some browser configurations if the document contains characters from outside the US-ASCII range. The character encoding of the page must to be declared in the document or in the transfer protocol.

1. **The Interactive JavaScript Console**. All console messages (console.log), output of execution is displayed here.
2. **Write commands here**. As long the JavaScript is valid, any command here will be executed in **real time.**
3. **Filtering.** We can filter messages by their type. Clicking will cause the console to break on all errors.

Modifying the DOM of an Element in Real Time

Method 2: Using the Interactive JavaScript Console



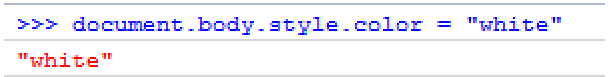
```
>>> document
```

1. Execute the following JavaScript line:
   (To execute JavaScript code, type the line starting at >>> and hit Enter)

```
document.body.style.color = “white”;
```

Neat! Firebug has autocomplete when typing in JavaScript commands

Result:



```
>>> document.body.style.color = "white"
"white"
```

Console

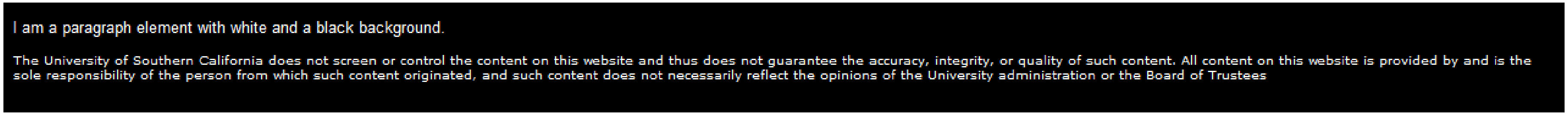
Browser view

What did we just do? We just set the text color of the body element to white, so currently we have white text on a white background (the border is there just to emphasize there is nothing)

2. Execute the following lines of JavaScript:

```
document.body.style.backgroundColor = “black”;
document.body.children[0].innerHTML = "I am a paragraph with white text and a black background“;
```

Result:



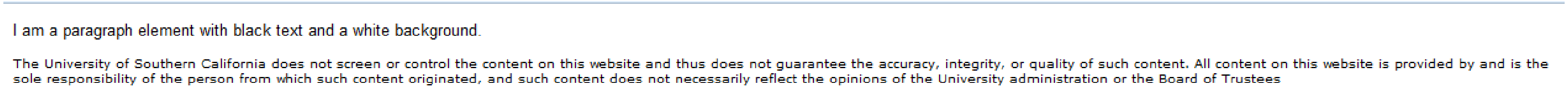
I am a paragraph element with white and a black background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

Modifying the DOM of an Element in Real Time

TUTORIAL OBJECTIVES

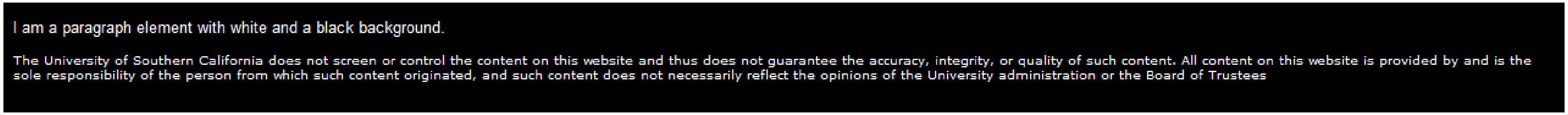
Transform this page: http://www-scf.usc.edu/~goei/571-firebug/lesson2.html



I am a paragraph element with black text and a white background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

TO



I am a paragraph element with white and a black background.

The University of Southern California does not screen or control the content on this website and thus does not guarantee the accuracy, integrity, or quality of such content. All content on this website is provided by and is the sole responsibility of the person from which such content originated, and such content does not necessarily reflect the opinions of the University administration or the Board of Trustees

HOW WE ACCOMPLISHED THE OBJECTIVE WITH FIREBUG WITH 2 METHODS:

1. Method 1: Changing HTML/CSS Properties using the HTML Panel and the CSS Pane
   Solution: Use the CSS Pane to change the background to black and the text-color, to white
   Solution: Use the HTML pane to change the text content of the element

2. Method 2: Use the Interactive JavaScript Console
   Solution: Write JavaScript DOM object code in the Console (`document.body.style.color….`)

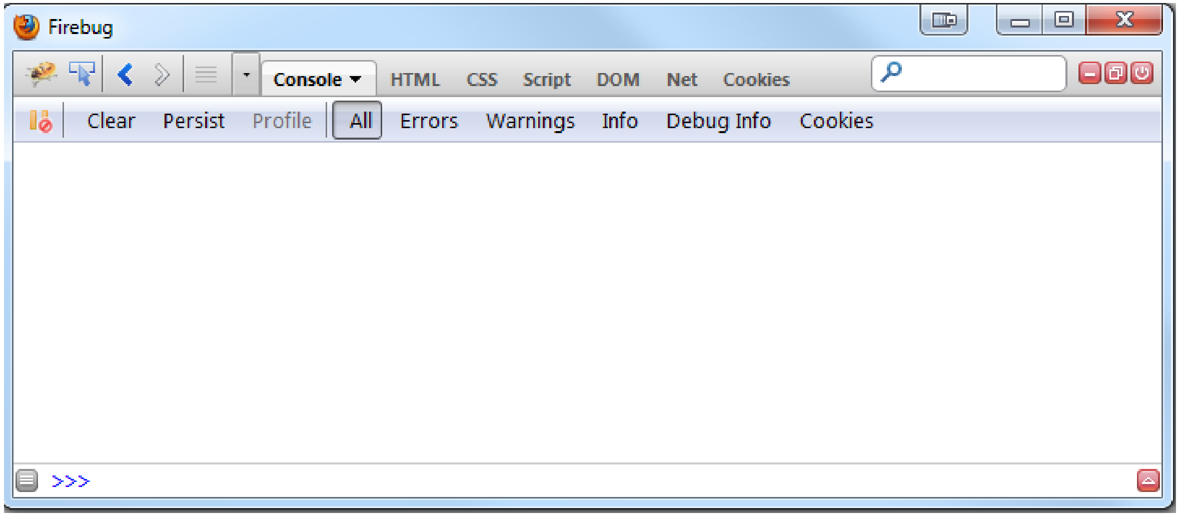
Side Note: Executing a Block of JavaScript Code

If you have **a block of JavaScript** code you would like to execute, such as the following:

```
var array = [“Harman”, “Goei”, “CSCI”, “571”];

for(var i = 0; i < array.length; i++) {
        console.log(array[i]);
}
```

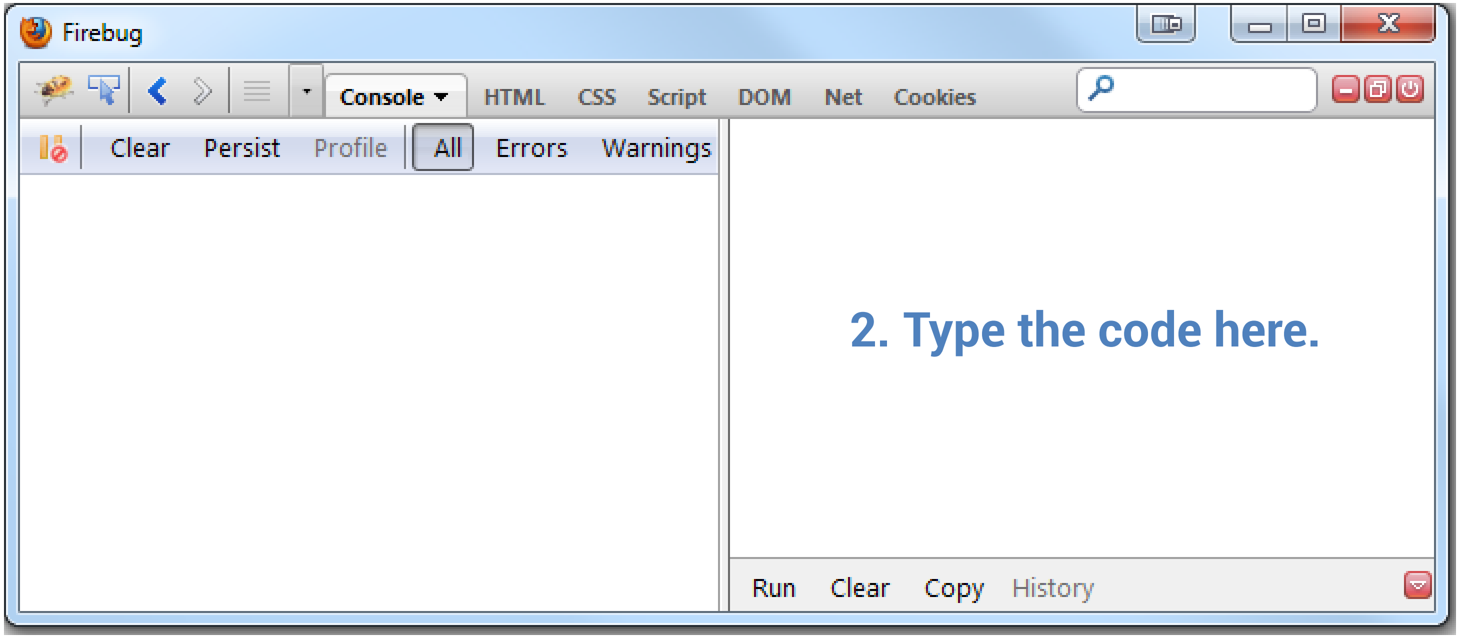
Instead of writing line by line, we can the entire block in the console. Here's how:

1. Press the ⏏ button in the Console tab.



Side Note: Executing a Block of JavaScript Code

The Console Panel should now be the following:

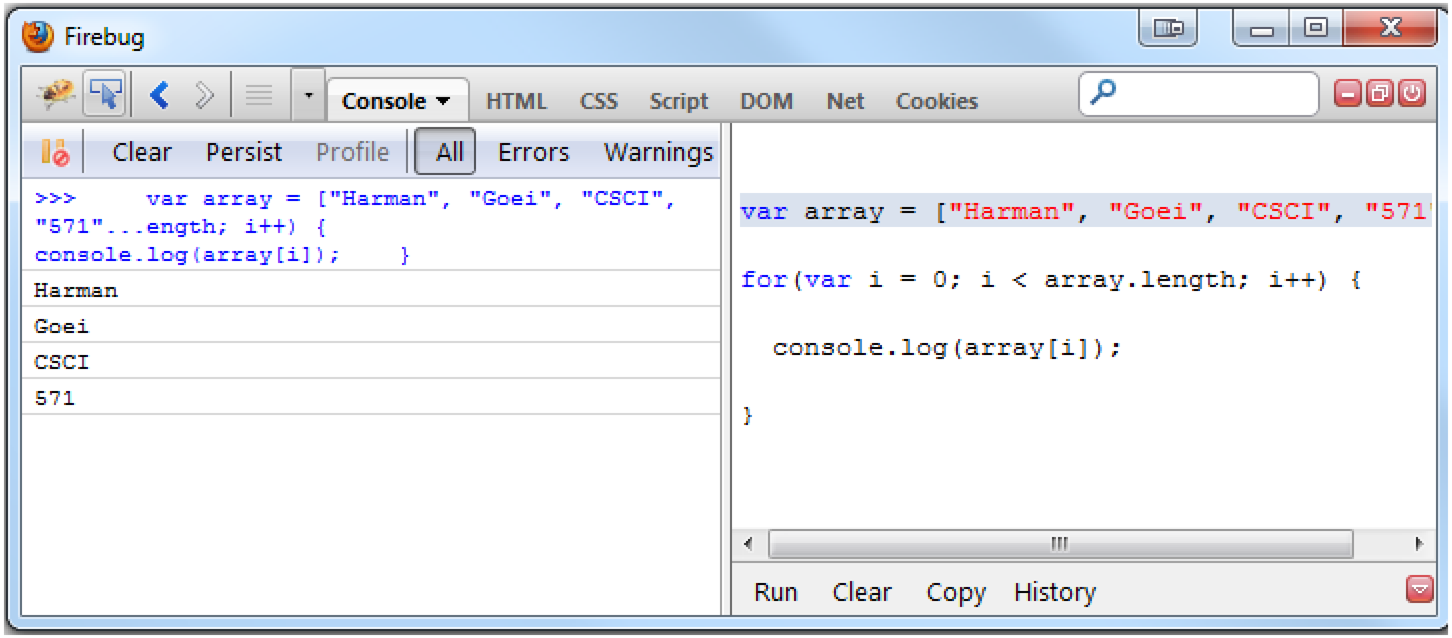


2. Type the code here.

3. Click **Run**.

Side Note: Executing a Block of JavaScript Code

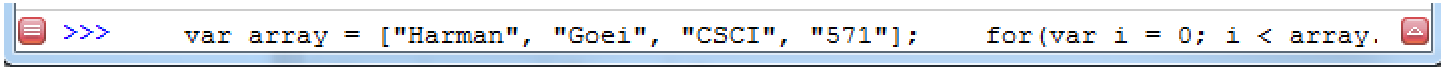
Result:



```
>>>     var array = ["Harman", "Goei", "CSCI",
"571"...ength; i++) {
console.log(array[i]);     }
Harman
Goei
CSCI
571
```

```
var array = ["Harman", "Goei", "CSCI", "571

for(var i = 0; i < array.length; i++) {

  console.log(array[i]);

}
```

Note: To go back to single-line mode, click on



```
>>>     var array = ["Harman", "Goei", "CSCI", "571"];     for(var i = 0; i < array.
```

WHY IS THIS USEFUL?

To write good JavaScript code, you have to test it frequently. We can write functions, test it, and ensure validity through the console. This practice is known as **test-driven development.**

Understanding Behavior of JavaScript Code & Detecting Errors with Firebug

TUTORIAL OBJECTIVES

http://www-scf.usc.edu/~goei/571-firebug/lesson3.html

1. Step through the behavior of the JavaScript code
2. Understand what happens in the console when JavaScript hits an error

We will be analyzing the following JavaScript:

```
var array = [1,2,3,4,5,6,7,8, "9"];

for(var i = 0; i < array.length; i++) {
      array[i] = array[i] + 1;
}
console.log(array);
setTimeout(function() { x=z; }, 5000);
```

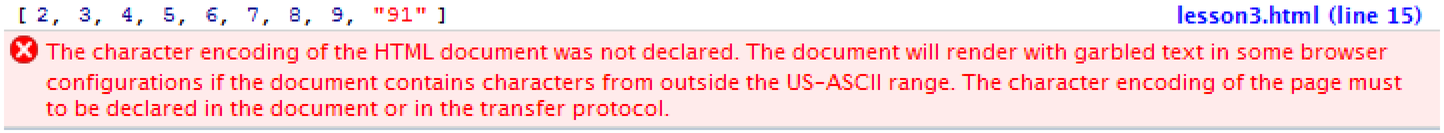
WHAT THE CODE DOES

1. Given an array, add 1 to each element.
2. In 5 seconds, a function will do an illegal operation in JavaScript.

Understanding Behavior of JavaScript Code & Detecting Errors with Firebug

1) Go to http://www-scf.usc.edu/~goei/571-firebug/lesson3.html

2) Go to the Console. You should see the following:

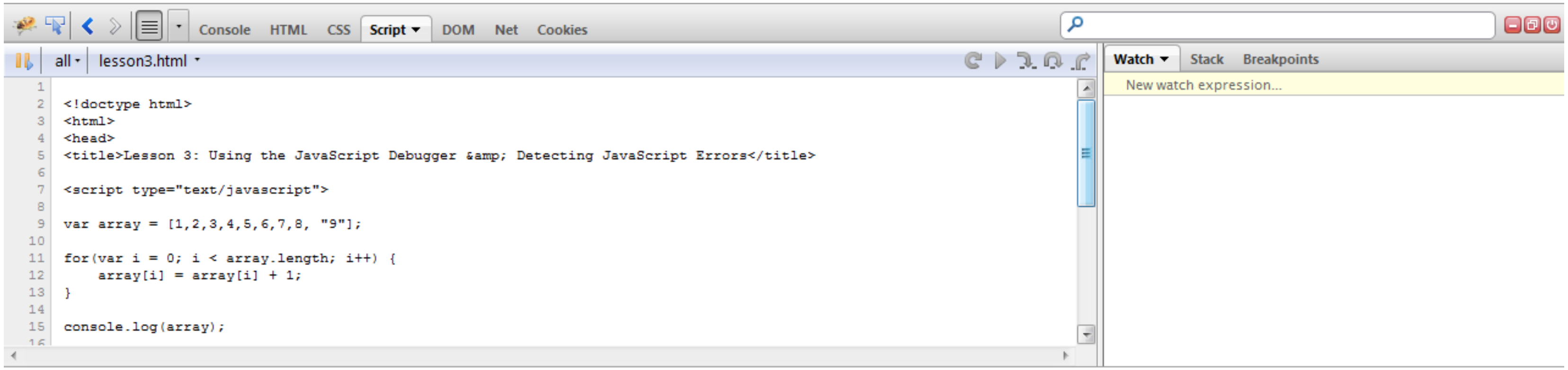


```
[ 2, 3, 4, 5, 6, 7, 8, 9, "91" ]                                    lesson3.html (line 15)
```

The character encoding of the HTML document was not declared. The document will render with garbled text in some browser configurations if the document contains characters from outside the US-ASCII range. The character encoding of the page must to be declared in the document or in the transfer protocol.

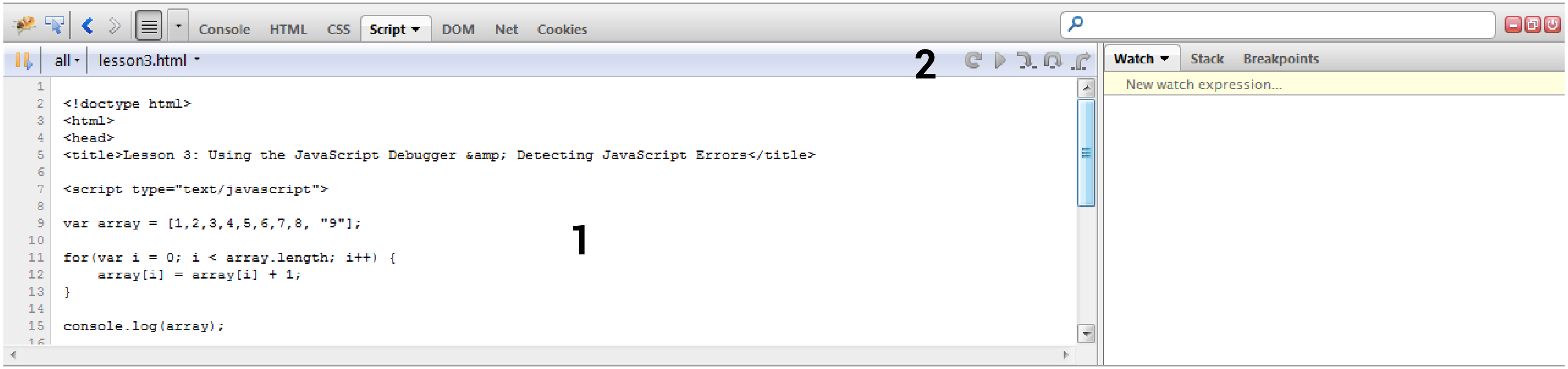
QUESTION: Why did the last element become 91?
We will analyze this by stepping through the code.

3) Click on Script



```

<!doctype html>
<html>
<head>
<title>Lesson 3: Using the JavaScript Debugger &amp; Detecting JavaScript Errors</title>

<script type="text/javascript">

var array = [1,2,3,4,5,6,7,8, "9"];

for(var i = 0; i < array.length; i++) {
    array[i] = array[i] + 1;
}

console.log(array);
```

Firebug - SCRIPT PANEL

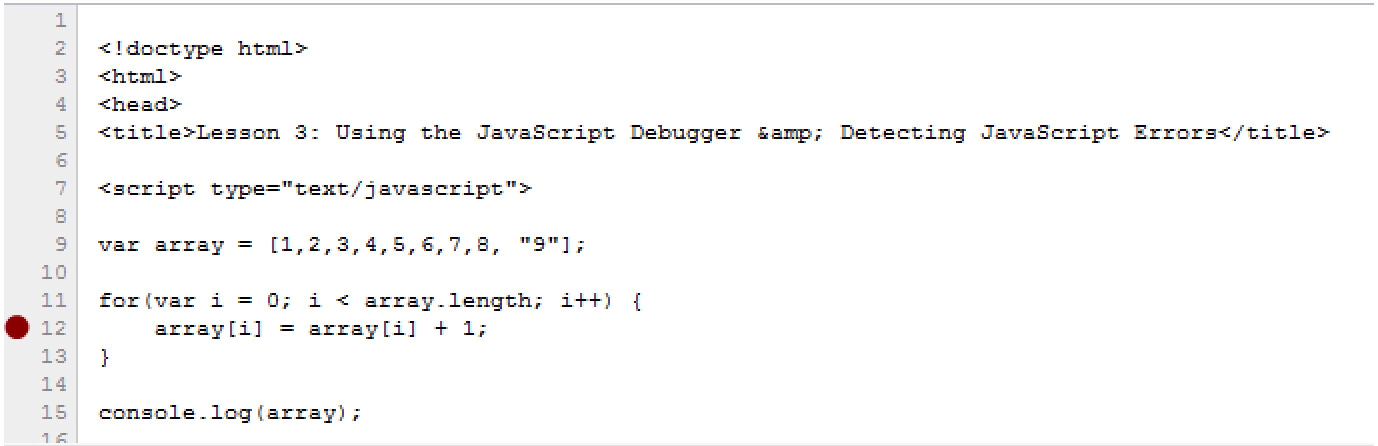


```

<!doctype html>
<html>
<head>
<title>Lesson 3: Using the JavaScript Debugger &amp; Detecting JavaScript Errors</title>

<script type="text/javascript">

var array = [1,2,3,4,5,6,7,8, "9"];

for(var i = 0; i < array.length; i++) {
    array[i] = array[i] + 1;
}

console.log(array);
```

1. The Script Panel View
2. Actions when JavaScript has hit a breakpoint
   1. Rerun – Shift + F8
   2. Continue – F8
   3. Step Into – F11
   4. Step Over – F10
   5. Step Out – Shift F11

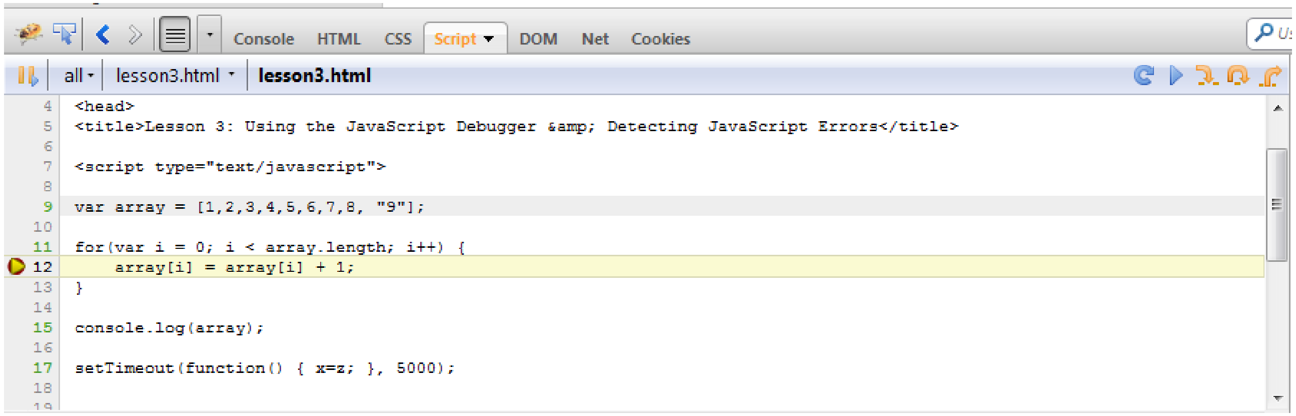
Stepping Through JavaScript

4) Create a breakpoint on line number 12 to analyze the problem. **Refresh the page**. (To create a breakpoint, click on the line number)



```
 1
 2  <!doctype html>
 3  <html>
 4  <head>
 5  <title>Lesson 3: Using the JavaScript Debugger &amp; Detecting JavaScript Errors</title>
 6
 7  <script type="text/javascript">
 8
 9  var array = [1,2,3,4,5,6,7,8, "9"];
10
11  for(var i = 0; i < array.length; i++) {
12      array[i] = array[i] + 1;
13  }
14
15  console.log(array);
```

5) JavaScript has stopped on the breakpoint.

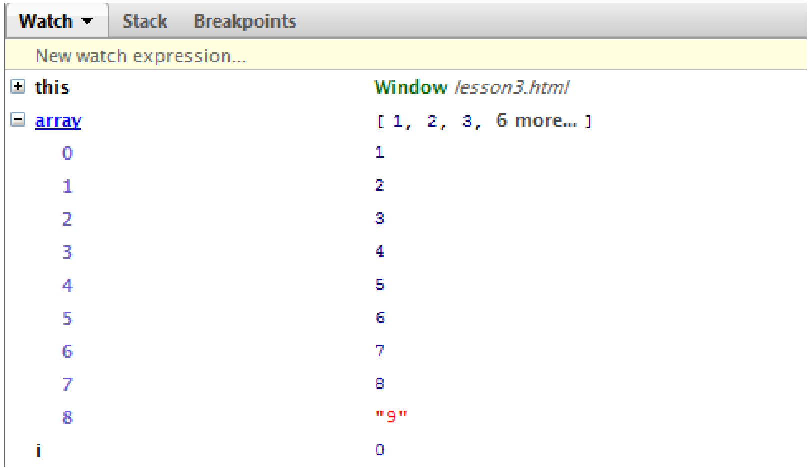


```
 4  <head>
 5  <title>Lesson 3: Using the JavaScript Debugger &amp; Detecting JavaScript Errors</title>
 6
 7  <script type="text/javascript">
 8
 9  var array = [1,2,3,4,5,6,7,8, "9"];
10
11  for(var i = 0; i < array.length; i++) {
12      array[i] = array[i] + 1;
13  }
14
15  console.log(array);
16
17  setTimeout(function() { x=z; }, 5000);
18
```

Notice: The page is still loading because the body element is not loaded.

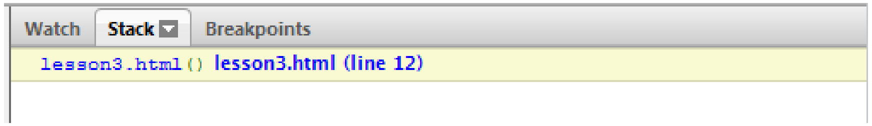
When Firebug Hits a Breakpoint, what else can we see?

Global elements & local elements
(click on Watch)



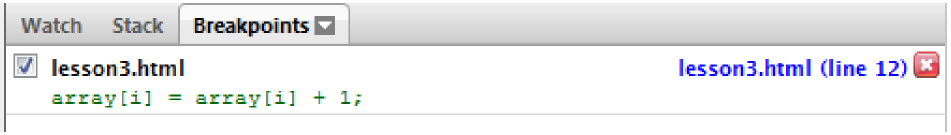
this	Window *lesson3.html*
array	[1, 2, 3, 6 more...]
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	"9"
i	0

The stack (useful for recursive functions)
(click on Stack)



```
lesson3.html() lesson3.html (line 12)
```

Breakpoints
(click on Breakpoints)



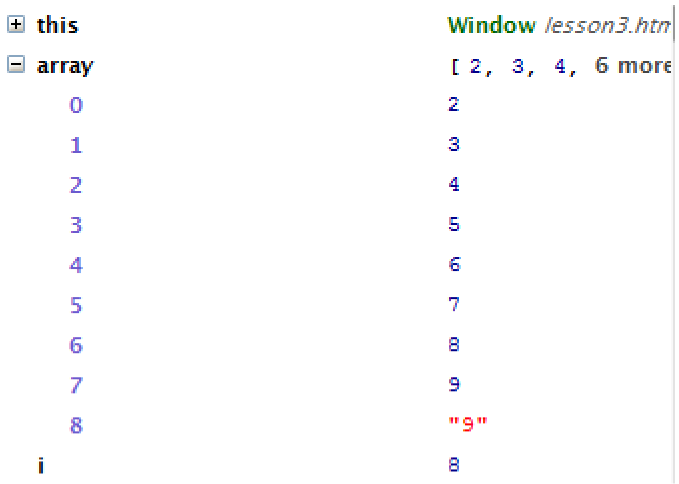
☑ lesson3.html lesson3.html (line 12)

```
array[i] = array[i] + 1;
```

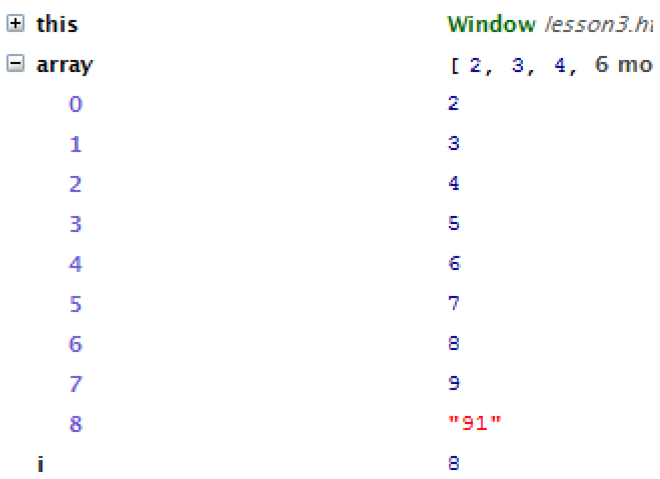
Analyzing the Array using the Watch Pane

4) Step through **8** times the dynamic execution trace by clicking on **Step Into…**

5) Step through once. The 8th element should be now “91”



```
⊞ this        Window lesson3.htm
⊟ array       [ 2, 3, 4, 6 more
    0         2
    1         3
    2         4
    3         5
    4         6
    5         7
    6         8
    7         9
    8         "9"
  i           8
```

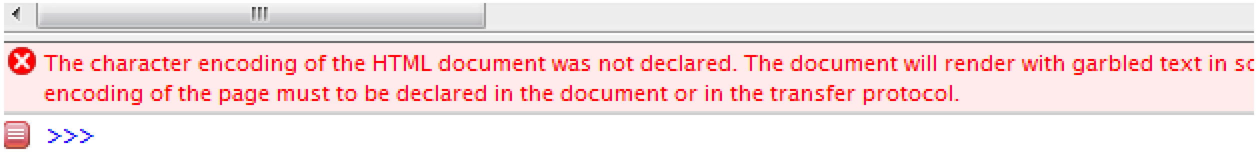


```
⊞ this        Window lesson3.h
⊟ array       [ 2, 3, 4, 6 mo
    0         2
    1         3
    2         4
    3         5
    4         6
    5         7
    6         8
    7         9
    8         "91"
  i           8
```

After pressing Step Into 8 times…

Stepping in once after 8

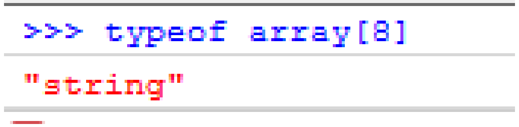
6) We can execute JavaScript code while JavaScript is still in the breakpoint. Hit



```
The character encoding of the HTML document was not declared. The document will render with garbled text in sc
encoding of the page must to be declared in the document or in the transfer protocol.
>>>
```

Analyzing the Array using the Watch Pane

6) Execute the following JavaScript. (after typing, hit enter)



```
>>> typeof array[8]
"string"
```

```
typeof array[8]
```

QUESTION: Why did the last element become 91?
It's because the last element is a string, hence it will do concatenation.

WHY IS STEPPING THROUGH JAVASCRIPT USEFUL?

1) It detects logic errors in JavaScript code
2) Useful when interpreting data via AJAX, as numbers may be interpreted as strings, and when we do number manipulation, it will cause a similar error as was described.

Speaking of errors, what does Firebug do when it executes the following illegal code?

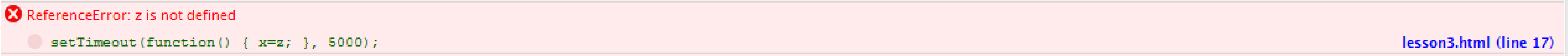
```
setTimeout(function() { x=z; console.log("Hello there!"); },
5000);
```

Errors in the Console

```
setTimeout(function() { x=z; console.log(“Hello there!”); },
5000);
```

How to detect JavaScript Syntax/Undefined errors the easy way:

1. Launch **Firebug**. Click on **Console** in the Firebug toolbar.
2. Due to the nature of the top code, the error will not appear until 5 seconds have passed. Then the following should show:



```
ReferenceError: z is not defined
setTimeout(function() { x=z; }, 5000);                    lesson3.html (line 17)
```

WHY IS THIS USEFUL?

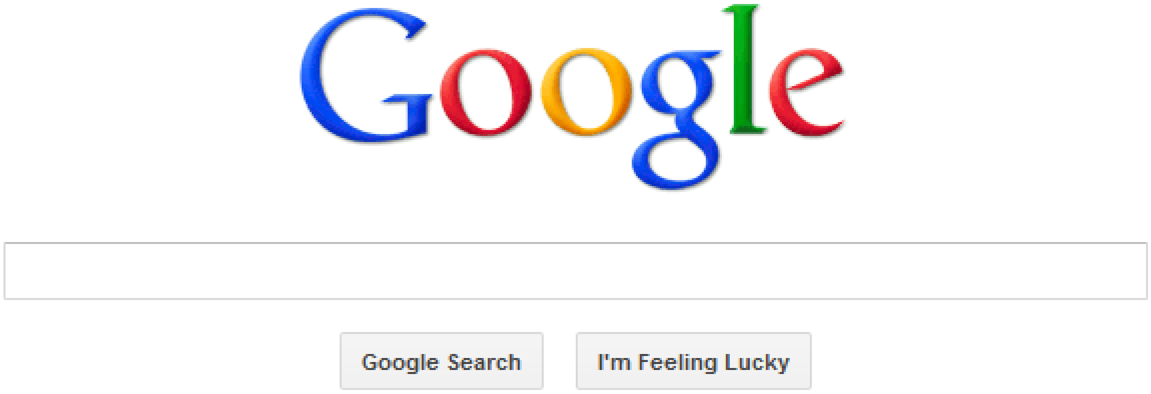
- By default, JavaScript code stops executing from the line an error occurs.
- If we didn't have Firebug, we would expect `Hello there`! to appear in the console, but it didn't, and we would debug manually by using alert() or document.write()

Detecting Web Performance Using Firebug

TUTORIAL OBJECTIVE

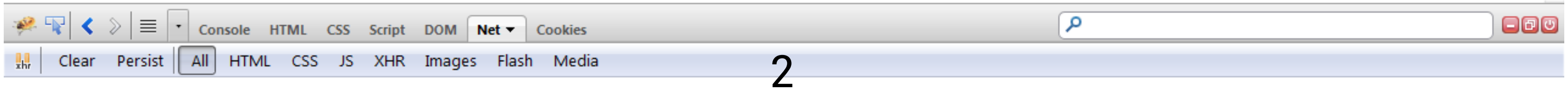
We will use http://www.google.com for this tutorial.

Determine what requests are made when going to http://www.google.com and how fast was each request.



Detecting Web Performance Using Firebug

1. Go to http://www.google.com
2. Launch Firebug and click on **Net**



2

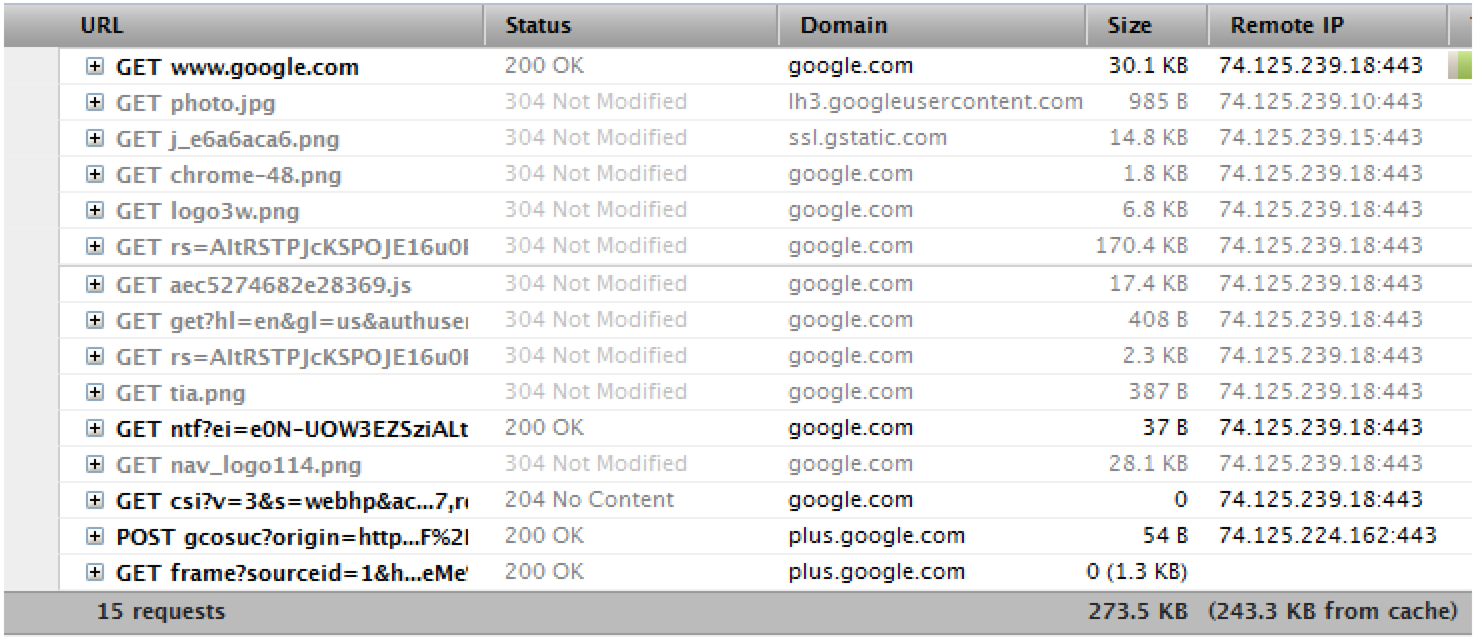
1

1. **The requests made**
2. **Filtering**

3. **Refresh the page.**

USC Viterbi School of Engineering

Detecting Web Performance Using Firebug



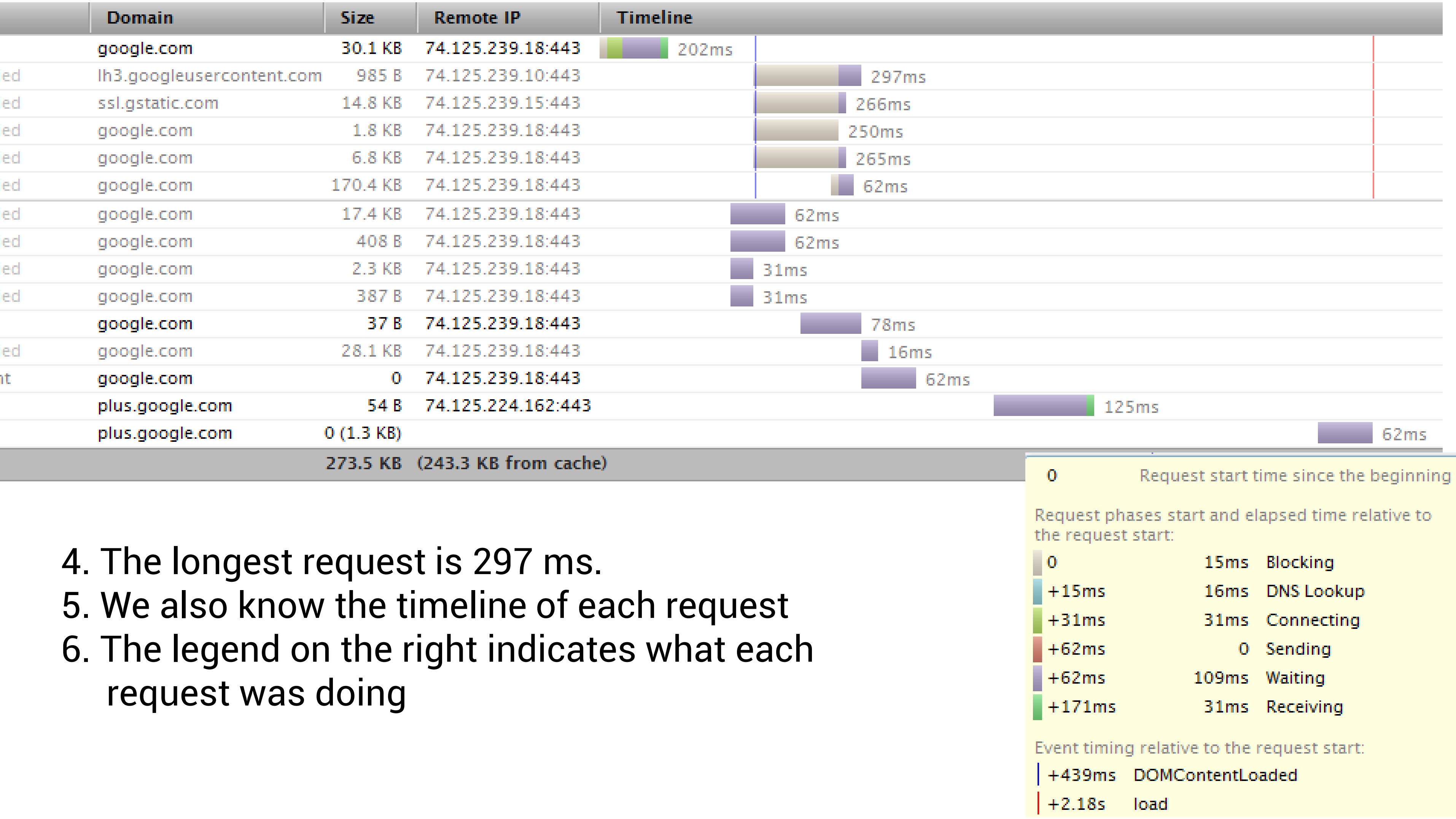
URL	Status	Domain	Size	Remote IP
⊞ GET www.google.com	200 OK	google.com	30.1 KB	74.125.239.18:443
⊞ GET photo.jpg	304 Not Modified	lh3.googleusercontent.com	985 B	74.125.239.10:443
⊞ GET j_e6a6aca6.png	304 Not Modified	ssl.gstatic.com	14.8 KB	74.125.239.15:443
⊞ GET chrome-48.png	304 Not Modified	google.com	1.8 KB	74.125.239.18:443
⊞ GET logo3w.png	304 Not Modified	google.com	6.8 KB	74.125.239.18:443
⊞ GET rs=AItRSTPJcKSPOJE16u0I	304 Not Modified	google.com	170.4 KB	74.125.239.18:443
⊞ GET aec5274682e28369.js	304 Not Modified	google.com	17.4 KB	74.125.239.18:443
⊞ GET get?hl=en&gl=us&authuse	304 Not Modified	google.com	408 B	74.125.239.18:443
⊞ GET rs=AItRSTPJcKSPOJE16u0I	304 Not Modified	google.com	2.3 KB	74.125.239.18:443
⊞ GET tia.png	304 Not Modified	google.com	387 B	74.125.239.18:443
⊞ GET ntf?ei=e0N-UOW3EZSziALt	200 OK	google.com	37 B	74.125.239.18:443
⊞ GET nav_logo114.png	304 Not Modified	google.com	28.1 KB	74.125.239.18:443
⊞ GET csi?v=3&s=webhp&ac...7,r	204 No Content	google.com	0	74.125.239.18:443
⊞ POST gcosuc?origin=http...F%2I	200 OK	plus.google.com	54 B	74.125.224.162:443
⊞ GET frame?sourceid=1&h...eMe	200 OK	plus.google.com	0 (1.3 KB)	
15 requests			273.5 KB	(243.3 KB from cache)

After refreshing, something like the above should appear.

Here's what we can say about the performance of http://www.google.com:

1. There were 15 requests, 273.5 KB in total size, 243.3 KB from cache.
2. The remote IP is 74.125.239.18: 443 and 74.125.224.162:443 for one req.
3. The status of each request, and what type it was

Detecting Web Performance Using Firebug



	Domain	Size	Remote IP	Timeline
	google.com	30.1 KB	74.125.239.18:443	202ms
ed	lh3.googleusercontent.com	985 B	74.125.239.10:443	297ms
ed	ssl.gstatic.com	14.8 KB	74.125.239.15:443	266ms
ed	google.com	1.8 KB	74.125.239.18:443	250ms
ed	google.com	6.8 KB	74.125.239.18:443	265ms
ed	google.com	170.4 KB	74.125.239.18:443	62ms
ed	google.com	17.4 KB	74.125.239.18:443	62ms
ed	google.com	408 B	74.125.239.18:443	62ms
ed	google.com	2.3 KB	74.125.239.18:443	31ms
ed	google.com	387 B	74.125.239.18:443	31ms
	google.com	37 B	74.125.239.18:443	78ms
ed	google.com	28.1 KB	74.125.239.18:443	16ms
nt	google.com	0	74.125.239.18:443	62ms
	plus.google.com	54 B	74.125.224.162:443	125ms
	plus.google.com	0 (1.3 KB)		62ms
		273.5 KB	(243.3 KB from cache)	

0 Request start time since the beginning

Request phases start and elapsed time relative to the request start:

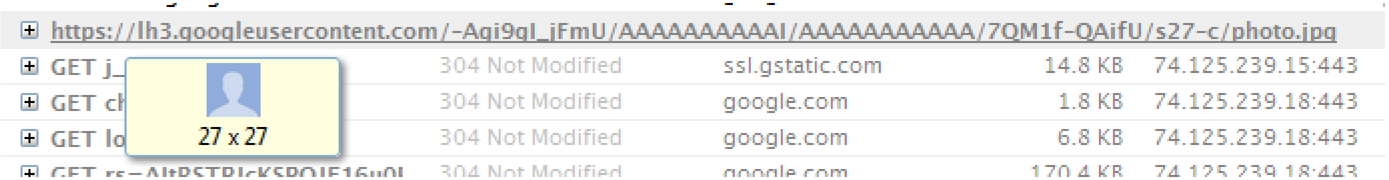
0	15ms	Blocking
+15ms	16ms	DNS Lookup
+31ms	31ms	Connecting
+62ms	0	Sending
+62ms	109ms	Waiting
+171ms	31ms	Receiving

Event timing relative to the request start:

+439ms	DOMContentLoaded
+2.18s	load

4. The longest request is 297 ms.
5. We also know the timeline of each request
6. The legend on the right indicates what each request was doing

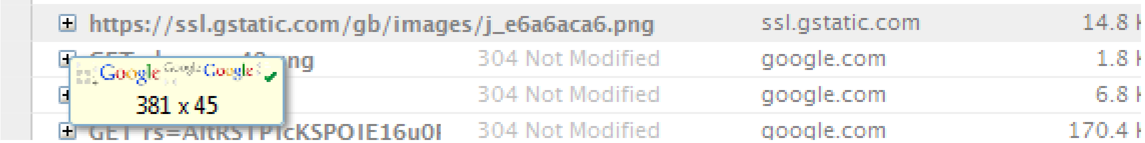
Analyzing a Request using Firebug



```
⊞ https://lh3.googleusercontent.com/-Agi9gI_jFmU/AAAAAAAAAAI/AAAAAAAAAAA/7QM1f-QAifU/s27-c/photo.jpg
⊞ GET j_         304 Not Modified   ssl.gstatic.com   14.8 KB   74.125.239.15:443
⊞ GET ch         304 Not Modified   google.com         1.8 KB   74.125.239.18:443
⊞ GET lo         304 Not Modified   google.com         6.8 KB   74.125.239.18:443
⊞ GET rs=AItRSTPIcKSPOIE16u0I   304 Not Modified   google.com   170.4 KB   74.125.239.18:443
```

TIP: Hovering over an image will show an info box of the image

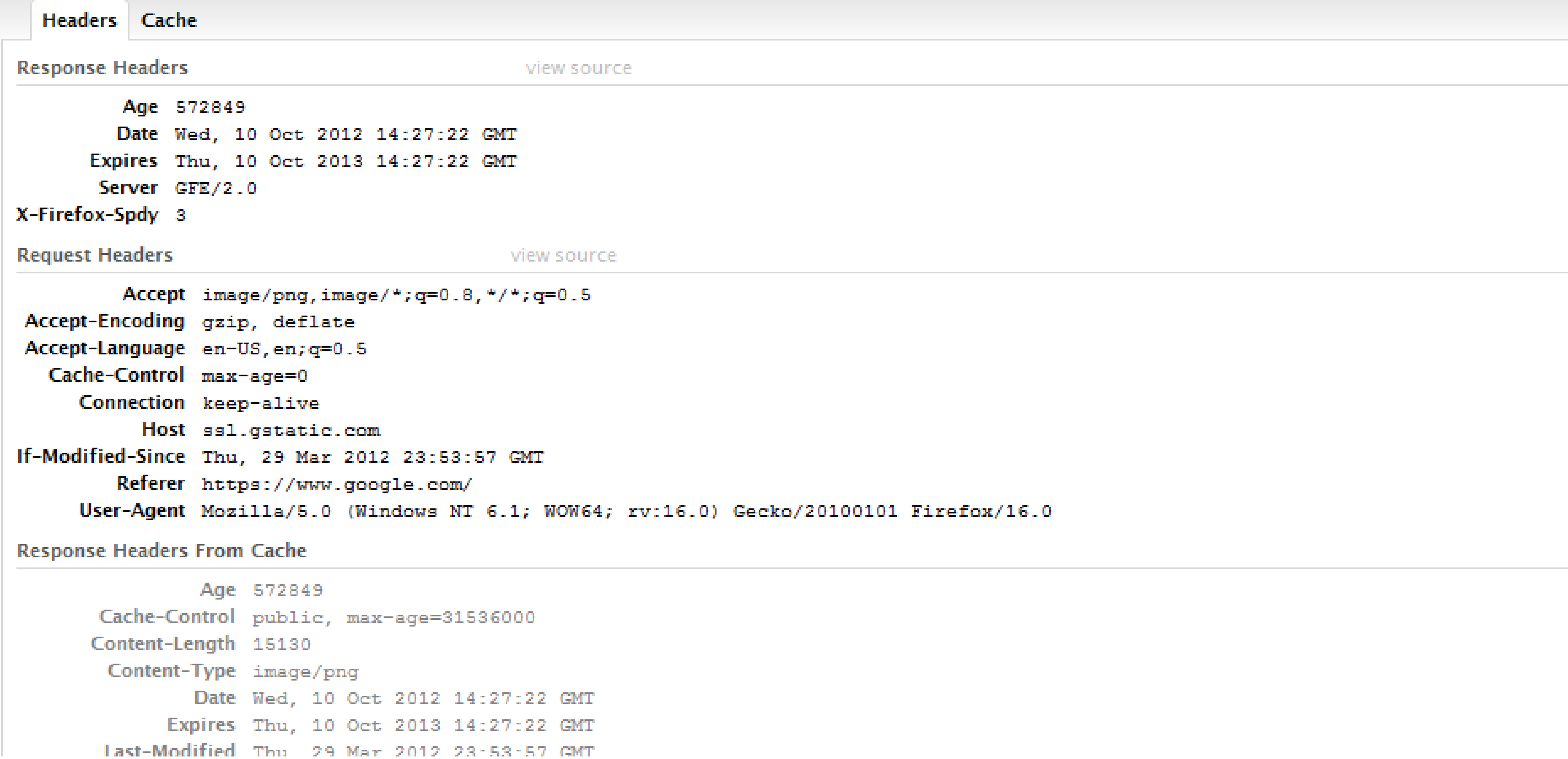
1) Expand the request by clicking on +



```
⊞ https://ssl.gstatic.com/gb/images/j_e6a6aca6.png         ssl.gstatic.com   14.8 I
⊞ GET ng                      304 Not Modified   google.com         1.8 I
⊞                             304 Not Modified   google.com         6.8 I
⊞ GET rs=AItRSTPIcKSPOIE16u0I 304 Not Modified   google.com       170.4 I
```

Analyzing a Request using Firebug

2) We can see the details of the request in the Headers Tab



Response Headers view source

```
          Age  572849
         Date  Wed, 10 Oct 2012 14:27:22 GMT
      Expires  Thu, 10 Oct 2013 14:27:22 GMT
       Server  GFE/2.0
X-Firefox-Spdy 3
```

Request Headers view source

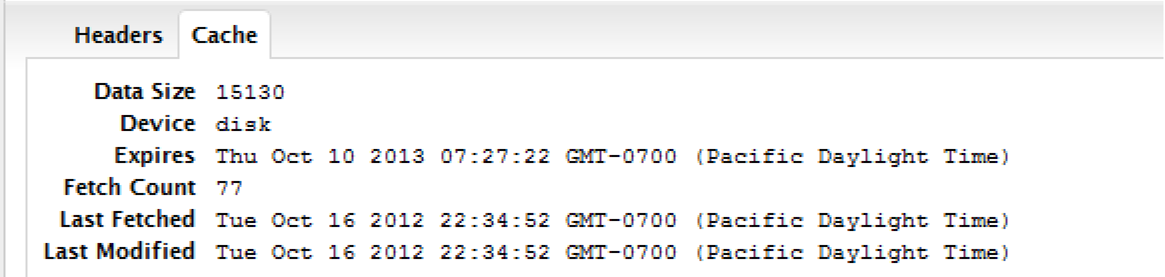
```
           Accept  image/png,image/*;q=0.8,*/*;q=0.5
  Accept-Encoding  gzip, deflate
  Accept-Language  en-US,en;q=0.5
    Cache-Control  max-age=0
       Connection  keep-alive
             Host  ssl.gstatic.com
If-Modified-Since  Thu, 29 Mar 2012 23:53:57 GMT
          Referer  https://www.google.com/
       User-Agent  Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0) Gecko/20100101 Firefox/16.0
```

Response Headers From Cache

```
           Age  572849
 Cache-Control  public, max-age=31536000
Content-Length  15130
  Content-Type  image/png
          Date  Wed, 10 Oct 2012 14:27:22 GMT
       Expires  Thu, 10 Oct 2013 14:27:22 GMT
 Last-Modified  Thu, 29 Mar 2012 23:53:57 GMT
```

Analyzing a Request using Firebug

3) We can see what was loaded in the cache from the cache tab



```
    Data Size  15130
       Device  disk
      Expires  Thu Oct 10 2013 07:27:22 GMT-0700 (Pacific Daylight Time)
  Fetch Count  77
 Last Fetched  Tue Oct 16 2012 22:34:52 GMT-0700 (Pacific Daylight Time)
Last Modified  Tue Oct 16 2012 22:34:52 GMT-0700 (Pacific Daylight Time)
```

WHY IS ANALYZING PERFORMANCE WITH FIREBUG USEFUL?

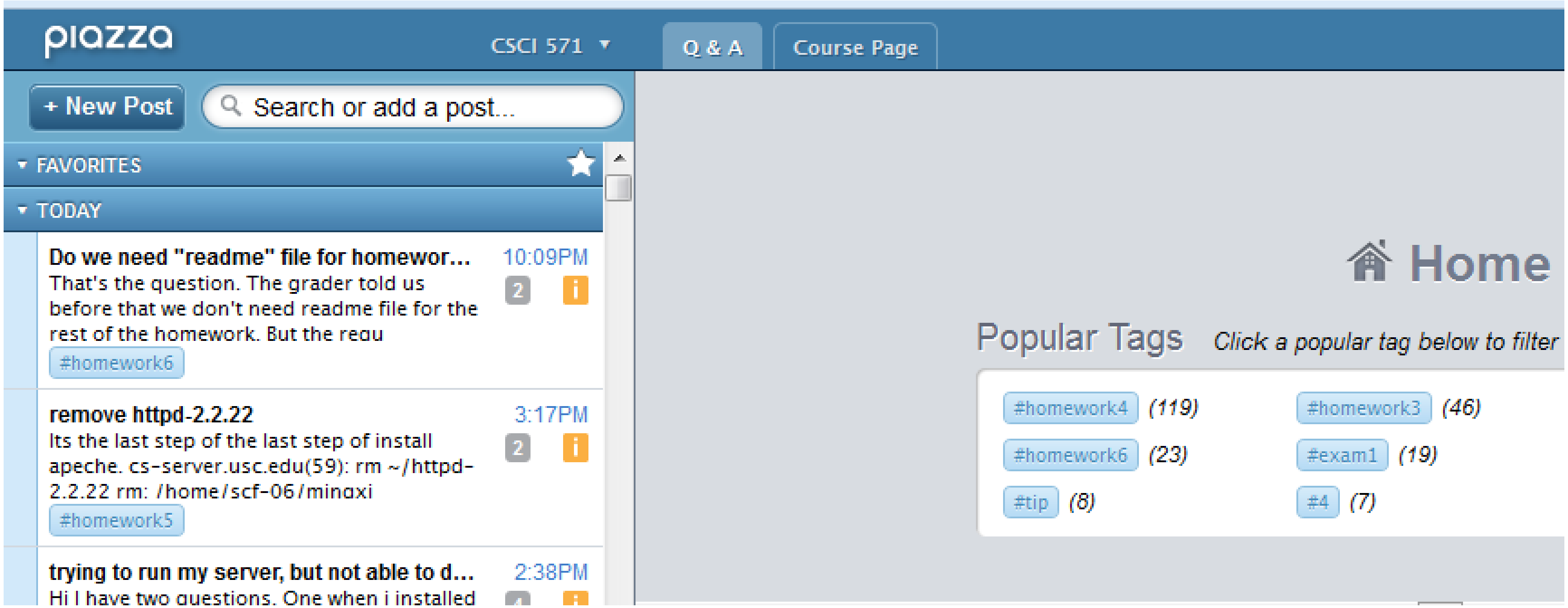
We can detect how large a request is, and where the website is really slow
- which is how we can improve performance for a site

Tracking Cookies and Sessions with Firebug

TUTORIAL OBJECTIVE

Track the cookies and sessions created by a website.

1) We will use http://www.piazza.com for this example.



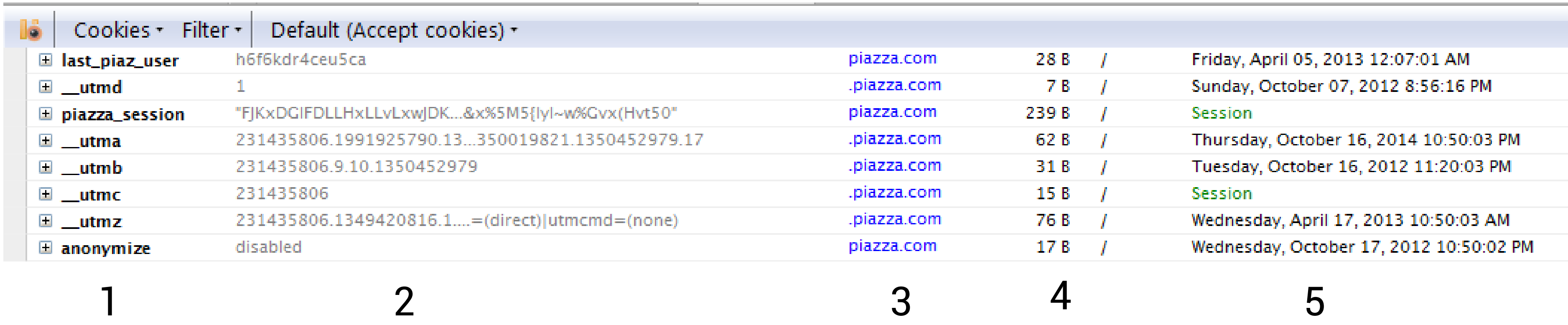
Do we need "readme" file for homewor... 10:09PM
That's the question. The grader told us before that we don't need readme file for the rest of the homework. But the requ
#homework6

remove httpd-2.2.22 3:17PM
Its the last step of the last step of install apeche. cs-server.usc.edu(59): rm ~/httpd-2.2.22 rm: /home/scf-06/minaxi
#homework5

trying to run my server, but not able to d... 2:38PM
Hi I have two questions. One when i installed

2) Launch Firebug and click on the cookies panel.

Tracking Cookies and Sessions with Firebug



⊞ last_piaz_user	h6f6kdr4ceu5ca	piazza.com	28 B	/	Friday, April 05, 2013 12:07:01 AM
⊞ __utmd	1	.piazza.com	7 B	/	Sunday, October 07, 2012 8:56:16 PM
⊞ piazza_session	"FJKxDGIFDLLHxLLvLxwJDK...&x%5M5{lyl~w%Gvx(Hvt50"	piazza.com	239 B	/	Session
⊞ __utma	231435806.1991925790.13...350019821.1350452979.17	.piazza.com	62 B	/	Thursday, October 16, 2014 10:50:03 PM
⊞ __utmb	231435806.9.10.1350452979	.piazza.com	31 B	/	Tuesday, October 16, 2012 11:20:03 PM
⊞ __utmc	231435806	.piazza.com	15 B	/	Session
⊞ __utmz	231435806.1349420816.1....=(direct)|utmcmd=(none)	.piazza.com	76 B	/	Wednesday, April 17, 2013 10:50:03 AM
⊞ anonymize	disabled	piazza.com	17 B	/	Wednesday, October 17, 2012 10:50:02 PM

1. Name and Expand button.

Expanding the cookie only reveals the full value of the value attribute.

2. Value of the cookie
3. Domain of the cookie
4. The size
5. The path
6. When it expires / or whether it is a session

Sources

- **Smashing Magazine - "15 Helpful In-Browser Web Development Tools"** http://www.smashingmagazine.com/2008/11/18/15-helpful-in-browser-web-development-tools/ (Used for Introduction)
- **SitePoint – "Firebug CSS Active Hover States" -** http://www.sitepoint.com/firebug-css-active-hover-states/ - Tip for :hover
- **Firebug** - https://getfirebug.com/ (Images, and Firebug Console)

- Tutorials **were made from scratch**, based on my **real-life** scenarios as a web developer:
  - Tutorial 1: **Client:** I really like **this button** from http://www.google.com . I want **that same button** on my website.
  - Tutorial 2: **Client:** Can we compare how the website looks with white text and black background, black background & white text?
  - Tutorial 3: When I was developing a canvas application for USC Viterbi, I was adding numbers like the following: **2 + 2**. I expected 4, but the result turned out to be **22**.
  - Tutorial 4: When I had the lecture in CSCI 571 about Web Performance, I played with Firebug
  - Tutorial 5: Internship made me deal with cookies/sessions