

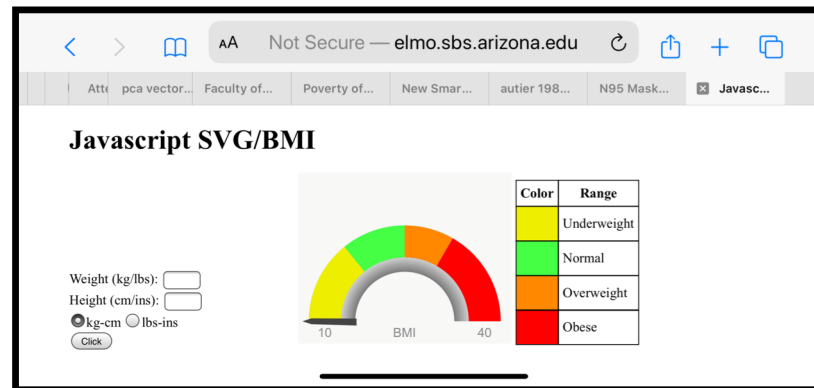
LING 408/508: Computational Techniques for Linguists

Lecture 21

Today's Topics

- Homework 7 Review
- Term projects

My iPhone:



The screenshot shows a mobile browser interface with a tab titled "Javascript SVG/BMI". The page content includes a BMI calculator with input fields for weight and height, a unit selector (kg-cm selected), and a "Click" button. A semi-circular gauge displays a BMI value of 10, with segments for Underweight (yellow), Normal (green), Overweight (orange), and Obese (red). A legend table is also present.

Color	Range
Yellow	Underweight
Green	Normal
Orange	Overweight
Red	Obese

- An example ... the parking lot game
- Javascript regex

Homework 7 Review

- Step by step:
 - Table cell clicking
 - `can_move()`
 - `test_move()`

Homework 7 Review

Step 1: test table cell clicking

```
1 <!DOCTYPE HTML>
2 <html> <head>
3 <title>15 Puzzle</title>
4 <style>
5 div { font-size: x-large }
6 table { border: medium solid blue }
7 td { border: 1px solid blue;
8     font-size: xx-large;
9     width: 90px;
10    height: 45px;
11    text-align: center;
12    vertical-align: middle
13  }
14 td:hover { background: yellow }
15 </style>
16 <script>
17 function f(e) {
18     var row = e.parentElement.rowIndex;
19     var col = e.cellIndex;
20     alert("row:" + row + " col:" + col)
21 }
22 </script>
23 </head>
```

```
24 <body>
25 <h1>15 Puzzle</h1>
26
27 <p>
28 <div>Tiles:
29 <button type="button" onclick="shuffle()">Shuffle</button>
30 <button type="button" onclick="location.reload()">Reset</button>
31 </div>
32 </p>
33 <p>
34 <table id="puzzle">
35   <tr>
36     <td onclick="f(this)">row: 0<br>col: 0</td>
37     <td onclick="f(this)">row: 0<br>col: 1</td>
38     <td onclick="f(this)">row: 0<br>col: 2</td>
39     <td onclick="f(this)">row: 0<br>col: 3</td>
40   </tr>
41   <tr>
42     <td onclick="f(this)">row: 1<br>col: 0</td>
43     <td onclick="f(this)">row: 1<br>col: 1</td>
44     <td onclick="f(this)">row: 1<br>col: 2</td>
45     <td onclick="f(this)">row: 1<br>col: 3</td>
46   </tr>
47 </table>
```

Homework 7 Review

Step 2: test can_move()

15 Puzzle

Tiles:

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

Can move row:3 col:2

Close

File: 15test2.html

```
16<script>
17var empty_r = 3;
18var empty_c = 3;
19function f(e) {
20    if (can_move(e)) {
21        var row = e.parentElement.rowIndex;
22        var col = e.cellIndex;
23        alert("Can move row:" + row + " col:" + col)
24    }
25}
26function can_move(e) {
27    var row = e.parentElement.rowIndex;
28    var col = e.cellIndex;
29    return ((row == empty_r && Math.abs(col - empty_c) == 1) ||
30            (col == empty_c && Math.abs(row - empty_r) == 1));
31}
32</script>
```

Homework 7 Review

File: 15test3.html

Step 3: test move()

15 Puzzle

Tiles:

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



15 Puzzle

Tiles:

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

```
16 <script>
17 var empty_r = 3;
18 var empty_c = 3;
19 function f(e) {
20     if (can_move(e)) {
21         move(e);
22     }
23 }
24 function can_move(e) {
25     var row = e.parentElement.rowIndex;
26     var col = e.cellIndex;
27     return ((row == empty_r && Math.abs(col - empty_c) == 1) ||
28             (col == empty_c && Math.abs(row - empty_r) == 1));
29 }
30 function move(e) {
31     // set empty cell non-empty
32     var t = document.getElementById("puzzle");
33     var tr = t.rows[empty_r];
34     var empty_cell = tr.cells[empty_c];
35     empty_cell.innerHTML = e.innerHTML;
36     empty_cell.style.border = "1px solid blue";
37     // set current cell to be new empty cell
38     e.innerHTML = "";
39     e.style.border = "initial";
40     empty_r = e.parentElement.rowIndex;
41     empty_c = e.cellIndex;
42 }
43 </script>
```

Homework 7 Review

- Getting a list of the cells:

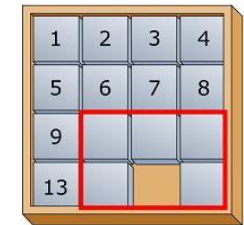
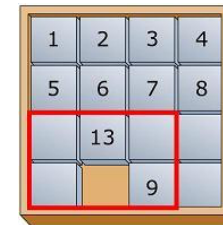
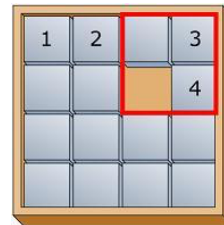
- `var cell_list = document.getElementById("puzzle").getElementsByTagName("td");`
- `var answer = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15];`

- Loop over `cell_list`:

- Check whether
 - `cell_list[i].innerHTML == answer[i]`

Homework 7 Review

- Term project idea:
 - animate solution
 - $16!/2 = 10,461,394,944,000$ valid configurations
 - i.e. write a function `solve()`
 - <https://cornellmath.wordpress.com/2008/01/27/puzzles-groups-and-groupoids/>



Stage 1: put 1 and 2 into position, get 3 and 4 into the corner
Stage 2: repeat for 5-8
Stage 3: put 9 and 13 into position
Stage 4: solve rest of the pieces

Term Project

Grading Scale and Policies

- homework exercises (50%)
- a term programming project (50%)

Time to start thinking about what you want to try to program up.

Still to come:

- Building and running the Apache webserver on your laptop
- Python programming (nlTK)

Parking Lot Game



- **Goal:**

- Get the X (red) car out
- Move the other vehicles out of the way
- Possible moves: forward or backwards

Parking Lot Game

- Demo ...



Javascript regular expressions

- Let's write our own browser-based tester to help you learn regular expressions:



String:

Regex: Global match (g):

form

- See: <http://elmo.sbs.arizona.edu/~sandiway/ling508-20/re-test.html>

Javascript regular expressions

String:

Regex: Global match (g):

`<input name=str>`

`<input name=g>`

`<input name=re>`

```
32 <body>
33 <form>
34   String: <input type=text name=str size=40>
35   <br>
36   Regex: <input type=text name=re size=40>
37   Global match (g): <input type=checkbox name=g value=g>
38   <input type=button value="Click" onclick="f(this)">
39 </form>
40 <script>
41   document.getElementsByTagName("form")[0].re.value = "Mr\. ([A-Z][a-z]*)"
42 </script>
43 <div id="output"></div>
44 </body>
45 </html>
```

Javascript regular expressions

```
1 <!DOCTYPE>
2 <html>
3 <head>
4 <script>
5   function f(e) {
6     var o = document.getElementById("output");
7     o.innerHTML = "";
8     var re_s = e.form.re.value;
9     var s = e.form.str.value;
10    if (re_s != "") {
11      var flag_s = "";
12      if (e.form.g.checked) {
13        flag_s += "g"
14      }
15
16      var regex = new RegExp(re_s, flag_s);
17      if (e.form.g.checked) {
18        var a;
19        while (a = regex.exec(s)) {
20          o.innerHTML += a.toString() + "<br>"
21        }
22      } else {
23        var m = s.match(regex);
24        if (m) {
25          o.innerHTML = m.toString()
26        }
27      }
28    }
29  }
30 </script>
```

What Javascript provides:

- RegEx object
 - **var re = new RegExp(string, flags)**
 - **var re = /[A-Z]([a-z)]*/gi**
(g= global; i=ignore case)
- Methods:
 - **var a = string.match(re)**
returns an array
 - [entire match, ...submatches...]
 - **var a = regex.exec(string)**
returns an array
 - different behaviors (under global flag)

optional

Javascript Regexp Tester

- Let's try the code:
 - <http://elmo.sbs.arizona.edu/~sandiway/ling508-20/re-test.html>

```
Mr. ([A-Z][a-z]*)
```

```
Mr\. ([A-Z][a-z]*)
```