WorkSheet 2 - DOM Scripting

February 6, 2013

1 Dom Scripting

Exercise 1.1 *Write a function that receives as input a node and a node type and outputs the number of its children that correspond to the given type.*

Exercise 1.2 *Write a function that receives as input a node and outputs an array with all the strings that occur within it as Text Nodes.*

Exercise 1.3 *Write a function that adds to end of the node given as its first input n empty divs where n is the number of originally existing divs.*

Exercise 1.4 *Write a function that receives as input a document object, a list of indexes $i_1, ..., i_n$ and removes from the document all the forms $i_1, i_2, ..., i_n$.*

Exercise 1.5 *Write a function that receives as input a document object, a list of indexes $i_1, ..., i_n$ and removes from the document the forms $i_1, i_2, ..., i_n$.*

Exercise 1.6 *Write a function that receives an Element node and reverses the order of its children. (Hint: use a DocumentFragment Node).*

Exercise 1.7 *Using the original event model, create an htlm page with a single button $b1$ and a script that alerts "Hello World" the first 30 times the button is clicked.*

Exercise 1.8 *Repeat the previous exercise, this time using $addEventListener$ and $removeEventListener$.*

2 jQuery

Exercise 2.1 *For each of the following cases write the appropriate selector:*

1. *All $\langle li \rangle$ that are direct children of an $\langle ol \rangle$*
2. *All the direct descendants of the element whose id is $\#output$*
3. *All $\langle p \rangle$ elements that come immediately after $\langle h1 \rangle$ inside a div.note*

4. *All headings* $\langle h1\rangle$, $\langle h2\rangle$, $\langle h3\rangle$, $\langle h4\rangle$, $\langle h5\rangle$ *and* $\langle h6\rangle$

Exercise 2.2 *Using the jQuery library, rewrite the following JavaScript programs:*

1. *P1:*

```
var elements = document.getElementByClassName("foo");
for(var i=0; i<elements.length; i++) {
   elements[i].className += "selected";
}
```

2. *P2:*

```
var element = document.getElementById("wem");
```

3. *P3:*

```
var elements = document.getElementByClassName("bar");
```

4. *P4:*

```
var elements = document.getElementByTagName("p");
```

5. *P5:*

```
var elements = document.getElementByTagName("p");
```

Exercise 2.3 *For each of the following cases write the corresponding JS program (using the jQuery library):*

1. *Select the descendants of the element whose id is 'bar' that are of class 'foo'*
2. *Select all inputs that have an attribute 'name' with value 'username'*

Exercise 2.4 *For each of the following cases write the corresponding JS program (using the jQuery library):*

1. *Add the letter Z to the beginning of each heading* $\langle h1\rangle$
2. *Replace* $\langle hr\rangle$ *tags with* $\langle br\rangle$ *tags*
3. *Add a paragraph with an arbitrary string before the beggining of each* $\langle h1\rangle$

Exercise 2.5 *Write a script that adds a new div to the current document that contains a copy of every link in the document.*

Exercise 2.6

1. *Write a script that adds a new div to the current document (called index) where it lists as normal strings the text content of the* $\langle h1 \rangle$ *elements that occur in the document.*

2. *Modify the previous list so that each item in the list is a link, that when clicked, redirects you to the appropriate part of the document (that is, the part of the document where the corresponding heading occurs).*

Exercise 2.7 *Write a program that deregisters all the events that were registered in any of the document elements. You may assume that these events were registered using the jQuery bind() method.*

Exercise 2.8 *Write an html page with two buttons:* $\langle b1 \rangle$ *and* $\langle b2 \rangle$*. When* $\langle b2 \rangle$ *is clicked, it should alert the string "b2 was clicked". When* $\langle b1 \rangle$ *is clicked, it should alert "b1 was clicked" and trigger all events registerd on* $\langle b2 \rangle$*.*

3 Performance

Exercise 3.1 *Rewrite the following JS function in order to increase performance.*

```
function initUI() {
  var bd = document.body,
      links = document.getElementByTagName("a");
      i=0;
  while(i<links.length){
     update(links[i++]);
  }
  document.getElementById("go-btn").onclick = function() {
     start();
  };
  bd.className = "active";
}
```

Exercise 3.2

1. *Implement the fibonacci function. Compute fib(100). Does it work? Why?*

2. *Implement the fibonacci function iteratively.*

3. *Write yet a new version of fib that caches intermidiate results.*

4. *Implement a function that takes a function f as an argument and returns a new version of that function that chaches results.*