

Lecture 12

408/508 *Computational Techniques for Linguists*

Today's Topics

- Homework 5
- Javascript console
 - is 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*

CSS in <style> in <head>

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

Selector	Example	Example description
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element</u>	div, p	Selects all <div> elements and all <p> elements
<u>element element</u>	div p	Selects all <p> elements inside <div> elements
<u>element>element</u>	div > p	Selects all <p> elements where the parent is a <div> element

<https://www.w3schools.com/css/>

CSS

- Inline (could go inside <head> ... </head>):
 - <style> ... </style>
- Good way to see what's possible.
 - Google, e.g. css table style
 - https://www.w3schools.com/css/css_table.asp

CSS Tables

[< Previous](#)[Next >](#)

The look of an HTML table can be greatly improved with CSS:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

[Try it Yourself >](#)

Tabs

- Read:
 - https://www.w3schools.com/howto/howto_js_tabs.asp

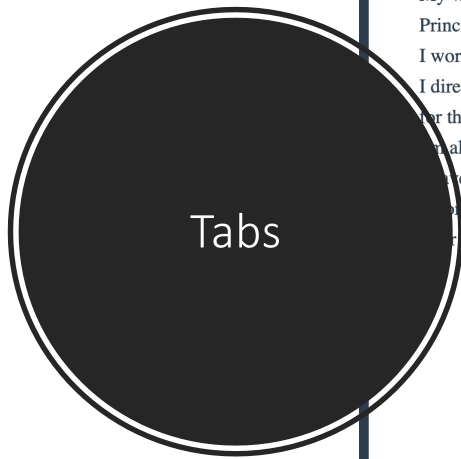
Tabs

Tabs are perfect for single page web applications, or for web pages capable of displaying different subjects:

London Paris Tokyo

London

London is the capital city of England.



Tabs

Sandiway Fong

About me

Research

Software

Classes

UA-related

Cycling

Running

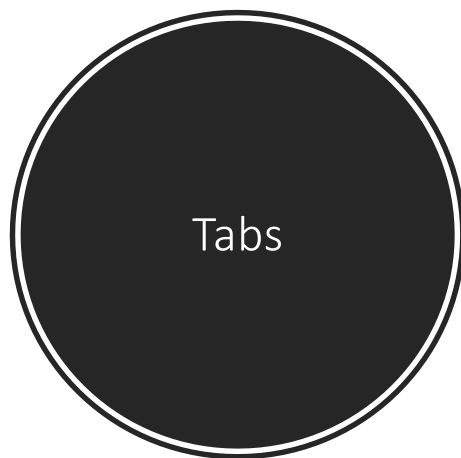
Trains

Coffee

My work intersects computer science and linguistics.
Principally, I'm interested in the computational modeling of linguistic theory.
I work at the [University of Arizona](#) (UA).
I direct and teach in the in-person Master's Program in Human Language Technology ([HLT](#))
for the [Department of Linguistics](#).
I am also a member of the UA [Cognitive Science Program](#) and [SLAT](#).
I have a long-term collaboration with linguistics faculty at Osaka Kyoiku University ([OKU](#)) in Japan.
I work primarily with professors [Jason Ginsburg](#) and [Hiroshi Terada](#).
(The two universities have a formal [academic exchange agreement](#).)



Picture courtesy of Leslie Hawthorne Klingler.



Sandiway Fong

About me	Research	Software	Classes	UA-related	Cycling	Running	Trains	Coffee
----------	----------	----------	---------	------------	---------	---------	--------	--------

Picture courtesy of Leslie Hav

Recent research projects, talks and papers.

On the nature of FormSet. [Abstract](#)
Fong, S. & M. Oishi.
FIND-Workshop 2023, December 11-12, Göttingen, Germany.

The Strong Minimalist Thesis (SMT): Form Copy (FC) and the Serial Verb Construction (SVC).
Fong, S., J. Ginsburg, M. Matsumoto & H. Terada.
In *Proceedings of Japanese/Korean Linguistics 30* (JK30). March 11th–13th 2023.
Simon Fraser University, BC, Canada, pp.411-421, CSLI.
[Link](#)

On the computational modeling of English relative clauses.
Fong, S. & J. Ginsburg. (in press)
Open Linguistics, vol. 9, no. 1, 2023. Article number: 20220246, De Gruyter Open Access.
[Link coming soon!](#)

Project: Minimalist Machine: computer implementation of a generative theory in the Minimalist Program (MP).
[See](#) theory implemented and many examples of derivations.
(Joint work with Jason Ginsburg of Osaka Kyoiku University.)

Parsing with Box theory: to be updated soon.

Linguistic Theory and Deep Learning.
Generative AI: [ChatGPT and Natural Language Understanding](#)

A portrait of Sandiway Fong, a man with short dark hair, wearing a light blue shirt, positioned in the bottom right corner of the page content area.

Tabs

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width,
initial-scale=1">
<style>
body {font-family: Arial;}
/* Style the tab */
.tab {
  overflow: hidden;
  border: 1px solid #ccc;
  background-color: #f1f1f1;
}
/* Style the buttons inside the tab */
.tab button {
  background-color: inherit;
  float: left;
  border: none;
  outline: none;
  cursor: pointer;
  padding: 14px 16px;
  transition: 0.3s;
```

```
  font-size: 17px;
}
/* Change background color of buttons on hover */
.tab button:hover {
  background-color: #ddd;
}
/* Create an active/current tablink class */
.tab button.active {
  background-color: #ccc;
}
/* Style the tab content */
.tabcontent {
  display: none;
  padding: 6px 12px;
  border: 1px solid #ccc;
  border-top: none;
}
</style>
</head>
```

Tabs

```
<body>
<h2>Tabs</h2>
<p>Click on the buttons inside the tabbed menu:</p>
<div class="tab">
  <button class="tablinks" onclick="openCity(event,
'London')">London</button>
  <button class="tablinks" onclick="openCity(event,
'Paris')">Paris</button>
  <button class="tablinks" onclick="openCity(event,
'Tokyo')">Tokyo</button>
</div>
<div id="London" class="tabcontent">
  <h3>London</h3>
  <p>London is the capital city of England.</p>
</div>
<div id="Paris" class="tabcontent">
  <h3>Paris</h3>
  <p>Paris is the capital of France.</p>
</div>
<div id="Tokyo" class="tabcontent">
  <h3>Tokyo</h3>
  <p>Tokyo is the capital of Japan.</p>
</div>
```

```
<script>
function openCity(evt, cityName) {
  var i, tabcontent, tablinks;
  tabcontent = document.getElementsByClassName("tabcontent");
  for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablinks");
  for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace("
active", "");
  }
  document.getElementById(cityName).style.display = "block";
  evt.currentTarget.className += " active";
}
</script>
</body>
</html>
```

Homework 5

1. Build your own **html** file, play with the **tags**
2. Embed a picture of yourself encoded in **base64**
3. Make sure your page has various **tabs**

Homework 5

- Instructions:

- email sandiway@arizona.edu
- submit everything in one PDF file!
 - screenshot of the webpage you built
 - the source html of the webpage
- subject of email: 408/508 Homework 5 *your name*
- cite any discussion or source
- due date: next Sunday by midnight

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

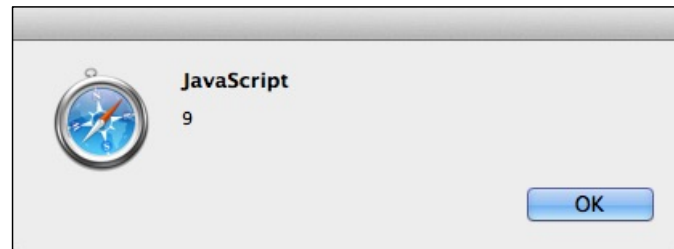
code snippets can be
placed in
`<head> .. </head>`
or
`<body> .. </body>`

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

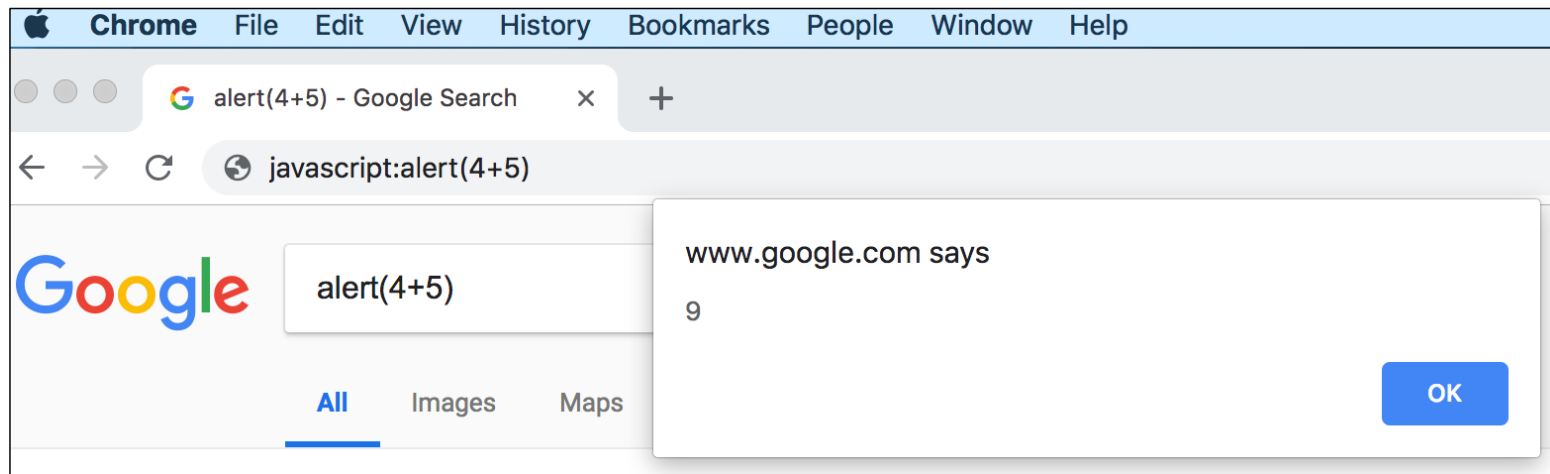
Safari doesn't allow JavaScript from the Smart Search Field.

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

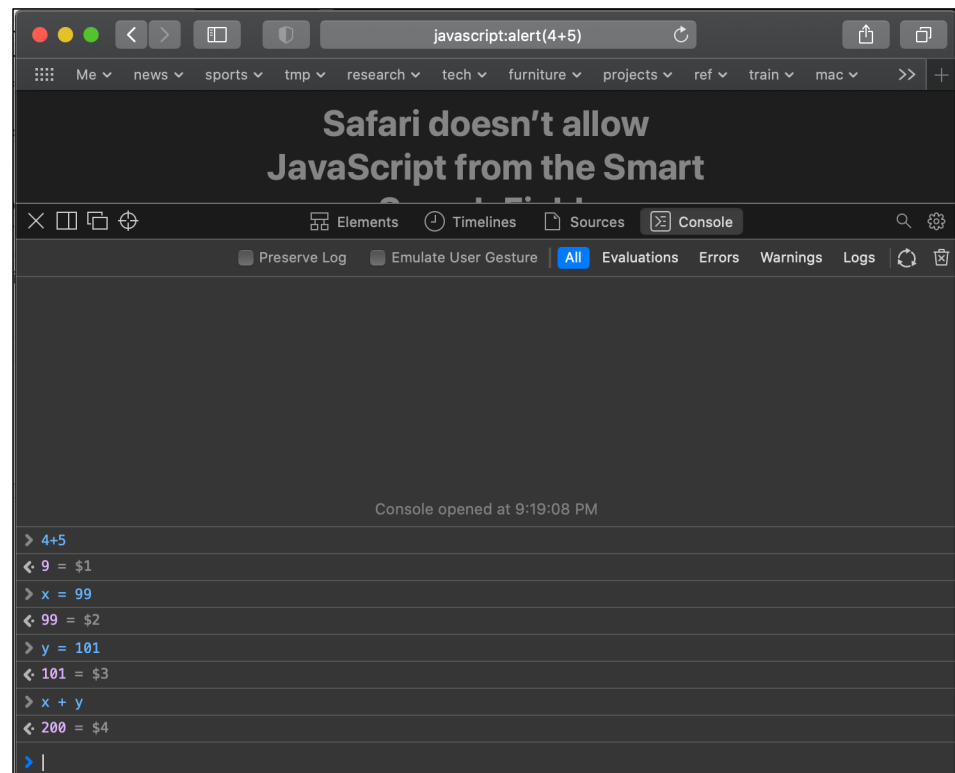
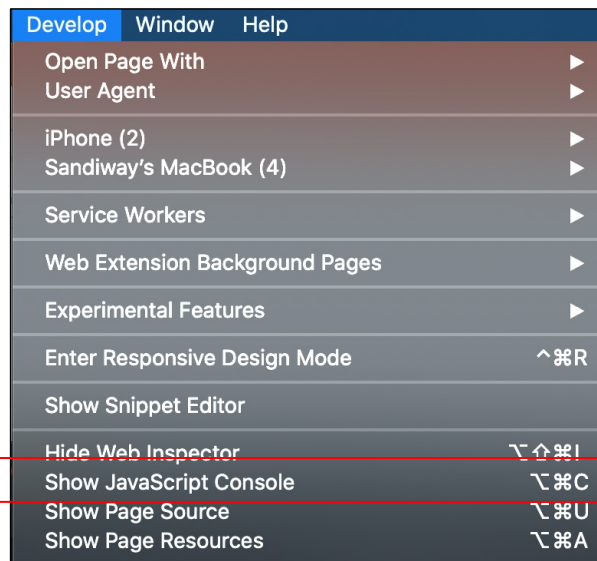


- Changing the html document using Javascript:
 - `document.write(string)` *overwrites html document*
 - `html_element.innerHTML = "string"` *modifies html_element*

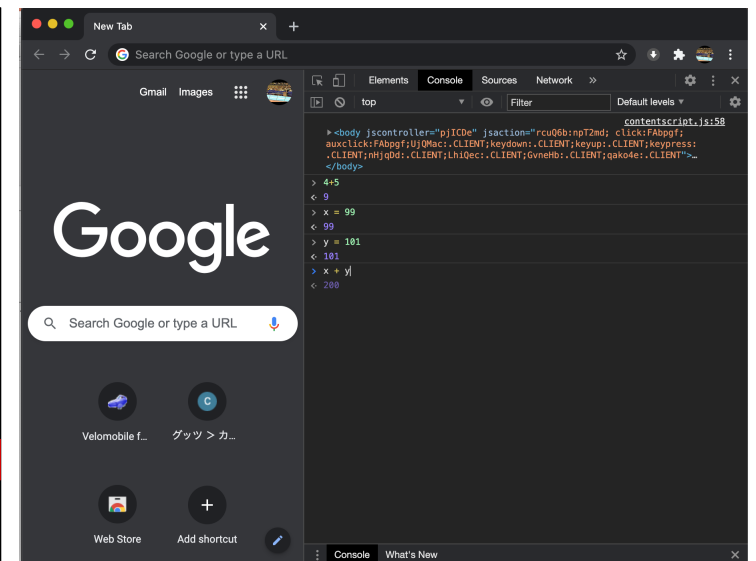
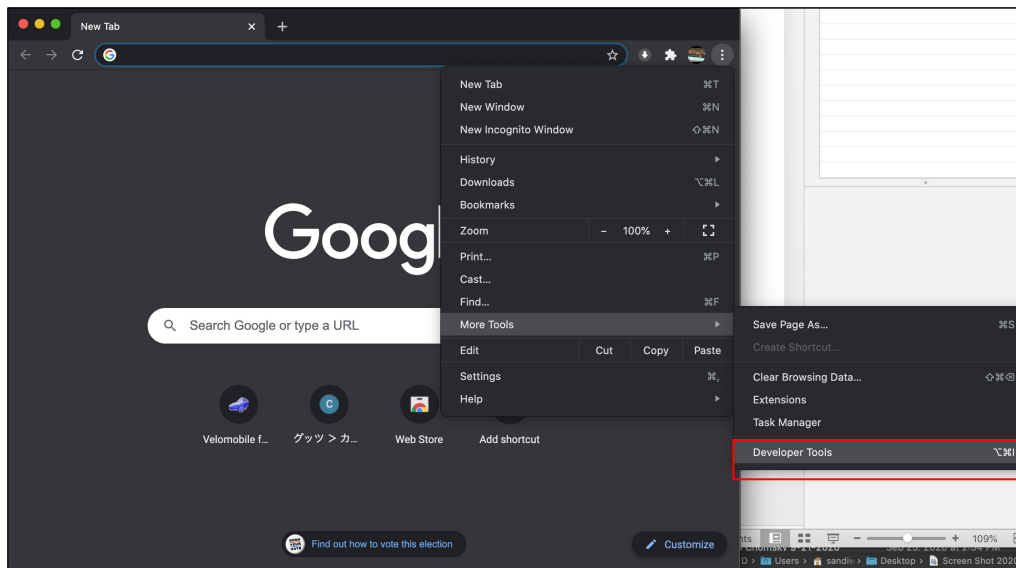
Javascript pop-up notification



Javascript Console

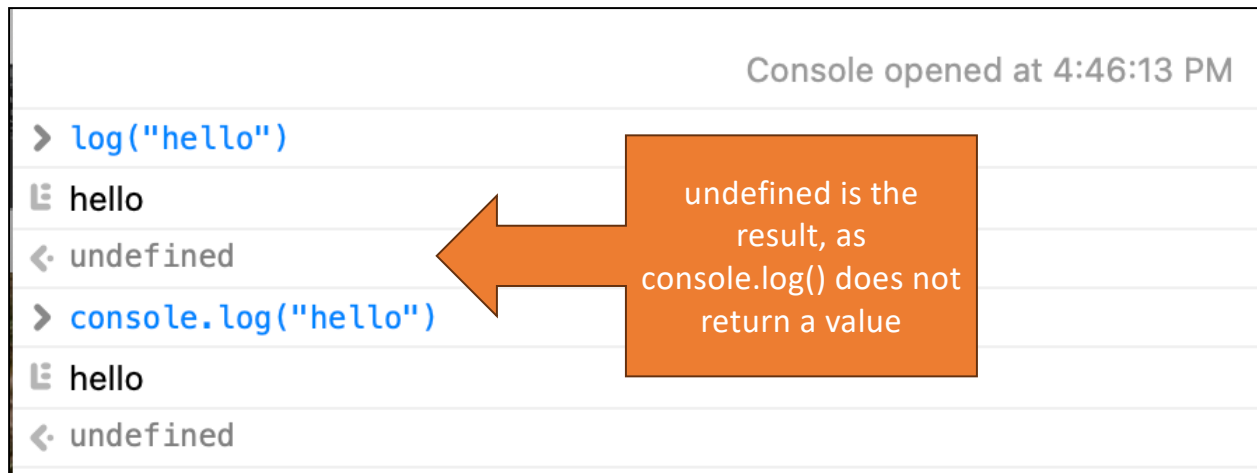


Javascript Console: Chrome



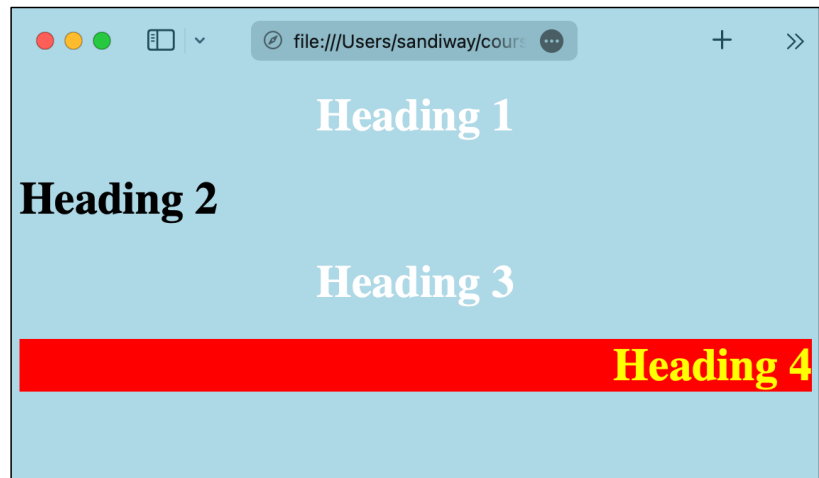
console.log(*object*)

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

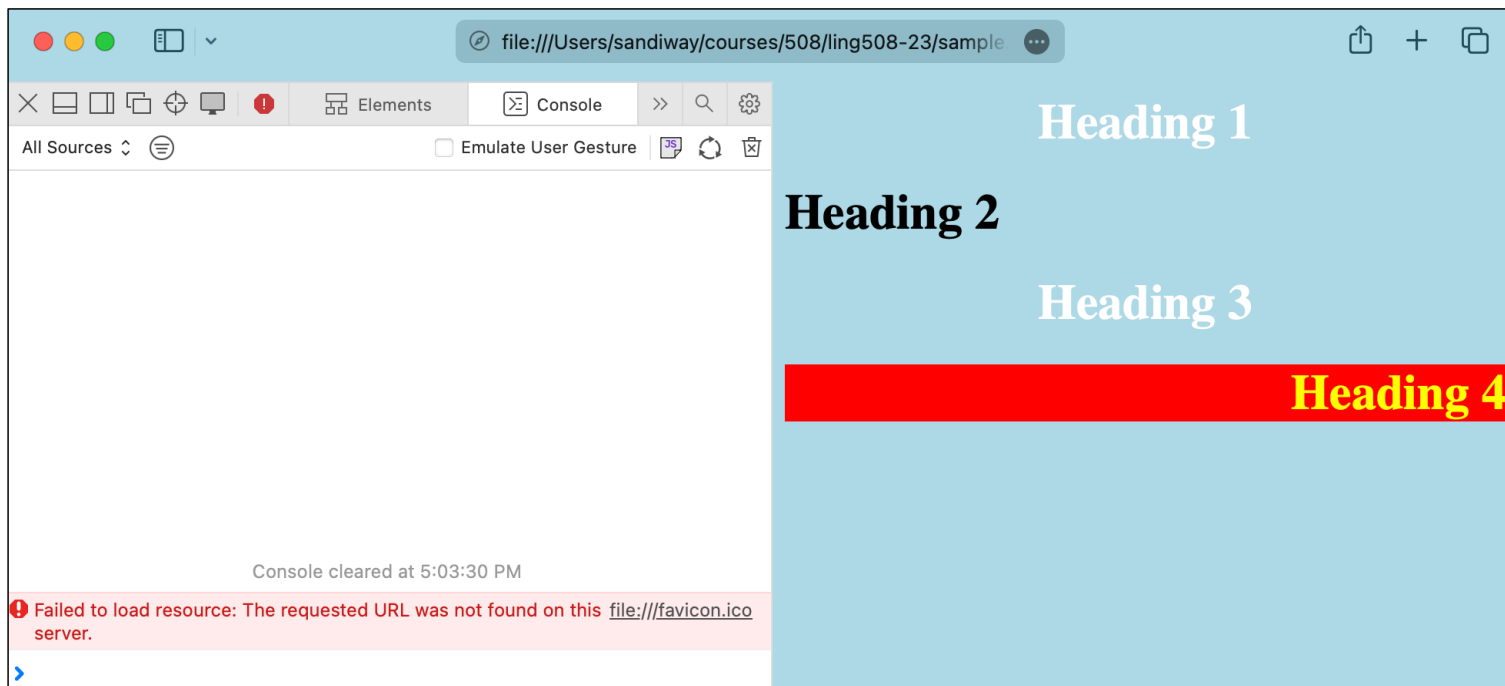


Javascript Console

- 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*

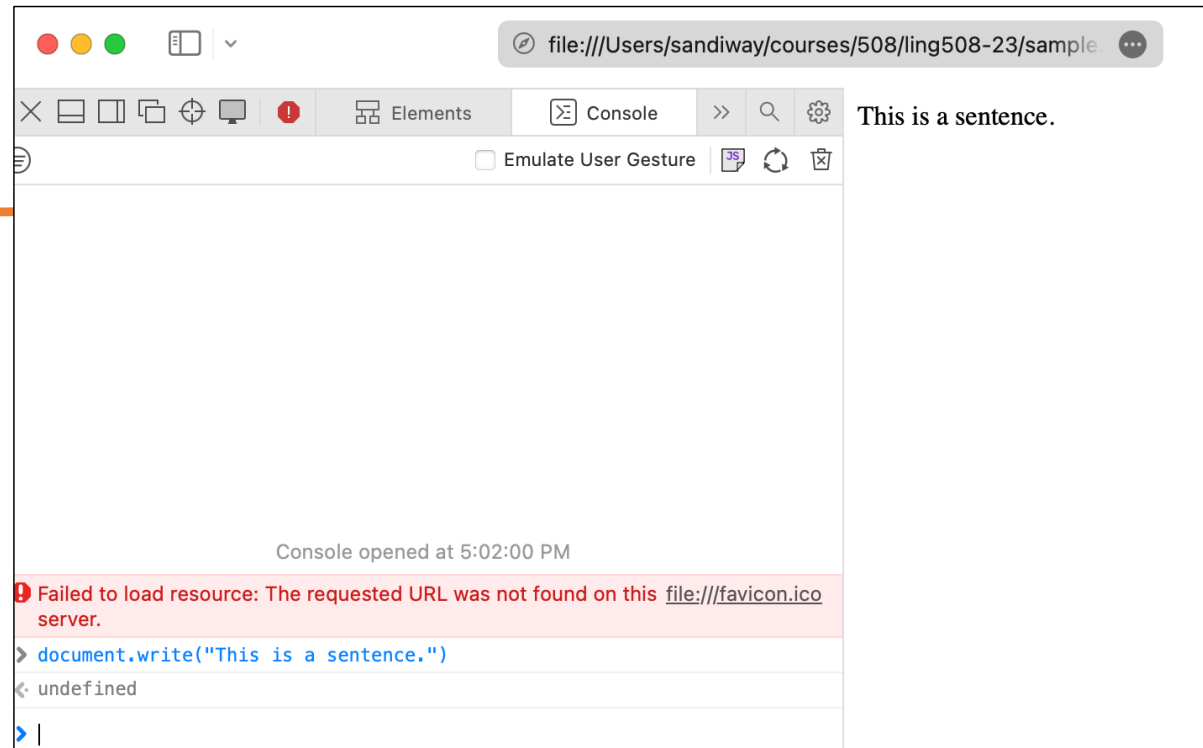


Javascript Console



Javascript Console

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



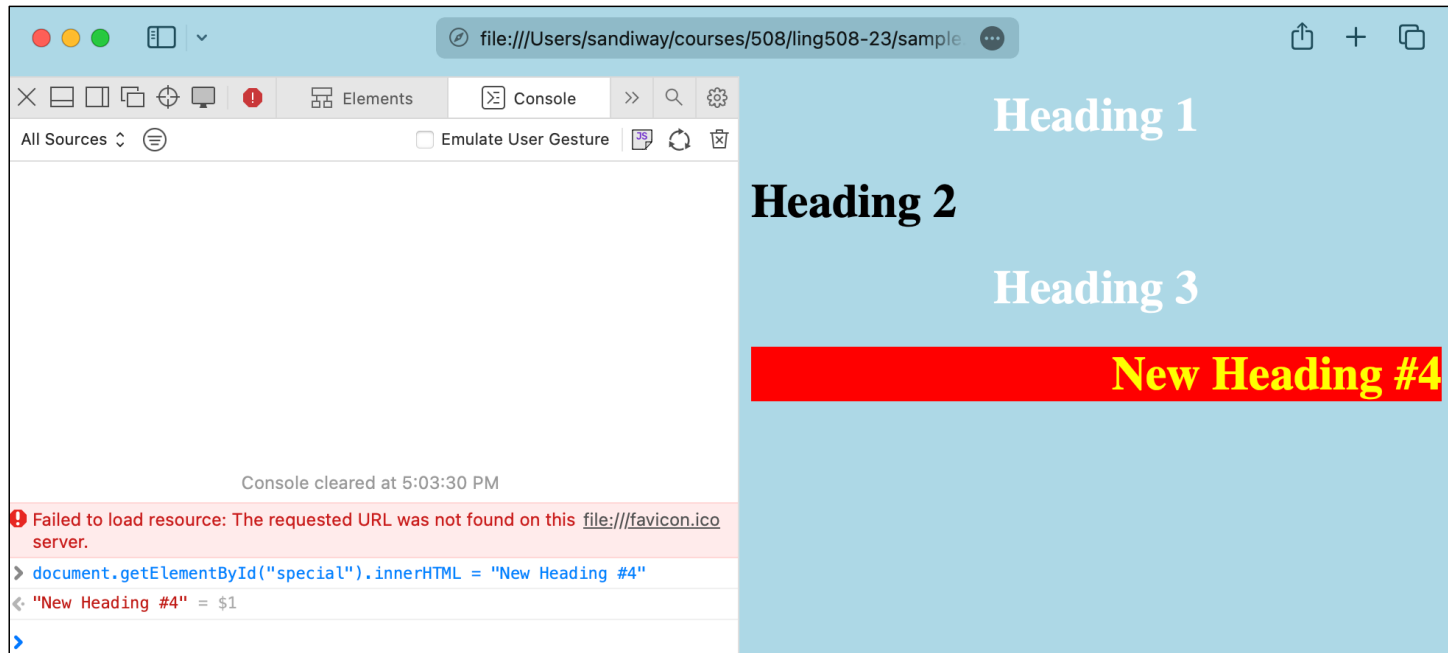
Javascript Console

- Unique id attribute:
 - id="special"

```
1<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
2<html>
3<head>
4<title>A webpage</title>
5<style>
6body { background-color: lightblue }
7.class1 { color: white; text-align: center;}
8#special {background-color: red; color: yellow; text-align: right}
9</style>
10</head>
11
12<body>
13<h1 class="class1">Heading 1</h1>
14<h1>Heading 2</h1>
15<h1 class="class1">Heading 3</h1>
16<h1 id="special">Heading 4</h1>
17</body> </html>
```

Javascript Console

- `html_element.innerHTML = "string"`



Javascript Console

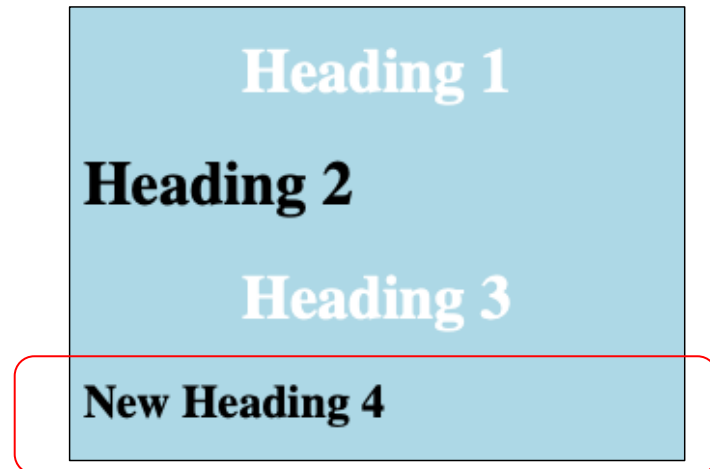
DOM (Domain Object Model): `document.getElementById(ID)`

```
Consol
> 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
> |
```



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

```
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>
```

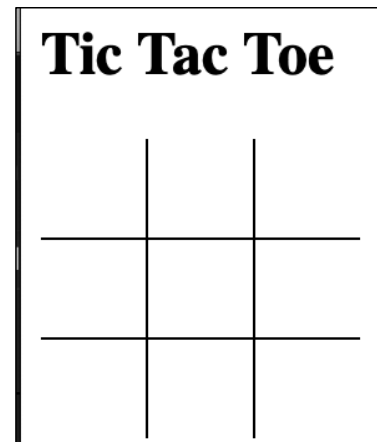
Tic Tac Toe: sample2.html

- CSS:

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

- collapse:

- removes the tiny gap between cells, see below:



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:

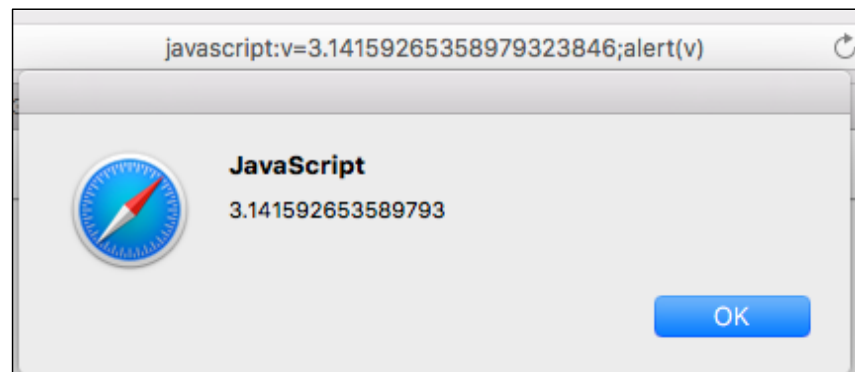
Value (aka Fraction/Mantissa)	Exponent	Sign
52 bits (0 - 51)	11 bits (52 - 62)	1 bit (63)

- 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: 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

```
1<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
2<html> <head>
3<title>Sample</title>
4<script>
5var n = 9;
6var x = 1 + Math.floor(Math.random()*n)
7</script>
8<style>
9table {
10 border-collapse: collapse;
11}
12td {
13 height: 40px; width: 40px
14}
15</style>
16</head>
17
18<body>
19<h2>Tic Tac Toe</h2>
20
```

File: sample3.html

```
21<table>
22  <tr>
23    <td id="1" style="border-right: 1px solid black; border-bottom: 1px solid
24      black"></td>
25    <td id="2" style="border-bottom: 1px solid black"></td>
26    <td id="3" style="border-left: 1px solid black; border-bottom: 1px solid
27      black"></td>
28  </tr>
29  <tr>
30    <td id="4" style="border-right: 1px solid black"></td>
31    <td id="5"></td>
32    <td id="6" style="border-left: 1px solid black"></td>
33  </tr>
34  <tr>
35    <td id="7" style="border-right: 1px solid black; border-top: 1px solid bl
36      ack"></td>
37    <td id="8" style="border-top: 1px solid black"></td>
38    <td id="9" style="border-left: 1px solid black; border-top: 1px solid bla
39      ck"></td>
40  </tr>
41</table>
42<script>
43document.getElementById(x).innerHTML = "X";
44</script>
45</body> </html>
```