

Name (please print): \_\_\_\_\_ ID #: \_\_\_\_\_  

First
Last

- This is a 50-minute closed-book exam. There are 3 questions worth a total of 90 points (everyone gets 10 free points). Budget your time so you get to all the questions.
 

1. \_\_\_\_\_  
 2. \_\_\_\_\_  
 3. \_\_\_\_\_
- Please read all of the instructions carefully before beginning to work on a problem, and if anything is not completely clear, please raise your hand and ask. Note that everything on this exam is JavaScript—no Perl allowed!
 

+ 10  
 Total.

**Question 1.** (15 points) Write a function, `shuffle`, that takes a single array argument, `a`, and returns an array consisting of the elements of `a` sorted in random order. (Each permutation of the elements of `a` should be equally likely.) The function must not modify the argument array `a` or introduce any global variables. (Reminder: every array has a `length` property, which is the number of elements it contains.)

**Question 2.** (15 points) Write a JavaScript function `Interval` that, when called, for example, as

```
intv = new Interval(4, 10);
```

returns an *object* with two properties, `left` and `right`, and one method, `inside`. The properties `left` and `right` should be initialized to the two arguments of `Interval`, and the `inside` method should take a single argument `n` and return `true` or `false` to indicate whether or not `n` is inside the interval (i.e., is greater than or equal to `left` and less than or equal to `right`). Thus, after executing the assignment to `intv` above,

- `intv.left` is 4,
- `intv.right` is 10,
- `intv.inside(4)` and `intv.inside(8)` are true, and
- `intv.inside(3)` and `intv.inside(10.1)` are false.

**Question 3.** (60 points) Write complete HTML/JavaScript files for the following application. When the page `digits.html` is loaded, a window with three frames appears as suggested below:

<input type="text" value="0"/> <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/> <input type="text" value="4"/> <input type="text" value="5"/>					
<input type="button" value="Restart"/>	<input type="button" value="Update"/>	Result	<input type="text" value="012345"/>	Digits: 012345	

The top frame consists of six images, all of which initially have source `x.gif`; the bottom-left frame consists of a Restart button, an Update button, and a six-character textfield; and the bottom-right frame displays an HTML document, initially blank (i.e., the document consisting of `<html><body></body></html>`). After the document loads, and also whenever the user pushes the Restart button, the six image files `0.gif`, `1.gif`, ..., `5.gif` are assigned to the six images in the top frame in numerical order (as shown above), and the corresponding string of digits is displayed in both the textbox and as part of an HTML document created “on the fly” in the lower-right frame. If the user clicks on any of the images, the contents of that image and the one to its right are swapped (the last image, if clicked, is swapped with the first one). Clicking the Update button updates the contents of the textbox and the lower-right frame so that they correspond with the images appearing in the top frame. Thus, after clicking the 1 and the 5 in the upper frame and then clicking Update, the resulting frames are:

<input type="text" value="5"/> <input type="text" value="2"/> <input type="text" value="1"/> <input type="text" value="3"/> <input type="text" value="4"/> <input type="text" value="0"/>					
<input type="button" value="Restart"/>	<input type="button" value="Update"/>	Result	<input type="text" value="521340"/>	Digits: 521340	

In your solution, make sure you clearly label your code with the file it's in. When giving the HTML for the top frame, you may save writing by giving just one of the images and indicating what changes are needed for the rest. (Hint: notice that the Restart and Update operations share quite a bit of functionality.)