## Tuesday Morning Boost

* Profile
* Search

# Tuesday Morning Boost

* **⏱ 5 minutes**
* **✅ Completed**

Score

You have submitted your quiz.

You got 4 questions correct, and 0 question(s) still being graded.

Your score so far is 100 out of 100.

Retake Quiz

# Tuesday Morning Boost

* function someSort(array) {
* if (array.length <= 1) {
* return array;
* }
* let midIdx = Math.floor(array.length / 2);
* let leftHalf = array.slice(0, midIdx);
* let rightHalf = array.slice(midIdx);
* let sortedLeft = someSort(leftHalf);
* let sortedRight = someSort(rightHalf);
* return helperCombineFunction(sortedLeft, sortedRight);
* }

**Which sorting algorithm is the above function?**

Insertion Sort

Quick Sort

Bubble Sort

Merge Sort

Selection Sort

##### EXPLANATION

Merge Sort is a classic example of a "Divide and Conquer" algorithm. In other words, you keep breaking the problem (i.e., array) into smaller and smaller subproblems (i.e., subarrays) until the subproblems are so small that you trivially know the answer to them (e.g., an array of length 0 or 1 is already sorted). Once you have solved the subproblems, you can then combine those results--in this case, by merging the left and right subarrays--to find the solution to the larger problem that was previously subdivided.

* const origObj = {
* 'val1': 10,
* 'val2': 20,
* 'val3': 30
* };
* let copy1 = {...origObj};
* let copy2 = ...origObj;
* let copy3 = {origObj};
* let copy4 = origObj;

**Which of these is the correct way to duplicate an object using the spread operator?**

copy1

copy3

copy2

copy4

##### EXPLANATION

Only copy1 is a duplicate of origObj: the spread operator (...) unpacks the contents of origObj inside a new object ({ }) that is then assigned (=) to copy1.

* **What function do you call in an event handler on a submission of a form to stop the page from reloading?**

e.preventDefault()

window.reload()

e.target.value

e.Default()

##### EXPLANATION

Submitting a form triggers a page reload by default. e.preventDefault() prevents this default behavior from happening.

* **What is the command to generate a JWT random string? (Hint: use documentation)**

require("brypto").randomBytes(32).toString("hex");

require("crypto").randomBytes(32).toString("hex");

require("brypto").randomBytes(32).toArray("hex");

require("crypto").randomBytes(64).toString("hex");

##### EXPLANATION

require("crypto").randomBytes(32).toString("hex"); uses the built-in crypto module to generate a random, 32-byte string in a hexadecimal value. This value can then be used in the creation of a JSON Web Token (JWT) to send user information securely between servers or across requests.