Lecture 13

408/508 Computational Techniques for Linguists

Today's Topics

- Javascript console
  - interactive: *a bit like the* Terminal/Bash shell
- Introduction to Javascript, the browser programming language
- Tic-tac-toe example
- Files: `sample.html`, `sample2.html` and `sample3.html`
  - *on course webpage*

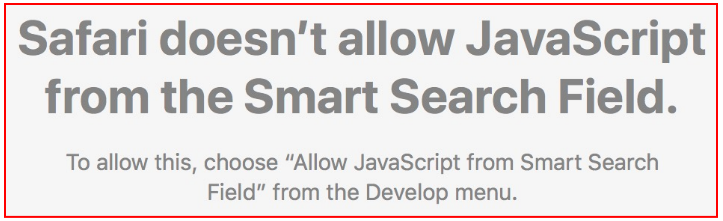
Javascript (JS)

- Javascript:
  - invented in 1995 as a browser programming language
  - *not the same as Java* (also appeared in 1995)
- Javascript code goes in the .html file:
  - `<script> ..   </script>`
  - `<script src="filename.js"></script>`
- HTML element events:
  - `onclick="js code"`
  - `onmouseover="js code"`
  - `onkeydown="js code"`
- Reference:
  - http://www.w3schools.com/jsref/dom_obj_event.asp

code snippets can be placed in
`<head>..</head>`
or
`<body>..</body>`

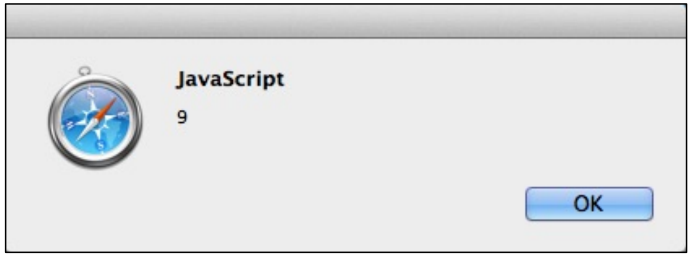
Javascript pop-up notification

- For browser programming:
  - alert(*string*)
  - *no print function*
- Example (from address bar):
  - instead of `http:` or `https:`
  - `javascript:alert(4+5)`
  - produces a pop-up



Safari doesn’t allow JavaScript from the Smart Search Field.

To allow this, choose “Allow JavaScript from Smart Search Field” from the Develop menu.

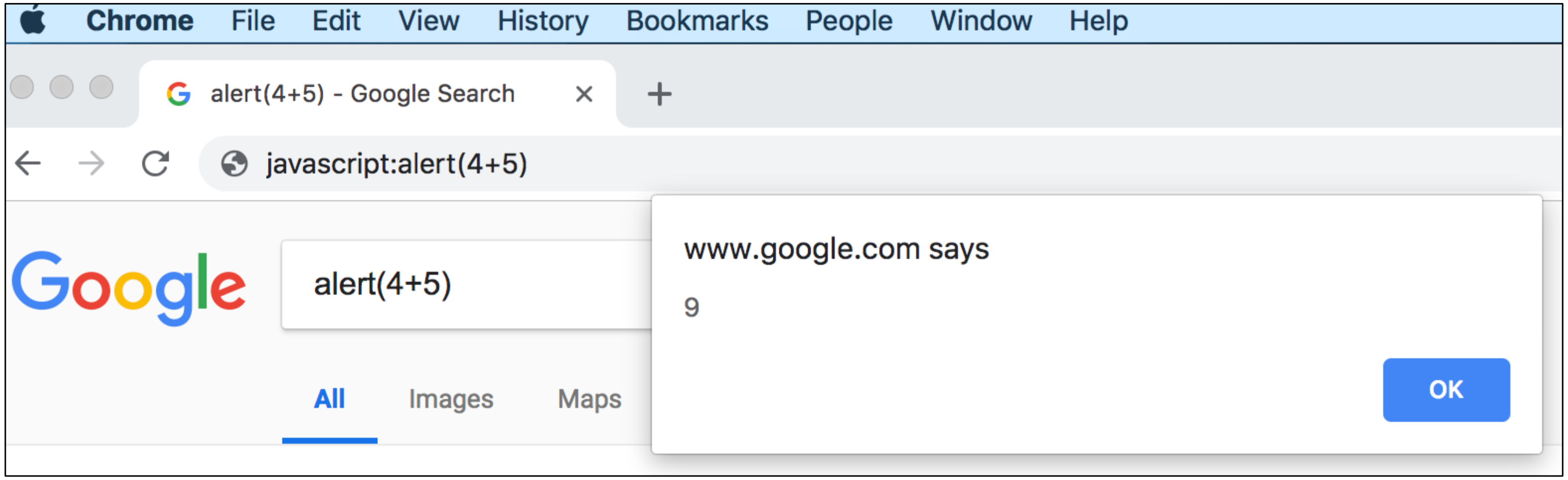


JavaScript

9

- Changing the html document using Javascript:
  - document.write(*string*) *overwrites html document*
  - *html_element*.innerHTML = "*string*" *modifies html_element*

Javascript pop-up notification



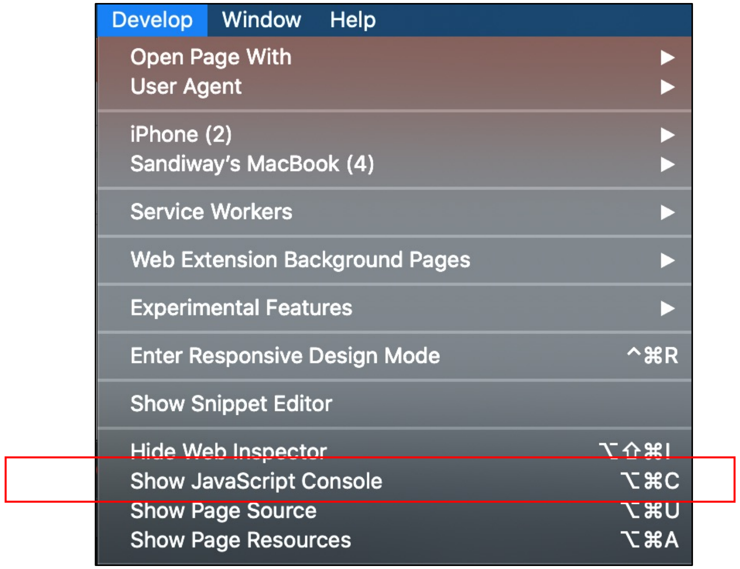
javascript:alert(4+5)

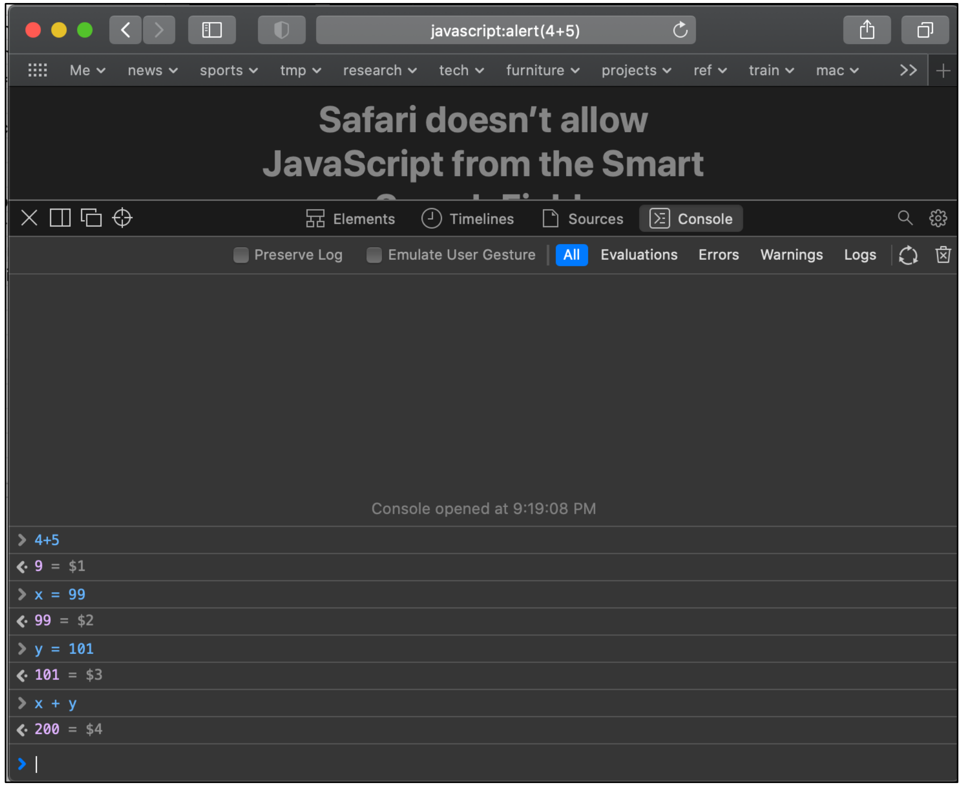
alert(4+5)

www.google.com says

9

Javascript Console

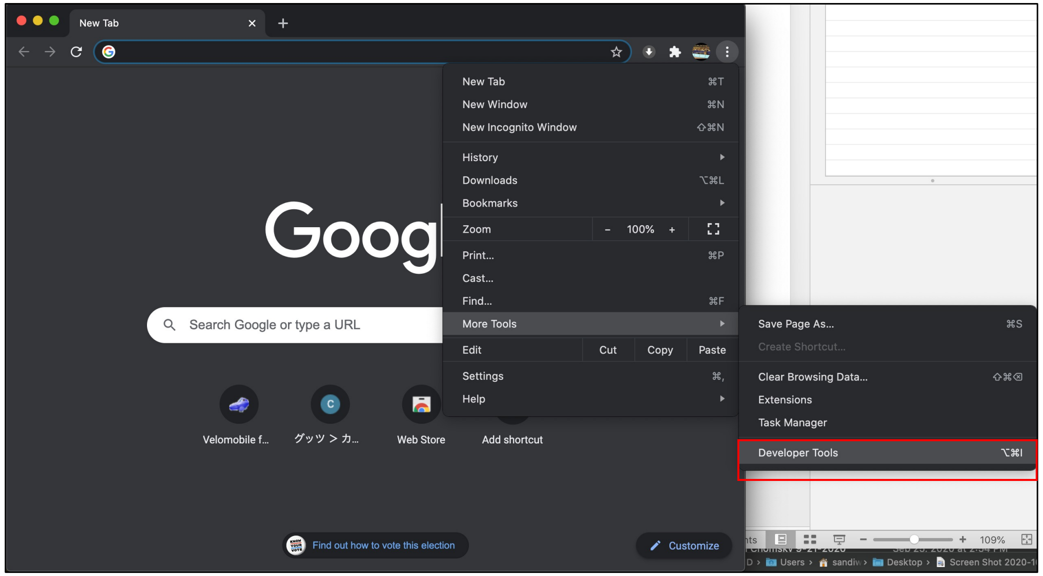


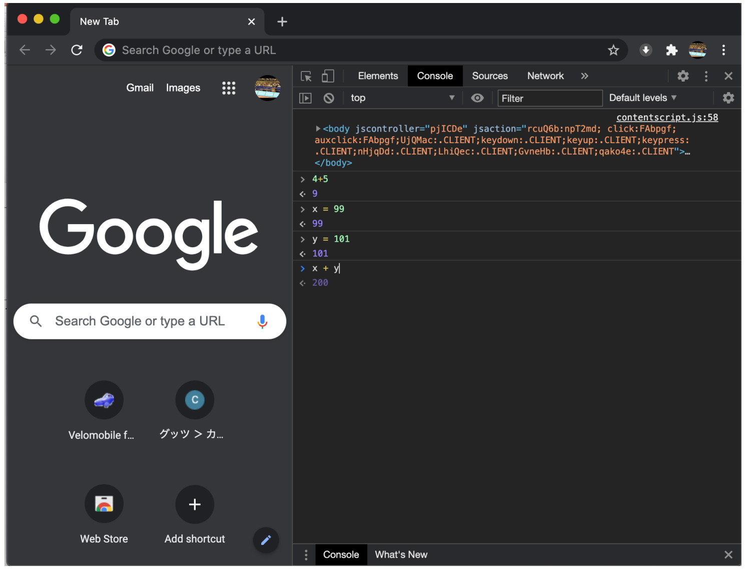


Safari doesn't allow JavaScript from the Smart

```
Console opened at 9:19:08 PM
> 4+5
< 9 = $1
> x = 99
< 99 = $2
> y = 101
< 101 = $3
> x + y
< 200 = $4
> |
```

Javascript Console: Chrome

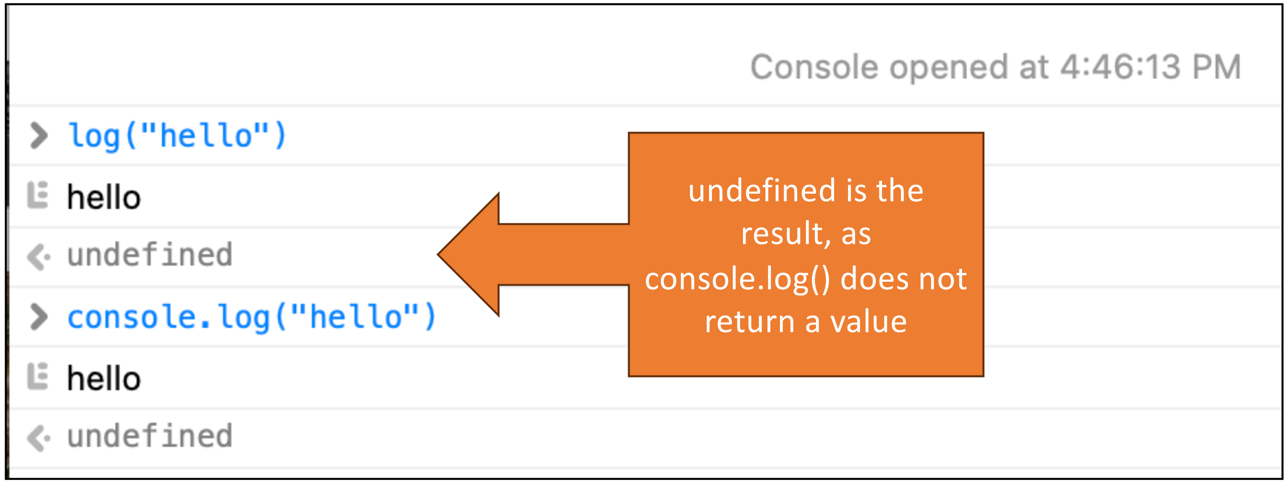




```
                                                          contentscript.js:58
 ▶<body jscontroller="pjICDe" jsaction="rcuQ6b:npT2md; click:FAbpgf;
 auxclick:FAbpgf;UjQMac:.CLIENT;keydown:.CLIENT;keyup:.CLIENT;keypress:
 .CLIENT;nHjqDd:.CLIENT;LhiQec:.CLIENT;GvneHb:.CLIENT;qako4e:.CLIENT">…
 </body>
> 4+5
<· 9
> x = 99
<· 99
> y = 101
<· 101
> x + y
<· 200
```

console.log(*object*)

- From Javascript code *or* the address bar (using `javascript:console.log(`*`object`*`)`) *or* the console directly:

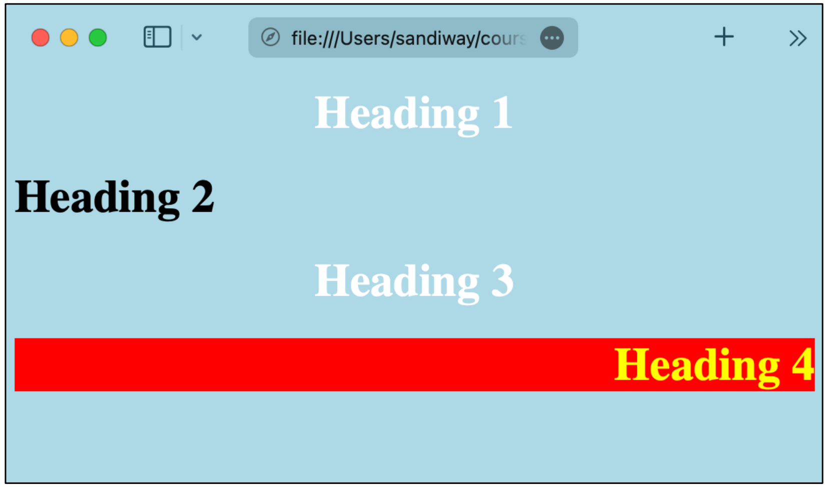


```
Console opened at 4:46:13 PM
> log("hello")
hello
< undefined
> console.log("hello")
hello
< undefined
```

undefined is the result, as console.log() does not return a value

Javascript Console

- Suppose we have a file `sample.html` loaded (*see course website*)
- Show Javascript Console
- Changing the html document:
  - `document.write(`*`string`*`)`
    - *overwrites html document*
  - *`html_element`*`.innerHTML = "`*`string`*`"`
    - *modifies html_element*



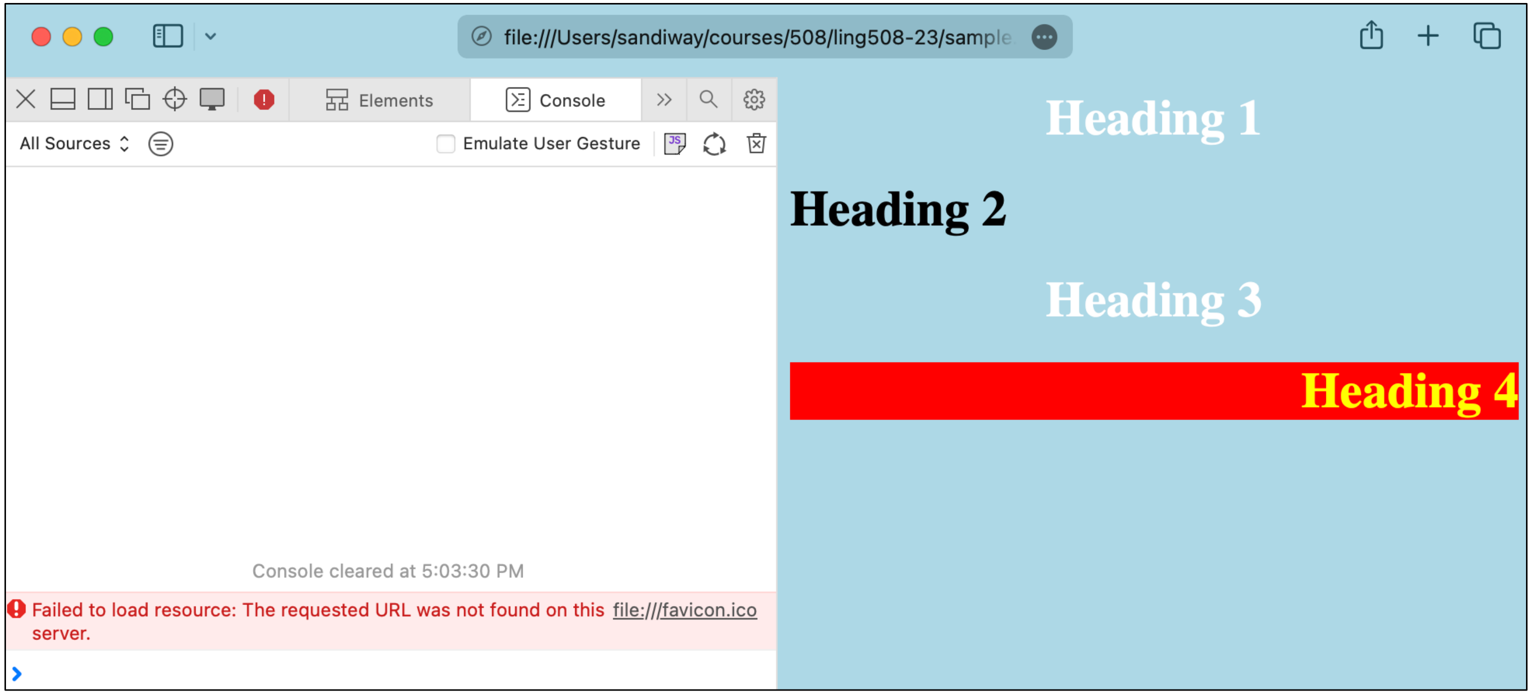
Heading 1

Heading 2

Heading 3

Heading 4

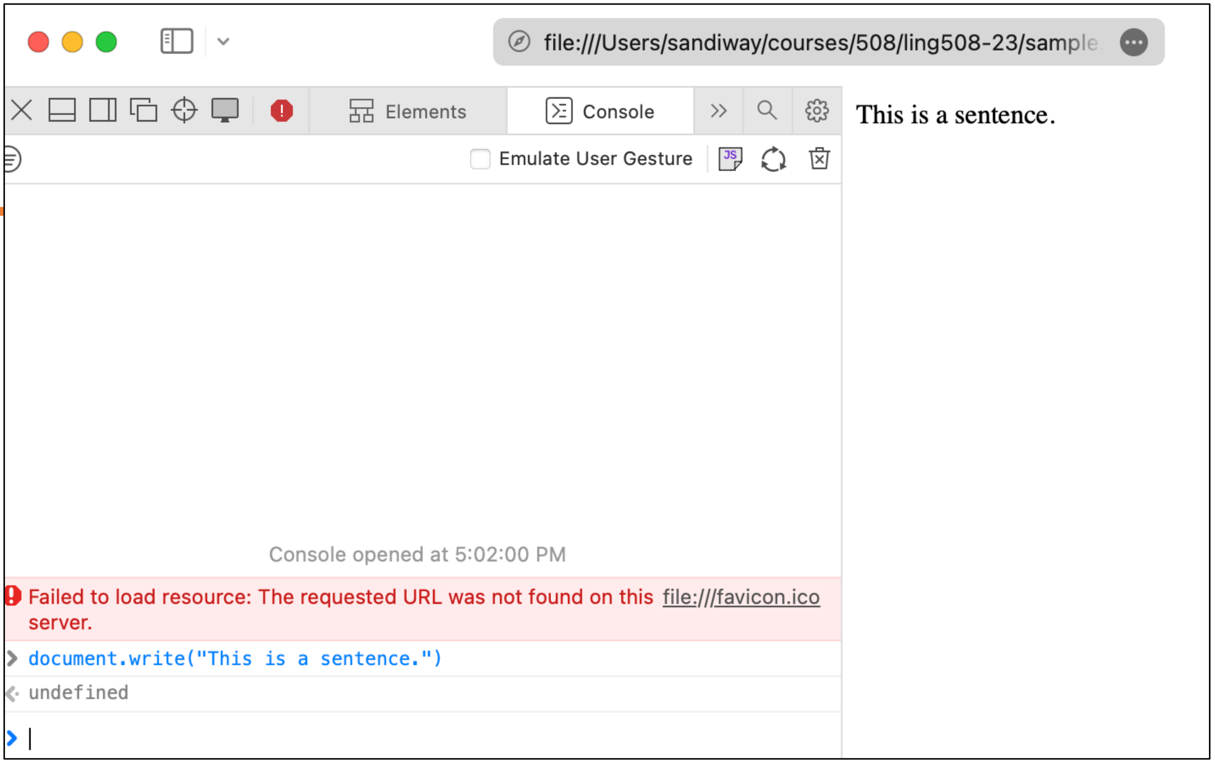
Javascript Console



```
Console cleared at 5:03:30 PM
Failed to load resource: The requested URL was not found on this server.  file:///favicon.ico
```

Javascript Console

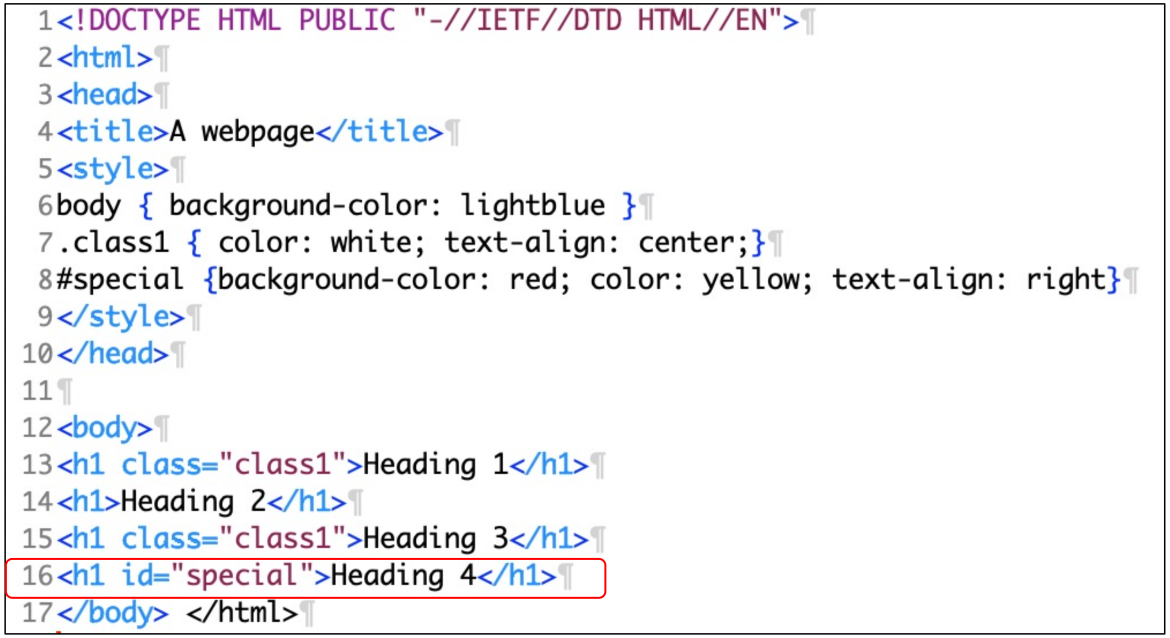
- Changing the html document: document.write(*string*)
  - *overwrites html document*



```
Console opened at 5:02:00 PM
Failed to load resource: The requested URL was not found on this server.   file:///favicon.ico
> document.write("This is a sentence.")
< undefined
>
```

Javascript Console

- Unique id attribute:
  - `id="special"`

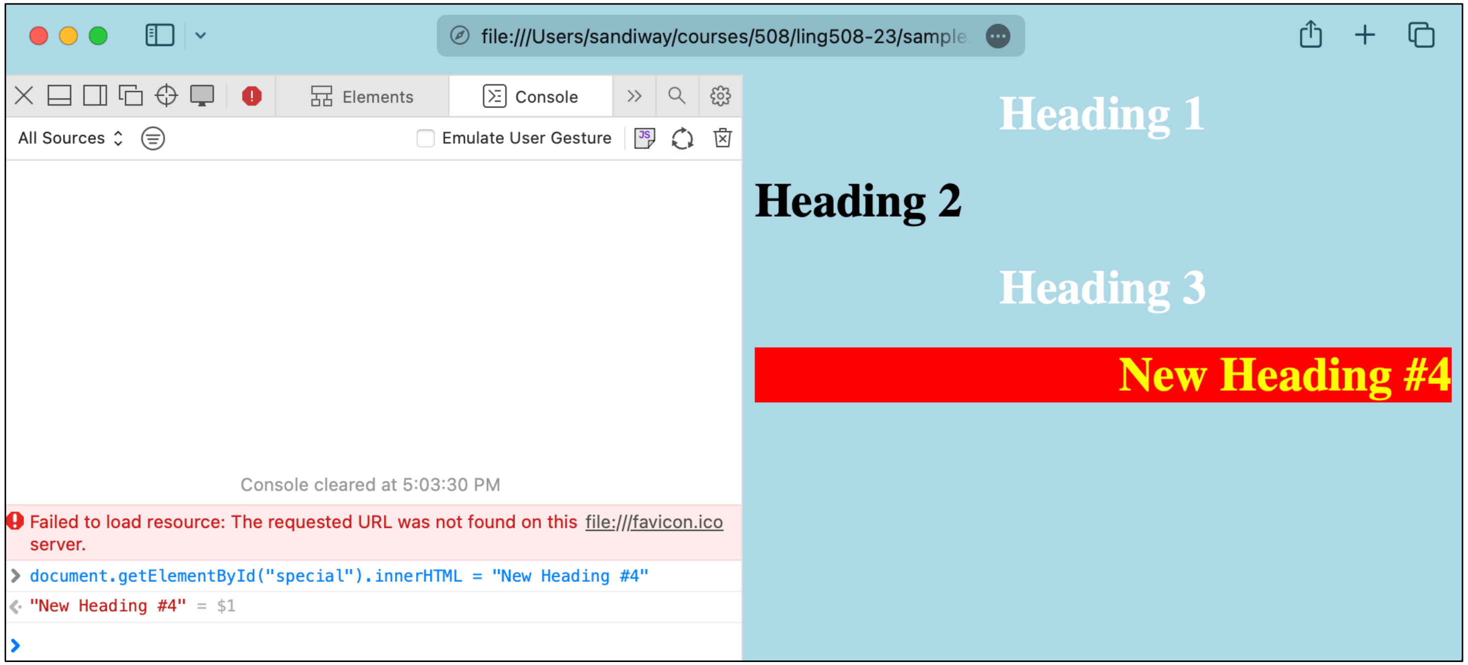


```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<title>A webpage</title>
<style>
body { background-color: lightblue }
.class1 { color: white; text-align: center;}
#special {background-color: red; color: yellow; text-align: right}
</style>
</head>

<body>
<h1 class="class1">Heading 1</h1>
<h1>Heading 2</h1>
<h1 class="class1">Heading 3</h1>
<h1 id="special">Heading 4</h1>
</body> </html>
```

Javascript Console

- `*html_element*.innerHTML = "*string*"`



```
Console cleared at 5:03:30 PM
Failed to load resource: The requested URL was not found on this server. file:///favicon.ico
> document.getElementById("special").innerHTML = "New Heading #4"
< "New Heading #4" = $1
>
```

Heading 1

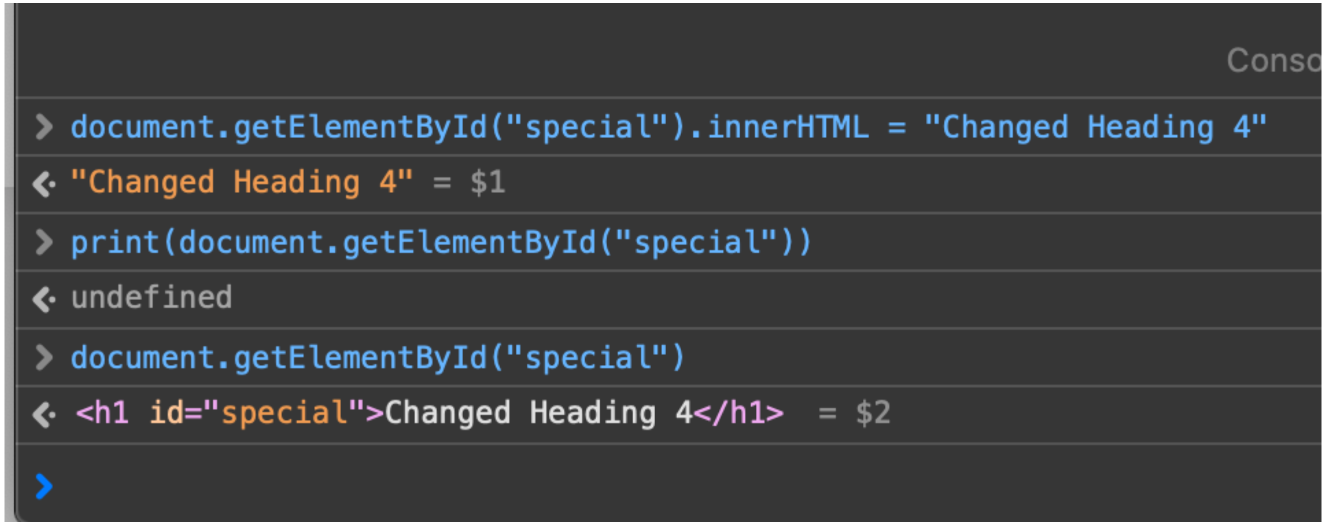
Heading 2

Heading 3

New Heading #4

Javascript Console

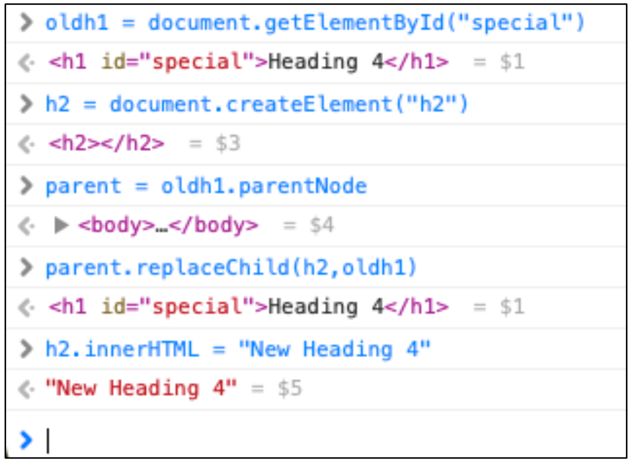
DOM (Domain Object Model): document.getElementById(ID)



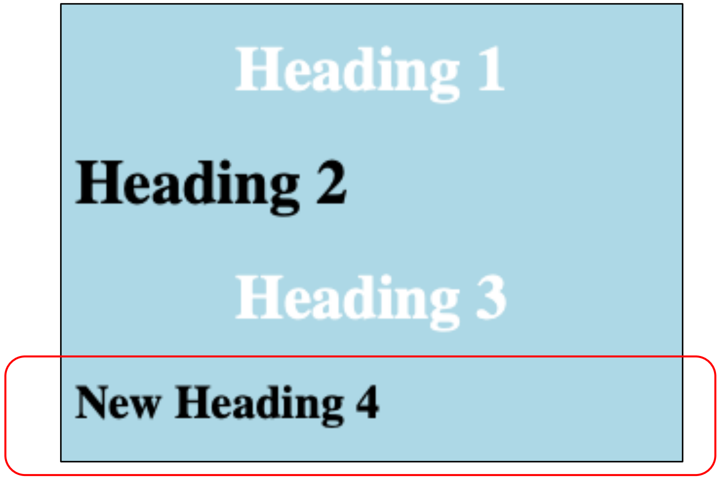
```
> document.getElementById("special").innerHTML = "Changed Heading 4"
<· "Changed Heading 4" = $1
> print(document.getElementById("special"))
<· undefined
> document.getElementById("special")
<· <h1 id="special">Changed Heading 4</h1>  = $2
>
```

Javascript Console

- Suppose you want to change *Heading 4* with tag h1 to tag h2?
- but you can't change the tag of an existing element
- What you can do is create a new tag and replace the child of the parent element:



```
> oldh1 = document.getElementById("special")
<· <h1 id="special">Heading 4</h1>  = $1
> h2 = document.createElement("h2")
<· <h2></h2>  = $3
> parent = oldh1.parentNode
<· ▶ <body>…</body>  = $4
> parent.replaceChild(h2,oldh1)
<· <h1 id="special">Heading 4</h1>  = $1
> h2.innerHTML = "New Heading 4"
<· "New Heading 4" = $5
> |
```



Heading 1

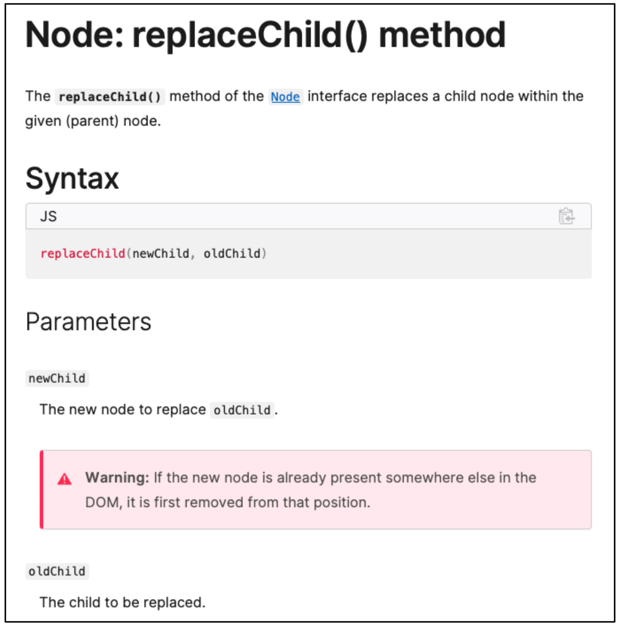
Heading 2

Heading 3

New Heading 4

Javascript Console

- Documented here:
  - https://developer.mozilla.org/en-US/docs/Web/API/Node/replaceChild



Node: replaceChild() method

The `replaceChild()` method of the `Node` interface replaces a child node within the given (parent) node.

Syntax

JS

```
replaceChild(newChild, oldChild)
```

Parameters

`newChild`

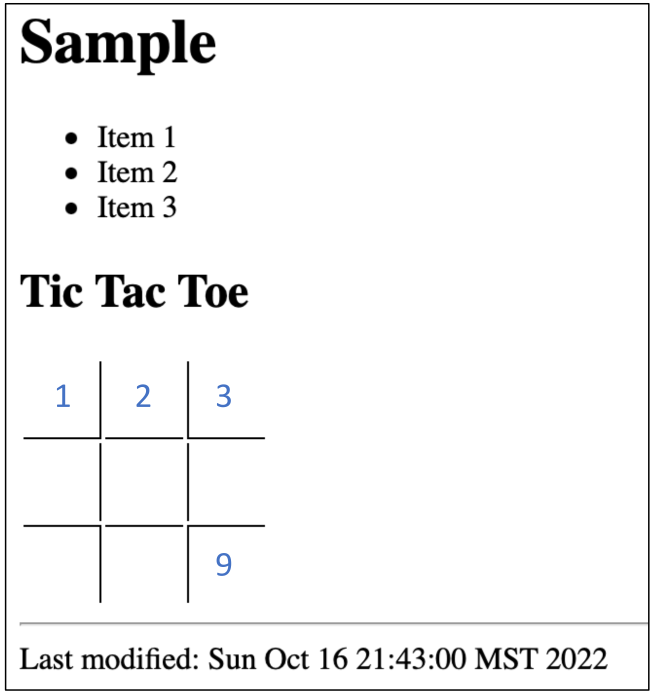
The new node to replace `oldChild`.

▲ **Warning:** If the new node is already present somewhere else in the DOM, it is first removed from that position.

`oldChild`

The child to be replaced.

Tic Tac Toe: `sample2.html`



Sample

- Item 1
- Item 2
- Item 3

Tic Tac Toe

1	2	3
		9

Last modified: Sun Oct 16 21:43:00 MST 2022

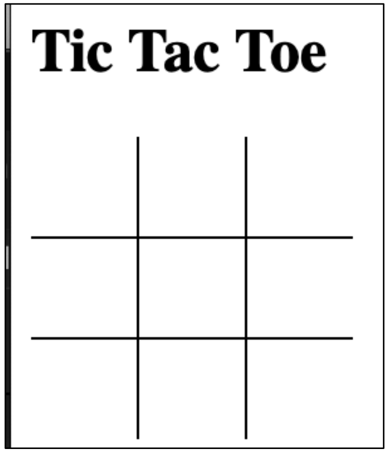
```
<table>
<tr>
  <td id="1" style="border-right: 1px solid black; border-bottom: 1px solid
black"></td>
  <td id="2" style="border-bottom: 1px solid black"></td>
  <td id="3" style="border-left: 1px solid black; border-bottom: 1px solid black"></td>
 </tr>
 <tr>
  <td id="4" style="border-right: 1px solid black"></td>
  <td id="5"></td>
  <td id="6" style="border-left: 1px solid black"></td>
 </tr>
 <tr>
  <td id="7" style="border-right: 1px solid black; border-top: 1px solid black"></td>
  <td id="8" style="border-top: 1px solid black"></td>
  <td id="9" style="border-left: 1px solid black; border-top: 1px solid black" ></td>
 </tr>
</table>
```

Tic Tac Toe: `sample2.html`

- CSS:

```
<style>
table {
  border-collapse: collapse;
}
td {
  height: 40px; width: 40px
}
</style>
```

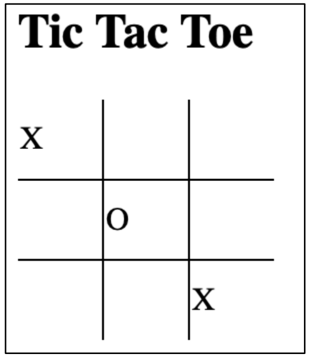
- `collapse`:
  - removes the tiny gap between cells, see below:



Tic Tac Toe

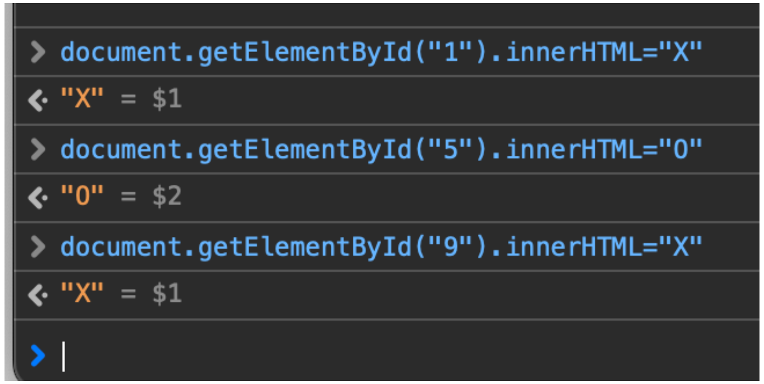
Javascript Console

- Suppose we have `sample2.html` loaded
- `Show Javascript Console`
- Changing the html document:
  - `html_element.innerHTML = "string"`
    - *modifies html_element*
- DOM:
  - `document.getElementById(ID)`
- Let's play tic tac toe!



Tic Tac Toe

X		
	O	
		X



```
> document.getElementById("1").innerHTML="X"
<· "X" = $1
> document.getElementById("5").innerHTML="O"
<· "O" = $2
> document.getElementById("9").innerHTML="X"
<· "X" = $1
> |
```

Javascript Console

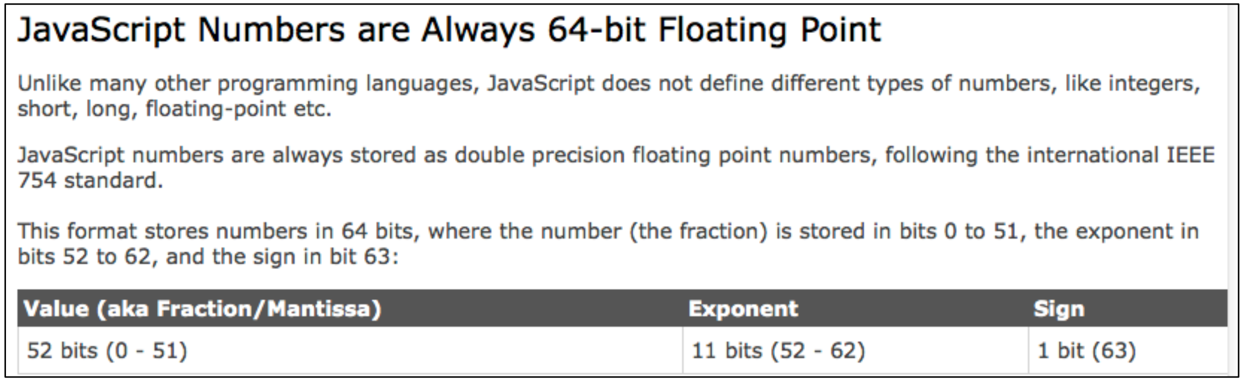
- Notice td cell text are left aligned (by default)
- CSS property:
  - `text-align: center`
    - https://developer.mozilla.org/en-US/docs/Web/CSS/text-align
  - `vertical-align: top|middle|bottom`
    - https://developer.mozilla.org/en-US/docs/Web/CSS/vertical-align

Javascript: Variables

- Declared variables:
  - **`var x = 9;`** *`// number variable`*
  - **`var name = "String";`** *`// string variable`*
  - **Note***: names are case sensitive, no $ like the bash shell, // for in-line comments*
- function local variables:
  - **`function f() {`**
  - **`  var x = 99;`**
  - **`}`**
- Not declared:
  - i.e. *just used without declaration*
  - treated as a global variable
- Strict: prevents use of undeclared variables etc.
  - **`"use strict";`** quotes for backwards compatibility with older versions
- Comments:
  - **`//`** *until the end of the line*
  - **`/* … */`** *multiline* (*same as CSS*)

Javascript: Numbers

http://www.w3schools.com/js/js_numbers.asp



JavaScript Numbers are Always 64-bit Floating Point

Unlike many other programming languages, JavaScript does not define different types of numbers, like integers, short, long, floating-point etc.

JavaScript numbers are always stored as double precision floating point numbers, following the international IEEE 754 standard.

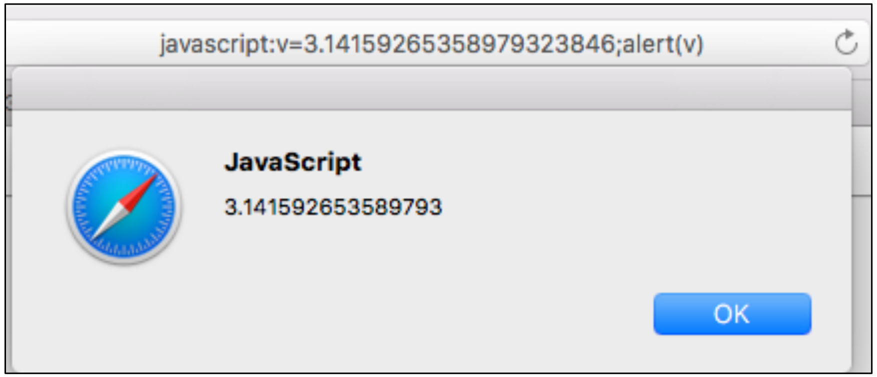
This format stores numbers in 64 bits, where the number (the fraction) is stored in bits 0 to 51, the exponent in bits 52 to 62, and the sign in bit 63:

Value (aka Fraction/Mantissa)	Exponent	Sign
52 bits (0 - 51)	11 bits (52 - 62)	1 bit (63)

- Recall the discussion at the beginning of the semester about number representation?
  - Integers are considered accurate up to 15 digits.

Javascript: Numbers

- Pi to 20 decimal places
  - 3.14159 26535 89793 23846
  - 3.14159 26535 89793 (Javascript)



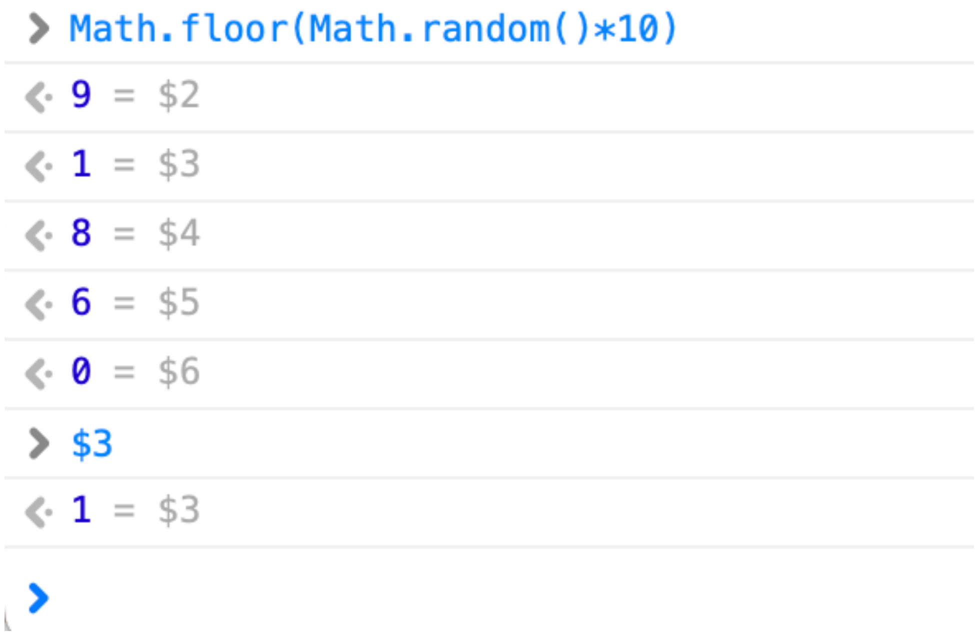
```
javascript:v=3.14159265358979323846;alert(v)
```

JavaScript

```
3.141592653589793
```

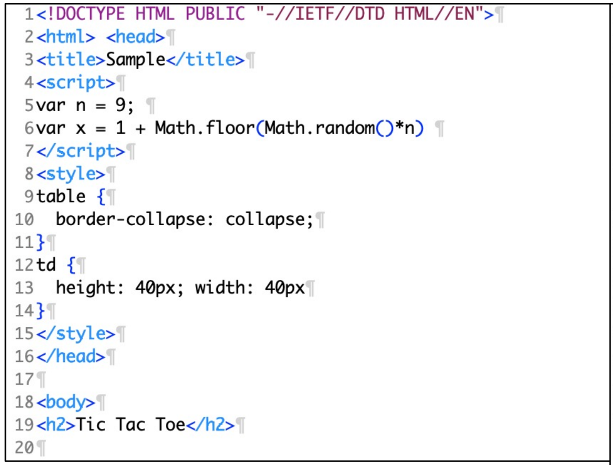
Javascript: Miscellaneous

- Generate a random number in range [0,n-1]:
    - Math.floor(Math.random()**n*)



```
> Math.floor(Math.random()*10)
<· 9 = $2
<· 1 = $3
<· 8 = $4
<· 6 = $5
<· 0 = $6
> $3
<· 1 = $3
>
```

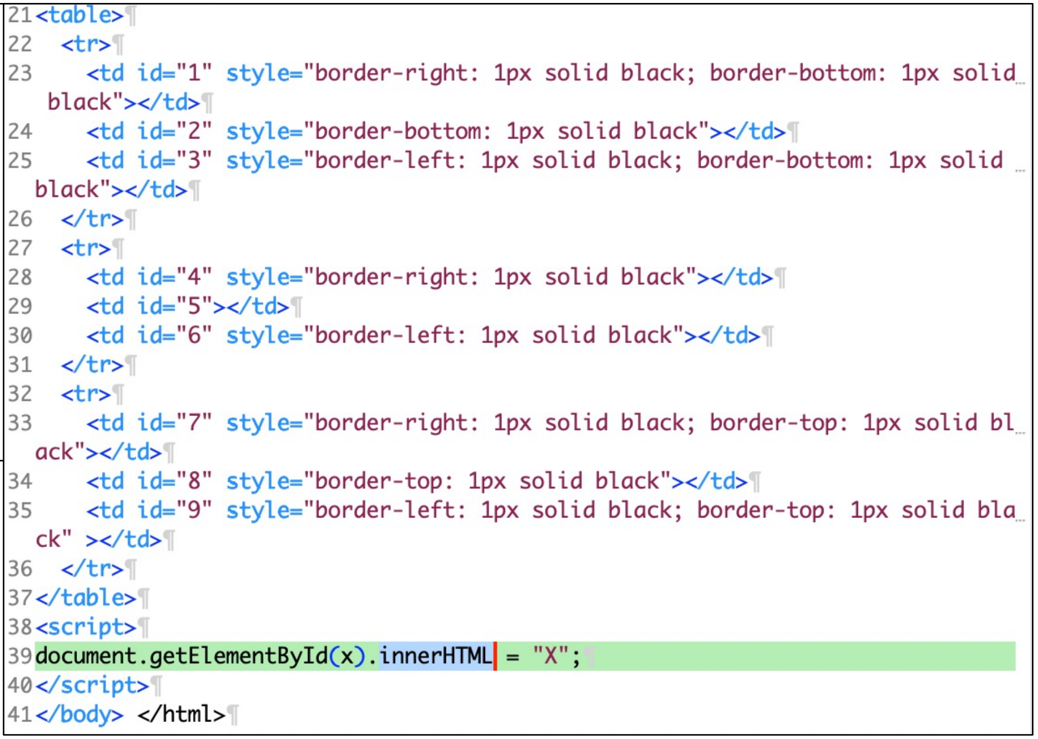
Tic Tac Toe Revisited



```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html> <head>
<title>Sample</title>
<script>
var n = 9;
var x = 1 + Math.floor(Math.random()*n)
</script>
<style>
table {
  border-collapse: collapse;
}
td {
  height: 40px; width: 40px
}
</style>
</head>

<body>
<h2>Tic Tac Toe</h2>

```



```
<table>
  <tr>
    <td id="1" style="border-right: 1px solid black; border-bottom: 1px solid black"></td>
    <td id="2" style="border-bottom: 1px solid black"></td>
    <td id="3" style="border-left: 1px solid black; border-bottom: 1px solid black"></td>
  </tr>
  <tr>
    <td id="4" style="border-right: 1px solid black"></td>
    <td id="5"></td>
    <td id="6" style="border-left: 1px solid black"></td>
  </tr>
  <tr>
    <td id="7" style="border-right: 1px solid black; border-top: 1px solid black"></td>
    <td id="8" style="border-top: 1px solid black"></td>
    <td id="9" style="border-left: 1px solid black; border-top: 1px solid black" ></td>
  </tr>
</table>
<script>
document.getElementById(x).innerHTML = "X";
</script>
</body> </html>
```

File: sample3.html