LING 408/508: Computational Techniques for Linguists

Lecture 18

Today's Topic

- DOM (Document Object Model)
  - writing code in the console
  - writing code in <script>
  - triggering code with <button>
- Homework 7

Document Object Model (DOM)

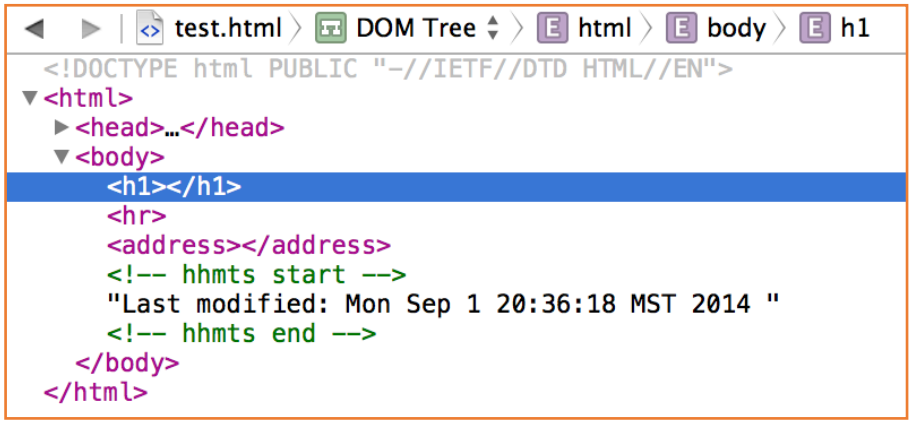
- HTML document

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html> <head>
<title></title>
</head>

<body>
<h1></h1>


<hr>
<address></address>
<!-- hhmts start --><!-- hhmts end -->
</body> </html>
```

- Tree representation



html: document.documentElement
body: document.body

Document Object Model (DOM)

Properties for traversing the DOM:

- *e*.childNodes
  - children of element el as an array, e.g. childNodes[0]
- *e*.children
  - element nodes only (excludes text nodes)
- *e*.firstChild
- *e*.lastChild
- *e*.parentNode
- *e*.nextSibling
- *e*.previousSibling

Object properties:

- *e*.nodeType
  - 1 = element, 3 = text
- *e*.nodeName
  - uppercase
- *e*.innerHTML
  - for element nodes
  - value is html as text
  - writeable
- *e*.nodeValue
  - for text nodes
    (null: for element nodes)
  - writeable

Document Object Model (DOM)

New content:

- document.createElement(tag)
  - tag = ‘div’, ‘p’ etc.
  - creates new DOM element
- document.createTextNode(text)
  - creates new DOM element of type text

For non-HTML elements:

- document.createElementNS(NS,tag)
  - NS = Namespace URL identifier
  - e.g. http://www.w3.org/2000/svg and tag “rect” (rectangle)

Place new_el:

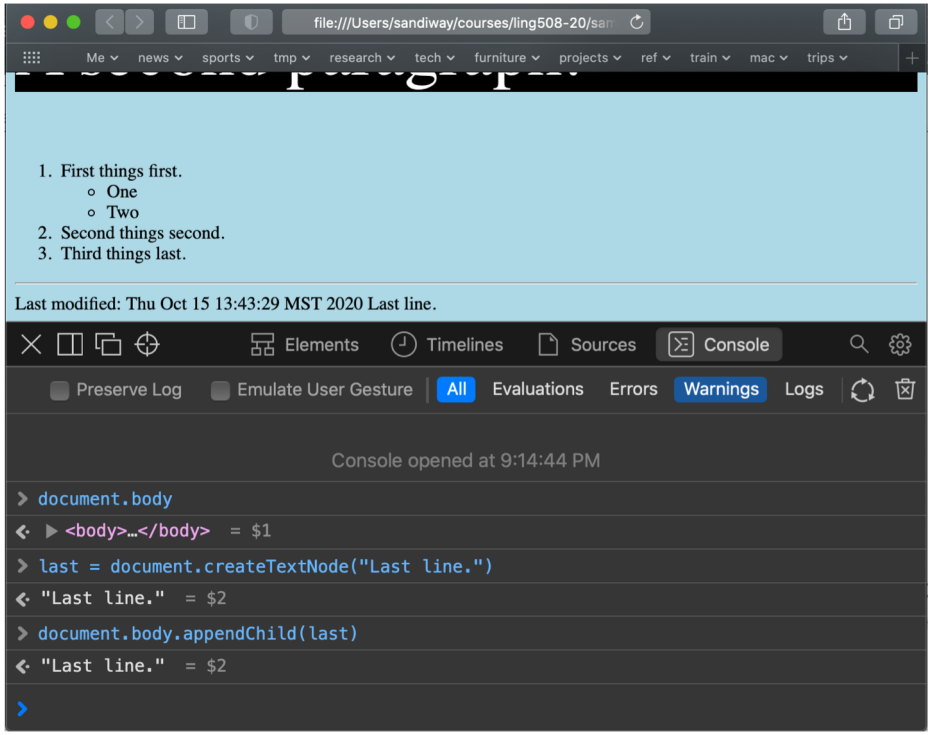
- e.appendChild(new_el)
  - new_el is inserted as last child of el
- e.insertBefore(new_el,next_el)
  - new_el inserted as previous sibling of next_el
  - el is common parent
- e.removeChild(child_el)
  - child_el is deleted
  - el is parent
- e.replaceChild(new_el,child_el)
  - new_el replaces child_el
  - el is parent

Old way:

- document.write(text)
- document.writeln(text)
  - adds a newline

Document Object Model (DOM)

- Let's modify sample.html by Javascript
- Steps on the console:
  1. get document body
  2. create a text node
  3. append it to the end of the body
- Make it a program:
  - put the code in a <script>



Document Object Model (DOM)

- Let's modify sample.html by Javascript
- Steps on the console:
  1. get document body
  2. create a text node
  3. append it to the end of the body
- Make it a program:
  - put the code in a <script>

```
18 </style>¶
19 <script>¶
20     function f(text) {¶
21         document.body.appendChild(document.createTextNode(text))¶
22     }¶
23 </script>¶
24 </head>¶
25 ¶
26 <body>¶
```

```
7709 </p>¶
7710 <script>¶
7711 f("Some text here")¶
7712 </script>¶
7713 <hr>¶
7714 <address></address>¶
7715 <!-- hhmts start -->Last modified: Wed Oct 21 21:32:28 MST 2020 <!-- hhmts
     end -->¶
7716 </body> </html>¶
```

Document Object Model (DOM)

- Let's modify sample.html by Javascript
- Steps on the console:
  1. get document body
  2. create a text node
  3. append it to the end of the body
- Make it a program:
  - put the code in a <script>
  - trigger it with a <button>
    - https://www.w3schools.com/jsref/event_onclick.asp


```
19  <script>¶
20     function f(text) {¶
21        document.body.appendChild(document.createTextNode(text))¶
22     }¶
23  </script>¶
24  </head>¶
25  ¶
26  <body>¶
27    <button onclick="f('Some text')">Click me</button>¶
```

Document Object Model (DOM)

Locating an element:

- document.getElementById(id)
  - useful if you have named the document element using the id='Name' property
- document.getElementsByTagName(tag)
  - all document elements of type tag
  - returns an array
- e.getElementsByTagName(tag)
  - all elements of type tag under el
  - returns an array
- document.getElementsByName(name)
  - useful for elements that support name='Name'
- (document|e).getElementsByClassName(class)
- (document|e).querySelector(query)
  - example query 'BODY > UL > LI'
  - '>' means immediately dominates
  - returns first matching element
- (document|e).querySelectorAll(query)
  - returns an array

Homework 7

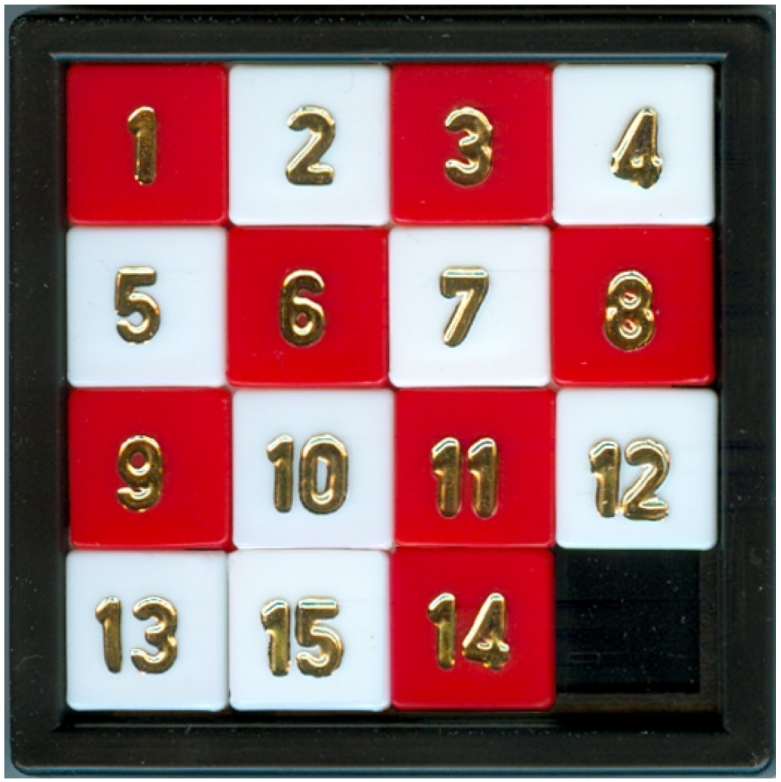


Image from wikipedia

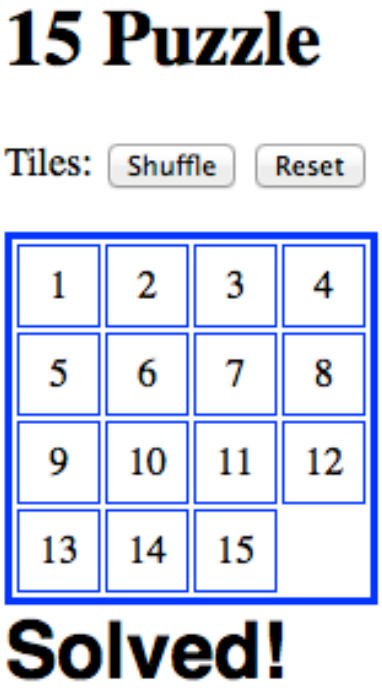
Write a html/Javascript program that simulates the 15 puzzle.

Tasks/Questions:

1. It should bring up a html page that allows the user to manually click on and move the tiles
2. It should be able to initially jumble the tiles
   - **Hint**: use the Math.random() method from last lecture
3. It should recognize and print a message when the user solves the puzzle

Homework 7

Example:



Homework 7

```
<table id="puzzle">
  <tr>
    <td onclick="f(this)">1</td>
    <td onclick="f(this)">2</td>
    <td onclick="f(this)">3</td>
    <td onclick="f(this)">4</td>
  </tr>
  <tr>
    <td onclick="f(this)">5</td>
    <td onclick="f(this)">6</td>
    <td onclick="f(this)">7</td>
    <td onclick="f(this)">8</td>
  </tr>
  <tr>
    <td onclick="f(this)">9</td>
    <td onclick="f(this)">10</td>
    <td onclick="f(this)">11</td>
    <td onclick="f(this)">12</td>
  </tr>
  <tr>
    <td onclick="f(this)">13</td>
    <td onclick="f(this)">14</td>
    <td onclick="f(this)">15</td>
    <td onclick="f(this)"></td>
  </tr>
</table>
```

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

```
<style>
td {
border: 1px solid blue;
width: 30px;
height: 30px;
text-align: center;
vertical-align: middle
}
td:hover {
background: yellow
}
</style>
```

DOM code to locate a specific td by row r (0-3) and col number c (0-3):
`document.getElementById("puzzle").rows[r].cells[c];`

Homework 7

- Table layout:

row: 0 col: 0	row: 0 col: 1	row: 0 col: 2	row: 0 col: 3
row: 1 col: 0	row: 1 col: 1	row: 1 col: 2	row: 1 col: 3
row: 2 col: 0	row: 2 col: 1	row: 2 col: 2	row: 2 col: 3
row: 3 col: 0	row: 3 col: 1	row: 3 col: 2	row: 3 col: 3

table cell:

```
<td onclick="f(this)">..</td>
```

```
<script>
function f(e) {
  .. code ..
}
</script>
```

row number:

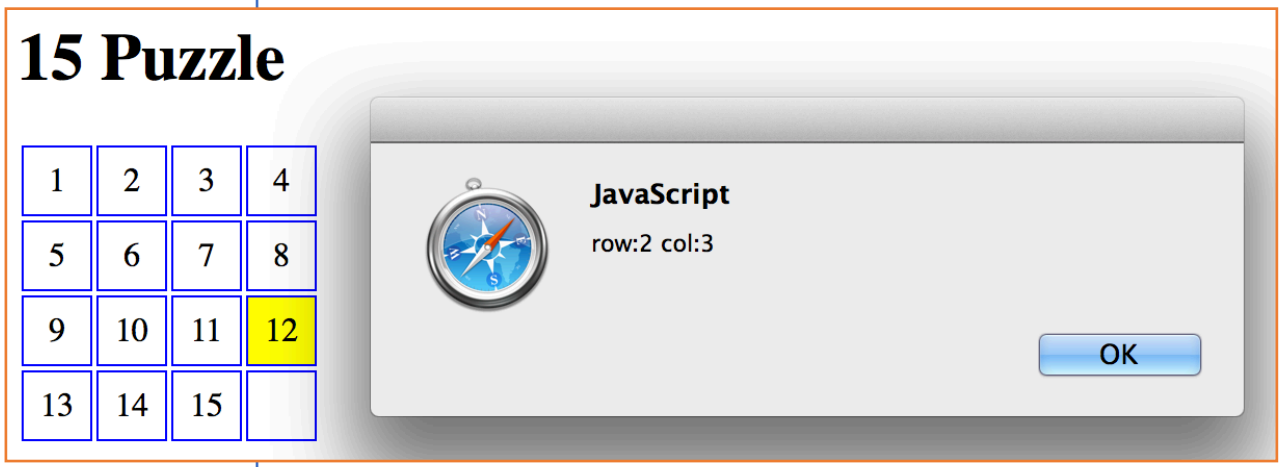
```
e.parentElement.rowIndex
```

column number:

```
e.cellIndex
```

Homework 7

```
1.      <style>
2.      td {
3.      border: 1px solid blue;
4.      width: 30px;
5.      height: 30px;
6.      text-align: center;
7.      vertical-align: middle
8.      }
9.      td:hover {
10.     background: yellow
11.     }
12.     </style>
13.     <script>
14.     function f(e) {
15.         var row = e.parentElement.rowIndex;
16.         var col = e.cellIndex;
17.         alert("row:" + row + " col:" + col)
18.     }
19.     </script>
```



Homework 7

- Table layout:

row: 0 col: 0	row: 0 col: 1	row: 0 col: 2	row: 0 col: 3
row: 1 col: 0	row: 1 col: 1	row: 1 col: 2	row: 1 col: 3
row: 2 col: 0	row: 2 col: 1	row: 2 col: 2	row: 2 col: 3
row: 3 col: 0	row: 3 col: 1	row: 3 col: 2	row: 3 col: 3

- You're going to have to do some arithmetic…
- Hint:

```
<script>
var empty_row = 3;
var empty_col = 3;
</script>
```

- **Example**: suppose the empty cell is row:2 col:1
- *Which cells can move into the empty cell?*
- Hint:
  - write a function can_move(e) that returns true/false depending on whether that number (when clicked) can move into the empty cell.
- Hint:
  - **`Math.abs(x)`** might be a useful method

Homework 7

- Your goal is to define that function f to simulate moving the tiles:
  - `<td onclick="f(this)">1</td>`
  - What is the *this* argument?

```
function f(e) {
    if (e.style.color == "red") {
        e.style.color = "black"
    } else {
        e.style.color = "red";
    }
}
```

JavaScript

[object HTMLTableCellElement]

- What can you do with the object?
  - set attribute values, e.g.
  - set content, e.g. `e.innerHTML = "3"`
  - find row and column numbers (*see earlier slides*)

Homework 7

```
<table id="puzzle">
  <tr>
    <td onclick="f(this)">1</td>
    <td onclick="f(this)">2</td>
    <td onclick="f(this)">3</td>
    <td onclick="f(this)">4</td>
  </tr>
  <tr>
    <td onclick="f(this)">5</td>
    <td onclick="f(this)">6</td>
    <td onclick="f(this)">7</td>
    <td onclick="f(this)">8</td>
  </tr>
  <tr>
    <td onclick="f(this)">9</td>
    <td onclick="f(this)">10</td>
    <td onclick="f(this)">11</td>
    <td onclick="f(this)">12</td>
  </tr>
  <tr>
    <td onclick="f(this)">13</td>
    <td onclick="f(this)">14</td>
    <td onclick="f(this)">15</td>
    <td onclick="f(this)"></td>
  </tr>
</table>
```

tricky!

- Note:
  - the innerHTML property of this TableCell is undefined!
  - i.e. there is no `document.getElementById("puzzle").rows[3].cells[3].innerHTML`
  - Solution: put a real "space" in there
  - can also use HTML nonbreaking space: ` `

Homework 7

- Attempt this before next class
- Next class, let's talk about residual problems you might have
- Due date (special):
  - next Wednesday midnight