Initial Report

Shuffle colleciton/array

March 27th 2017, 3:40 pm MDT

Q1 - Choose which version you prefer

#	Answer	%	Count

1	shuffle(obj) {
	  var set = isArrayLike(obj) ? obj : _.values(obj);
	  var length = set.length;
	  var shuffled = Array(length);
	  for (var index = 0, rand; index < length; index++) {
	   rand = _.random(0, index);
	   if (rand !== index) shuffled[index] = shuffled[rand];
	    shuffled[rand] = set[index];
	  }
	  return shuffled;
	}
		72.73%	40

2	shuffle(obj) {
	  var set = isArrayLike(obj) ? obj : _.values(obj);
	  var l = set.length;
	  var shuffled = Array(l);
	  for (var i= 0, r; i< l; i++) {
	   r = _.random(0, i);
	   if (r!== i) shuffled[i] = shuffled[r];
	    shuffled[r] = set[i];
	  }
		27.27%	15

	return shuffled; }		
	Total	100%	55

Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count	Bottom 3 Box	Top 3 Box
Choose which version you prefer	1.00	2.00	1.27	0.45	0.20	55	100.00%	100.00%

Q7 - Why?

Why?
the variables are more informative
More informative names
שמות משתנים בעלי משמעות
מבינים יותר מה זה עושה כי השמות יותר משמעותיים ויש יותר רווחים
למשתנים יש שמות יותר מובנים
הקוד הימני קצר יותר, יותר נוח להבנה (אין שמות משתנים ארוכים, שמסרבלים את הקריאה)
i like a minimized code espesially when it is known what each variable mean because it is the return of a meaningful function
lenght makes the code more readable
clearer
better variables names
miningfull variables name
Var names are more meaningful
שמות יותר מפורטים
יותר מסודר
כי שמות המשתנים יותר ברורים
basically because lines seem to be shorter - easy to read
the variable names are nore explained
the name of variables are better
more informative names
documentation of length and index
names are nore informative
names
formative names
real names and not letters
כי צריך לכתוב פחות טקסט
שמות משתנים מפורטים
variable are clearer to understand
prefer full words

at every line the variables are informative
more indicative
easier to understand
The lines shoreter and readable
Even though i and l are common shorthands, I prefer the more indicative names.
שמות משתנים מלאים
ברור מה מסמנים המשתנים, אין צורך להאריך
משתנים יותר אינפורמטיביים
משתנים בעלי משמעות
names are better defiend
קצר יותר ומובן יותר
שמות המשתנים ברורים
הקיצורי משתנים בגרסה שלא בחרתי מקשים לי לעקוב אחרי הקוד
אני לא מבין את הסינטקס

Q8 - Choose which version you prefer

#	Answer	%	Count

1	shuffle(obj) {
	  var set = isArrayLike(obj) ? obj : _.values(obj);
	  var length = set.length;
	  var shuffled = Array(length);
	  for (var index = 0, rand; index < length; index++) {
	   rand = _.random(0, index);
	   if (rand !== index) shuffled[index] = shuffled[rand];
	    shuffled[rand] = set[index];
	  }
	  return shuffled;
	}
		71.70%	38

2	shuffle(obj) {
	  var o = isArrayLike(obj) ? obj : _.values(obj);
	  var l = o.length;
	  var s = Array(l);
	  for (var i= 0, r; i< l; i++) {
	   r = _.random(0, i);
	   if (r!== i) s[i] = s[r];
	    s[r] = o[i];
	  }
	  return s;
	}
		28.30%	15

	Total	100%	53

Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count	Bottom 3 Box	Top 3 Box
Choose which version you prefer	1.00	2.00	1.28	0.45	0.20	53	100.00%	100.00%

Q9 - Why?

Why?
same reason as before
More informative names
שמות בעלי משמעות
אותו נימוק שמיקודם , זה אותם קודים אבל הפוכים
למשתנים יש שמות יותר מובנים
הקוד השמאלי- מאותה הסיבה
same as before
the names of variables makes the code understood
clearer
better variables name
miningfull variable names
Var names + function names are more meaningful
שמות יותר אינדיקטיביים
יותר יפה פחות מסובך
יותר ברור למי שקורא אותו
vars are more explainary and easy to understand
השני ארוך מדי
the variables name are better
more informative names
documentation
names are more informative
use of name
formative names
real names and not letters
פחות אותיות
גם יותר מפורט בשמות משתנים
same reasomn
prefer full words, more understandable

againg every line has info that you can understad with out reading the above lines
same as before
readable code
The lines shorter and readable
Again, I prefer more indicative names (even though I don't understand the name set).
שמות משתנים מלאים
שמות קצרים אם המשתנים ברורים (ותפקידם כמובן)
בגלל שאינטואיטבית הפונקצייה יותר ברורה לי
גם כאן - משתנים בעלי משמעות
better names values
קצר יותר ומובן יותר
שמות המשתנים ברורים
אותה הסיבה כמו מקודם. קיצורי משתנים מקשים במעקב

Q10 - Choose which version you prefer

#	Answer	%	Count

1	shuffle(obj) {
	  var set = isArrayLike(obj) ? obj : _.values(obj);
	  var l = set.length;
	  var shuffled = Array(l);
	  for (var i= 0, r; i< l; i++) {
	   r = _.random(0, i);
	   if (r!== i) shuffled[i] = shuffled[r];
	    shuffled[r] = set[i];
	  }
	  return shuffled;
	}
		81.82%	45

2	shuffle(obj) {
	  var o = isArrayLike(obj) ? obj : _.values(obj);
	  var l = o.length;
	  var s = Array(l);
	  for (var i= 0, r; i< l; i++) {
	   r = _.random(0, i);
	   if (r!== i) s[i] = s[r];
	    s[r] = o[i];
	  }
	  return s;
	}
		18.18%	10

	Total	100%	55

Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count	Bottom 3 Box	Top 3 Box
Choose which	1.00	2.00	1.18	0.39	0.15	55	100.00%	100.00%

version you prefer								

Q11 - Why?

Why?
More informative names when neccessary
יותר קרי
למשתנים יש שמות יותר מובנים
הפעם שמות המשתנים ארוכים, אבל בסך הכול הקוד כתוב מסודר וקל להבנה, לכן העובדה שהם אינפורמטיבים עוזרת
i think it is important to name shuffle because you can understand its meaning when you define, only later when you assign value to it
clearer
better variables names
because set insted of o
same
שמות יותר אינדיקטיביים
more indicative
יותר ברור למי שקורא
vars are easier to understand
הלולאה נראית ארוכה מדי
shuffled and set is a better name then s and o
more informtive names
names are more informative
use if names
כנל
like above
The names more informative
Even if most variables names aren't indicative, it is better to have a few indicative names than none.
אינפורמטיבי ולא מסורבל בעין
למשתנים יש משמעות
מרגישה מסודרת יותר
its only short and not mixed like the other one
קצר יותר וומובן יותר
פירוט שמות המשתנים

כאן מקצרים את הגריסה שלא בחרתי אפילו יותר ואני לא מסתדר עם זה
השמות של המשתנים יותר ברורים

Q12 - Choose which version you prefer

#	Answer	%	Count
1	var compare = function(first,second, firstStack, secondStack) { if (first == null || second == null) return first === second; if (first !== first) return second !== second; return deepCompare (first, second, firstStack, secondStack); }; var deepCompare = function(first, second, firstStack, secondStack) { var className = toString.call(first); if (className !== toString.call(second)) return false; var areArrays = className === ' [object Array]'; if (!areArrays) { if (typeof first != 'object' || typeof second != 'object') return false; var firstCtor = first.constructor, secondCtor = second.constructor; if (firstCtor !== secondCtor && ! (_.isFunction(firstCtor) && firstCtor instanceof firstCtor && _.isFunction(secondCtor)	72.73%	40

	&& secondCtor instanceof secondCtor) && ('constructor' in first && 'constructor' in second)) {
	return false;
	}
	}
	
	firstStack = firstStack || [];
	secondStack = secondStack || [];
	var length = firstStack.length;
	while (length--) {
	if (firstStack[length] === first) return secondStack[length] === second;
	}
	firstStack.push(first);
	secondStack.push(second);
	if (areArrays) {
	
	length = first.length;
	if (length !== second.length) return false;
	while (length--) {
	if (!compare(first[length], second[length], firstStack, secondStack)) return false;
	}
	} else {
	var keys = _.keys(first), key;
	length = keys.length;
	if (_.keys(second).length !== length) return false;
	while (length--) {
	key = keys[length];
	if (!(_.has(second, key) && compare(first[key], second[key], firstStack,		

	secondStack))) return false;
	}
	}
	firstStack.pop();
	secondStack.pop();
	return true;
	};		

2	var compare = function(a,b, aStack, bStack) {
	  if (a == null || b == null) return a === b;
	  if (a !== a) return b !== b;
	  return deepCompare (a, b, aStack, bStack);
	};
	
	var deepCompare = function(a, b, aStack, bStack) {
	var className = toString.call(a);
	if (className !== toString.call(b)) return false;
	
	var areArrays = className === '[object Array]';
	if (!areArrays) {
	if (typeof a != 'object' || typeof b != 'object') return false;
	var aCtor = a.constructor, bCtor = b.constructor;
	if (aCtor !== bCtor && !(_.isFunction(aCtor) && aCtor instanceof aCtor && _.isFunction(bCtor) && bCtor instanceof bCtor)
		27.27%	15

```
&amp;&amp;
p;
(&#39;constructor&#39;
in a &amp;&amp;
&#39;constructor&#39; in
b)) {
return false;
}
}

aStack = aStack || [];
bStack = bStack || [];
var length =
aStack.length;
while (length--) {
if (aStack[length] === a)
return bStack[length] ===
b;
}
aStack.push(a);
bStack.push(b);
if (areArrays) {

length = a.length;
if (length !== b.length)
return false;
while (length--) {
if (!
compare(a[length],
b[length], aStack, bStack))
return false;
}
} else {
var keys = _.keys(a),
key;
length = keys.length;
if (_.keys(b).length !==
length) return false;
while (length--) {
key = keys[length];
if (!(_.has(b, key)
&amp;&amp;
compare(a[key], b[key],
aStack, bStack))) return
false;
}
}
aStack.pop();
bStack.pop();
return true;
};
```

	Total	100%	55

Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count	Bottom 3 Box	Top 3 Box
Choose which version you prefer	1.00	2.00	1.27	0.45	0.20	55	100.00%	100.00%

Q13 - Why?

Why?
למשתנים יש שמות יותר מובנים
השמאלי- שוב מהסיבה של קיצור שמות המשתנים
i prefer second and first , because it is a lengthy code anyway , so why not write first and second instead of 'a' and 'b'
when the function is long, i prefer short names
better variables name
variable names are miningfull
They're pretty much the same, this one looks a bit cleaner
שמות יותר אינדיקטיביים
more words less signs
יותר ברור למי שקורא
now i would choose the shorter one - less intimidating TBH
נראה יותר מרווח ומוסבר יותר טוב מבחנית שמות משתנים
first and second is better, becuase i know for sure who is the first are and how is second (a and b is not 100%)
the names are better,the indantaion and one liner not so much
use of names
יותר מובן
כנל
clearer
shorter but still very readable
shorter and the second doesn't give relevant information
THe identation and the informative names
Even here I prefer first and second over a and b which seem arbitrary.
ארוך ולכן אני מעדיף שמות אינפורמטיביים
המשתנים יותר ברורים לי
רק אחריה ניתן לעקוב
informative is the key preference
הפעם הקוד הימני נראה בררו יותר
קל להבין את הקוד כאשר שמות המשתנים קשורים לתפקידם

בגלל שהקוד כבר מתחיל להיות יותר ארוך אז הגריסה המקוצרת עם המשתנים המקוצרים עדיפה
קצר יותר