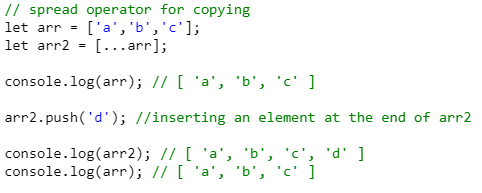
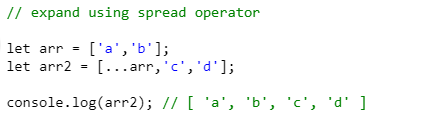
**Spread Operator**

It has multiple uses. In case of arrays we can expand, copy, concat, use with math object etc. It allows us the privilege to obtain a list of parameters from an array.

* In case of **copy an array** its best to use spread operator. Because when we copy array using the assignment operator, both the variable point at the same memory location. So, any changes made to one of those arrays will affect the other one. To make a separate copy, we should always use spread operator.

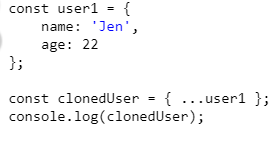


* In case of expanding an array it’s also best to use the spread operator. If we expand array using assignment operator, instead of getting the members of the array individually, we will get a whole array. This is unwanted. To fix this we should use spread operator.

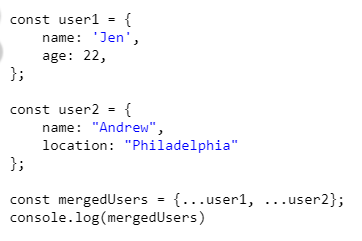


* Another case is during the uses of Math functions, if we pass array as parameter there it will not be able to process and give NaN. We have to use the spread operator to send the elements of the array as list.

**For Objects:** The spread operator (…) with objects is used to create copies of existing objects with new or updated values or to make a copy of an object with more properties.



We can copy multiple objects too same as arrays. here one thing to notice is, in case when both objects have same name property, it will take only one property and update the value from the last object. For an example,



**Output:**



**Rest Operator**

It has exactly same syntax as spread operator but does the opposite. It collects multiple elements and condense them in an array.

**When … used on the left side of an assignment, its rest operator. When it’s used on the right side its spread operator.**

We can also pass function parameters dynamically using the rest operator.

