

CT1114

Web Development

HTML, CSS, JavaScript

Javascript: `window.confirm()`, Parse functions, Debugging tools in Chrome and Firefox

Dr. Sam Redfern



OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Last Week's Exercise

- Create a web page which displays an image (using absolute referencing) from the following URL:
<http://www.psychicsoftware.com/ct1113/banner-image-1.png>
- When the image is clicked, you should (with Javascript) change its **src** attribute to:
<http://www.psychicsoftware.com/ct1113/banner-image-2.png>
- On each subsequent click, display the next image (banner-image-3.png, banner-image-4.png, banner-image-5.png, banner-image-6.png)
- After image 6, revert to image 1

The Window Object

- Represents an open window in a browser
- **window** is an object just like **document**
- Objects have associated **attributes** (pieces of data) and **methods** (things they can do)
- Some important window methods:
 - alert (covered last week)
 - prompt (covered last week)
 - confirm (see below)
 - setTimeout (see later)

The Confirm Method

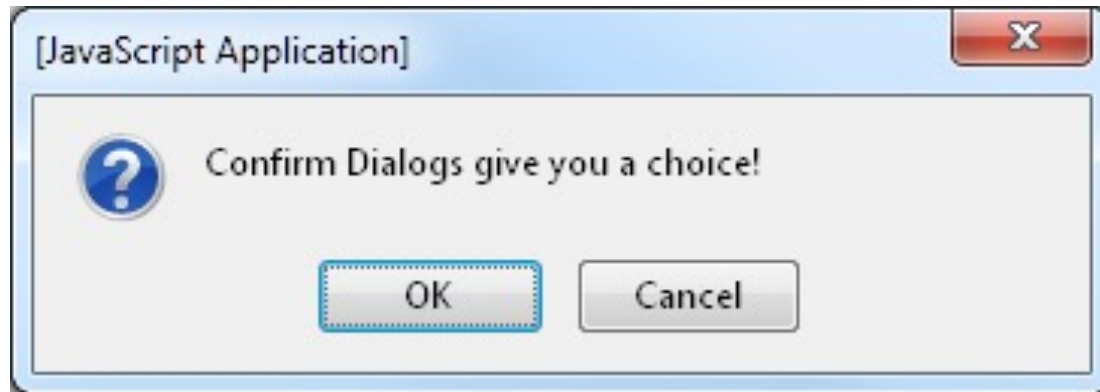
- Used if you want the user to verify or accept something
- User presented with the choice of clicking **OK** or **Cancel** to proceed
- If the user pressed OK
 - **true** is returned
- If the user pressed Cancel
 - **false** is returned

The Confirm Method

- Syntax:

```
var choice;
```

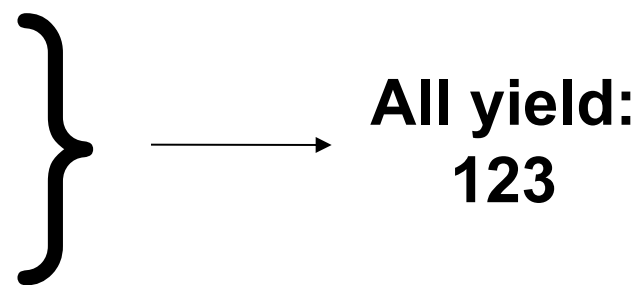
```
choice=confirm("Confirm Dialogs give you a choice!");
```



Confirm Example

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function GiveChoice() {
        if (confirm("Make the document red?")==true)
          document.body.style.background = "red";
        else {
          alert("Right so, it'll be green.");
          document.body.style.background = "green";
        }
      }
    </script>
  </head>
  <body onload="GiveChoice();" style="background:blue;">
</body>
</html>
```

Parse Functions

- `parseInt(string)`
 - Converts the string to an Integer and returns it
 - `parseFloat(string)`
 - Converts the string to a Floating Point number
 - String argument must **not** be empty
 - String argument must **start** with a valid number
 - `parseInt("123");`
 - `parseInt("123abc");`
 - `parseInt("123abc456");`
- 
- The diagram illustrates that the `parseInt` function extracts the leading integer from a string. It shows three function calls: `parseInt("123");`, `parseInt("123abc");`, and `parseInt("123abc456");`. These are grouped by a large right-facing curly brace. An arrow points from the brace to the text "All yield: 123", indicating that all three calls return the same integer value.
- All yield:
123**

Parse Functions - isNaN() - *"Is Not a Number"*

```
var number = parseInt("abc");  
alert("The number is: " + number + " and  
    multiplied by two is " + (number*2));
```

Output: *The number is NaN and multiplied by two is NaN*

- If string is not a number, it is set to a special **NaN** value
- **isNaN(value)** checks if a value is a number
- Returns **true** if value is Not a Number
- Returns **false** if value is a Number

isNaN() function in use

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function test() {
        var str = prompt("Please type a number");
        if (isNaN(str)) {
          alert("That's not a number");
        }
        else {
          var number = parseInt(str);
          alert("The number is: " + number + " and multiplied
by two is " + (number*2));
        }
      }
    </script>
  </head>
  <body>
    <button onclick='test();'>Click Me!</button>
  </body>
</html>
```

Exercise

- Make a webpage which, **as soon as it loads**, prompts the user for an integer between 1 and 100, using the `prompt()` method
- If the input given cannot convert to a number (e.g. if they type their name), or if it converts to a number that's not in the correct range, tell them the error. Otherwise, thank them.

Debugging Tools

Firefox

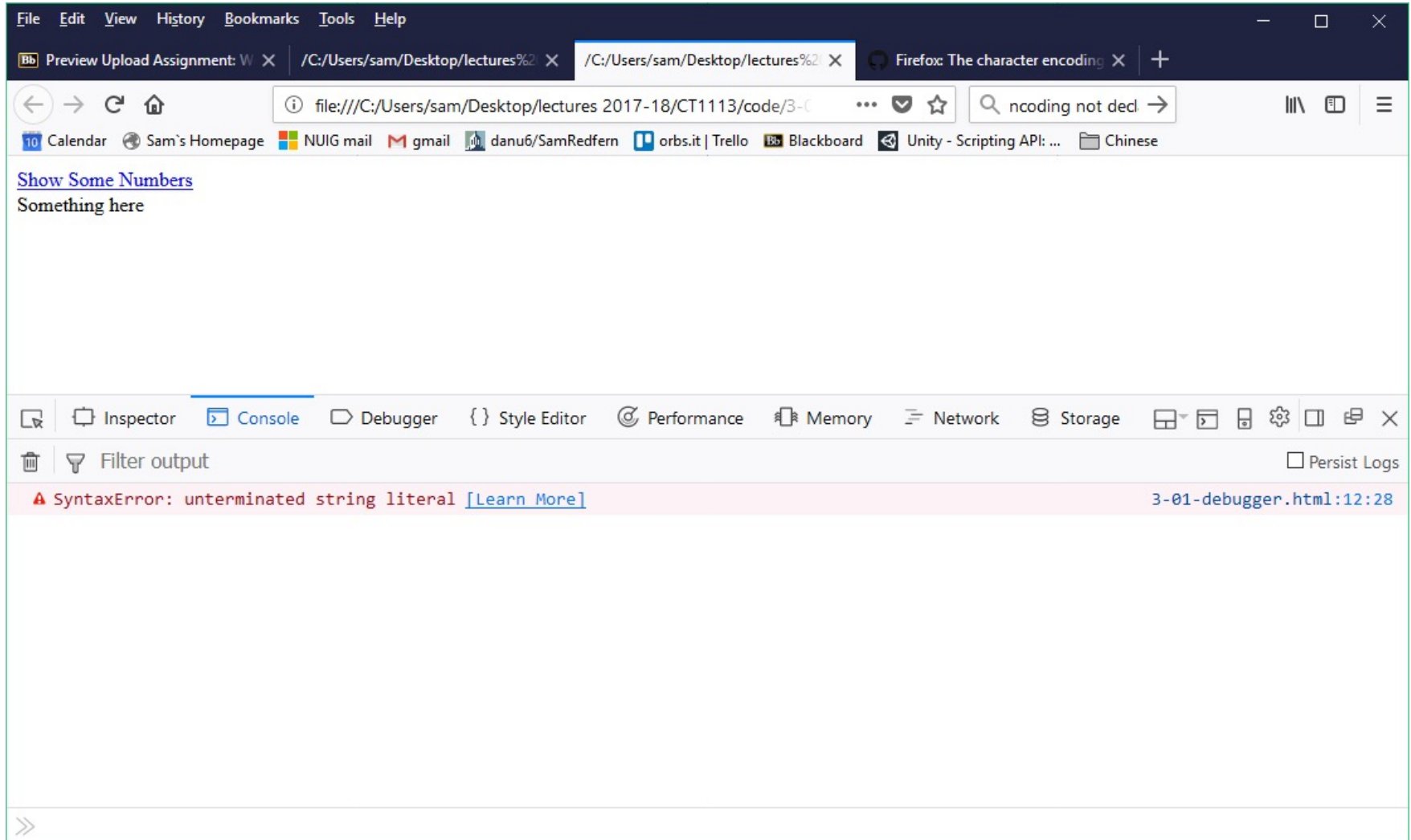
Chrome

Consider this HTML+Javascript..

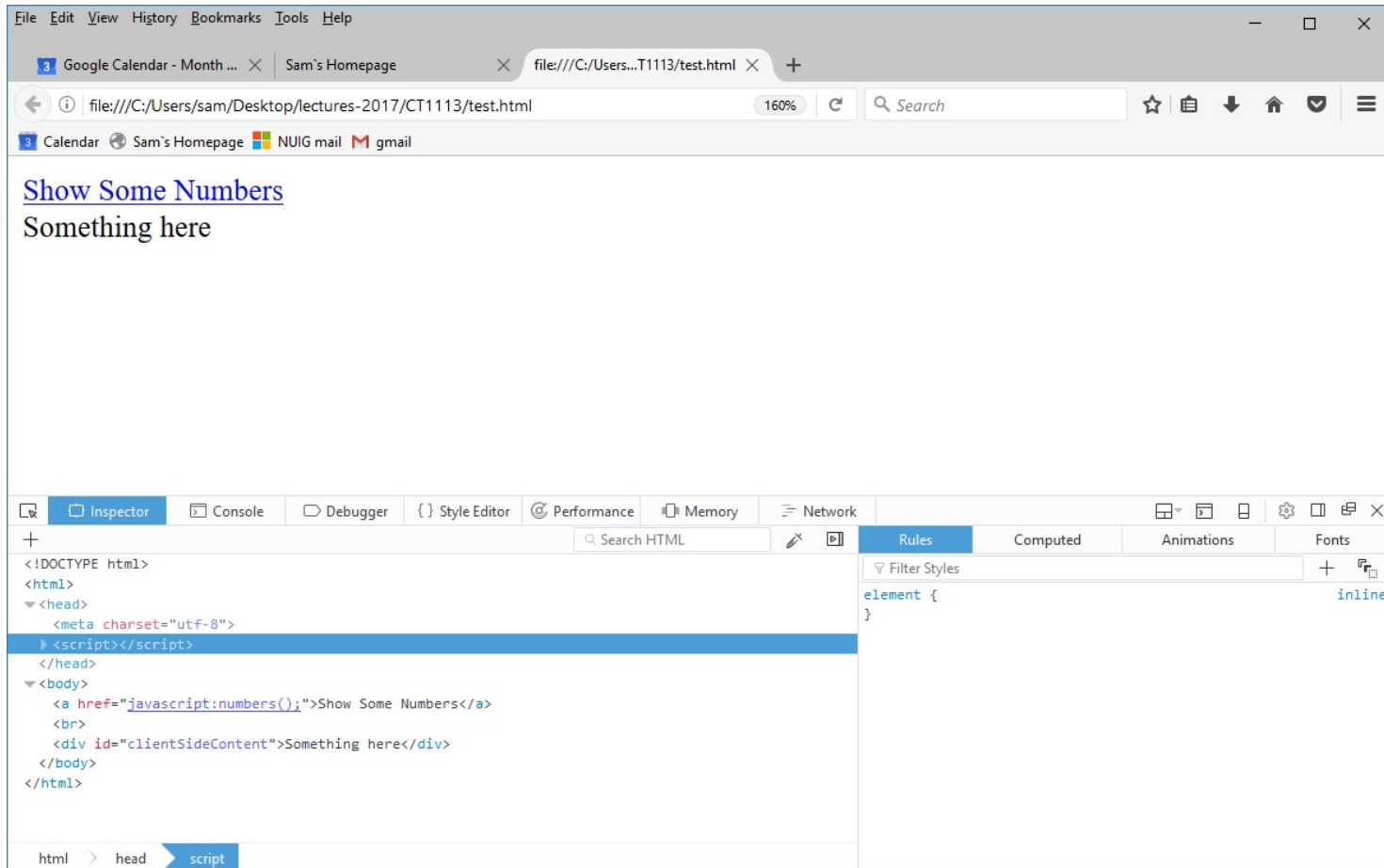
```
<!DOCTYPE html>
<html>
<meta charset="utf-8"/>
<head>
<script>
  function numbers() {
    var sHTML = "<p>";
    for (var i=1; i<=20; i++) {
      sHTML += Math.random() + "<br>";
    }
    sHTML += "</p>";
    document.getElementById("myPara").innerHTML = sHTML;
    console.log("Final HTML generated in loop: "+sHTML);
  }
</script>
</head><body>
  <button onclick='numbers();'>Show Some Random Numbers</button>
  <p id='myPara'>Something here</p>
</body>
</html>
```

Does it work? If not, why not?
The Web Console might help

Web Console (Firefox)

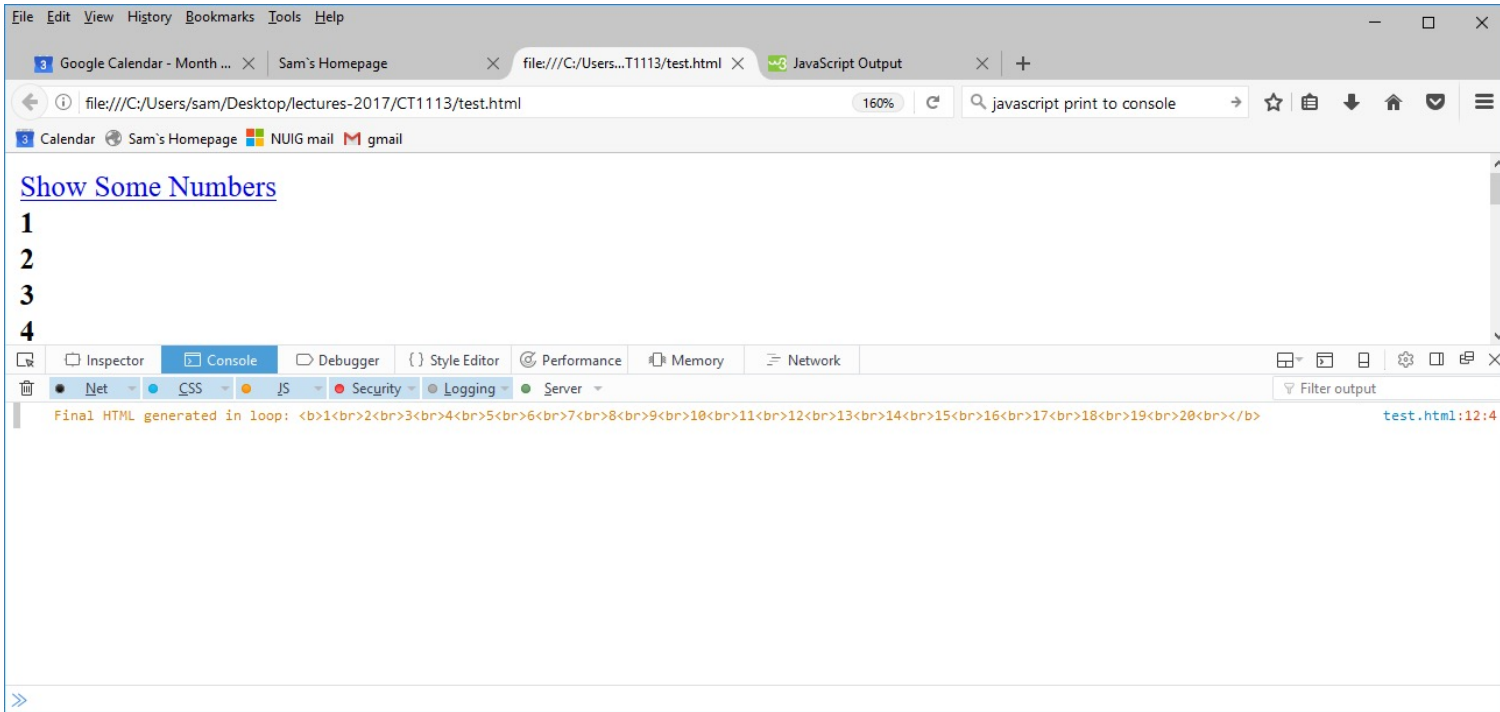


Firefox: Inspector



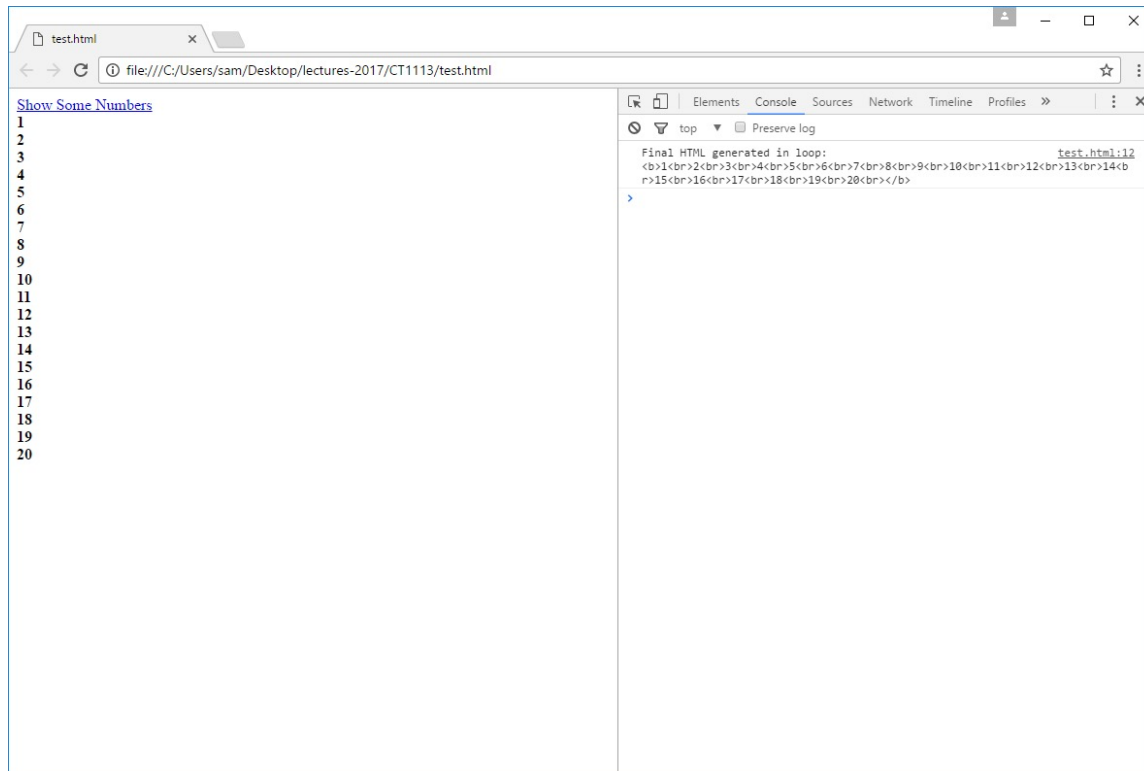
- Browse the live Document Object Model (DOM)
- Even make changes!

Firefox: Web Console

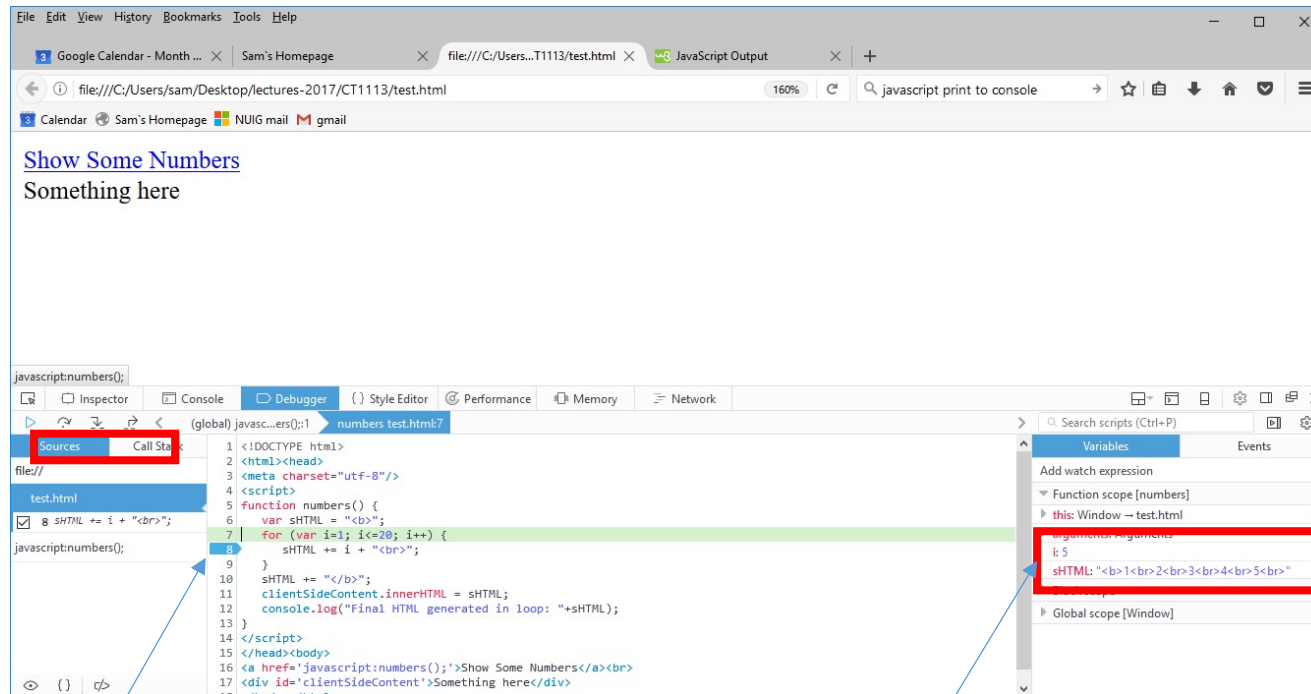


- Use `console.log(value);` to write data to the web console, to see the sequence/flow-of-control and values of variables during execution

Chrome: Console = Firefox: Web Console



Firefox: Debugger

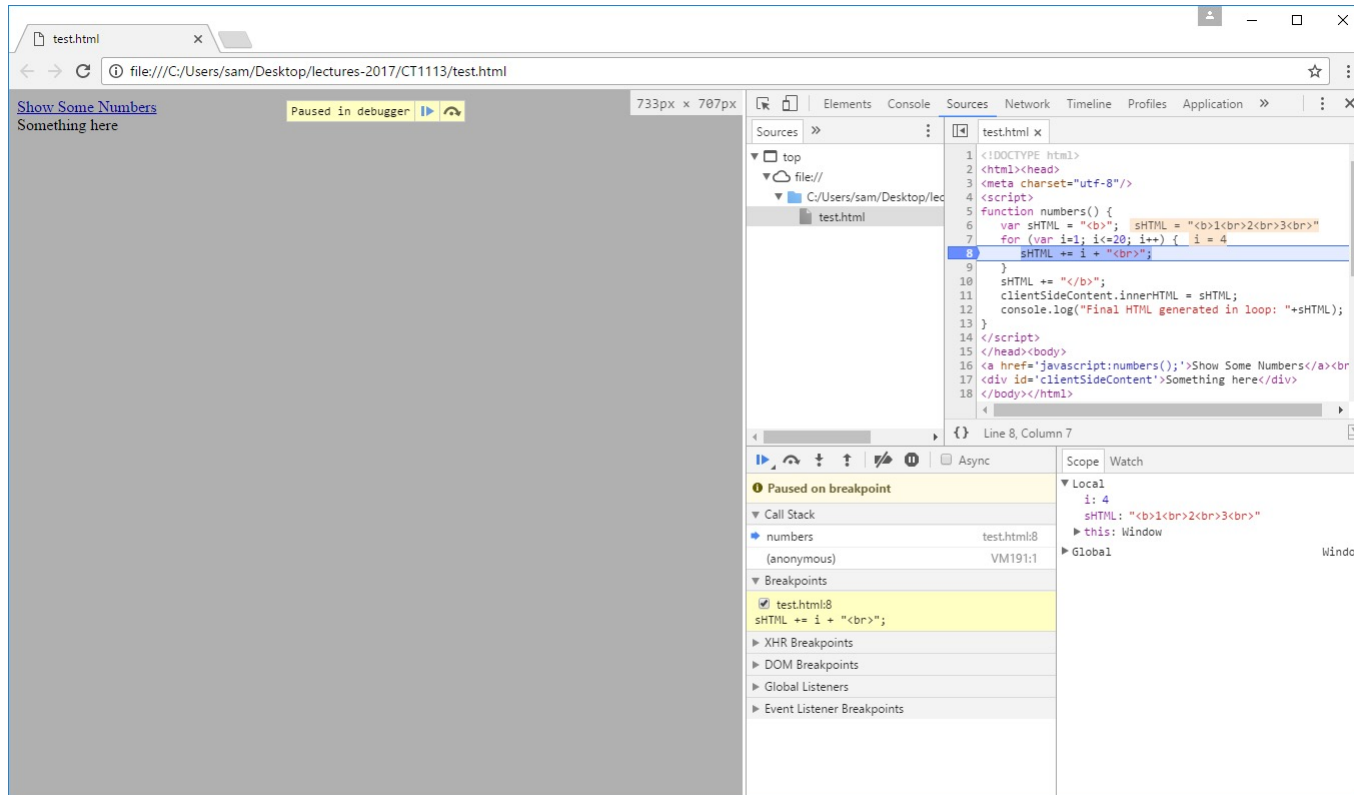


Resume
Step Over
Step In
Step Out

Rightclick, Add Breakpoint
(or leftclick in left margin)

Observe variables as you step line-by-line

Chrome: Sources = Firefox: Debugger



Debugging Exercise

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function isPrime(n) {
        for (var i=2; i<=n/2; i++) {
          if (n%i==0)
            return false;
        }
        return true;
      }

      function showPrimes() {
        var output = "";
        var num = 0;
        for (var i=2; i<5000; i++) {
          if (isPrime(i)) {
            num++;
            output += i;
            if (num%10==0)
              output += "<br>";
            else
              output += " &nbsp;";
          }
        }
        document.getElementById("pOutput").innerHTML = output;
      }
    </script>
  </head>
  <body onload="showPrimes();" >
    <p>Here's some prime numbers:</p>
    <p id='pOutput'></p>
  </body>
</html>
```

The Javascript on this webpage has 3 errors.

Fix this code so that prime numbers are displayed

Exercise (not graded)

- Building on the previous exercise where you obtained a number between 1-100 from the user, and checked its validity
- If the input from the user *does* convert to a number in the correct range (1-100), produce output in the webpage which counts from 1 up to their number, with a line-break `
` between each number
- Hint: use a 'for' loop to build up your output string and assign it into the innerHTML of a `` or `<p>`