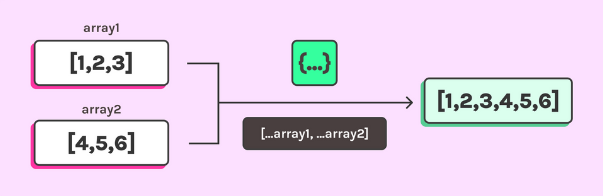
As a student I have quite often come across this not too much known operator called spread operator. As beginner you must be keen to know what it does and how it is use as was I as a student. In this blog I will make you understand what is spread operators use and its efficiency along with its syntax. For its practical implementation we will learn,

* How to use the spread operator to spread props in a react component.
* How to use the spread operator to update state in react
* Pros and cons of using the spread syntax to spread props.

**What is spread operator**

A spread operator spreads a multiple array of objects providing it with the structure of a single array which can consists of further arrays or objects or any property, Like copying multiple arrays into one array as a separate array.



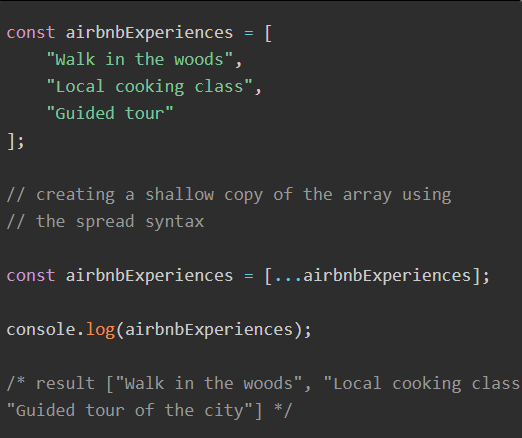
However if we talk in reference to React, spread operator spreads an object of props onto an element in available in React component.

**Syntax:**

A spread operator is denoted by three consecutive dots: … .

**Copy an array with spread**

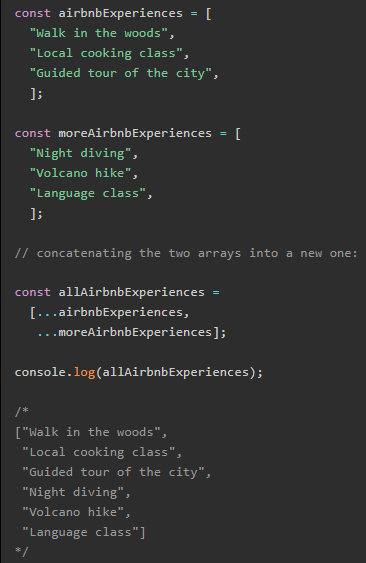
If you want to work on multiple arrays at a single time without effecting the original data of those multiple arrays this is where spread operator comes handy. A spread operator can create a shallow copy of array(s) you want to work on.



Although this is not something a developer is usually going for, however it is to show the efficiency of this operator.

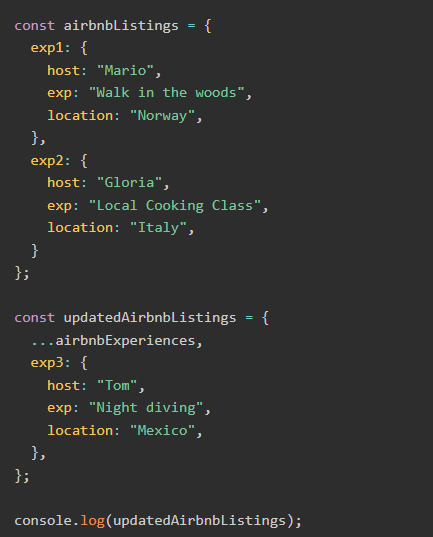
**Join arrays with spread**

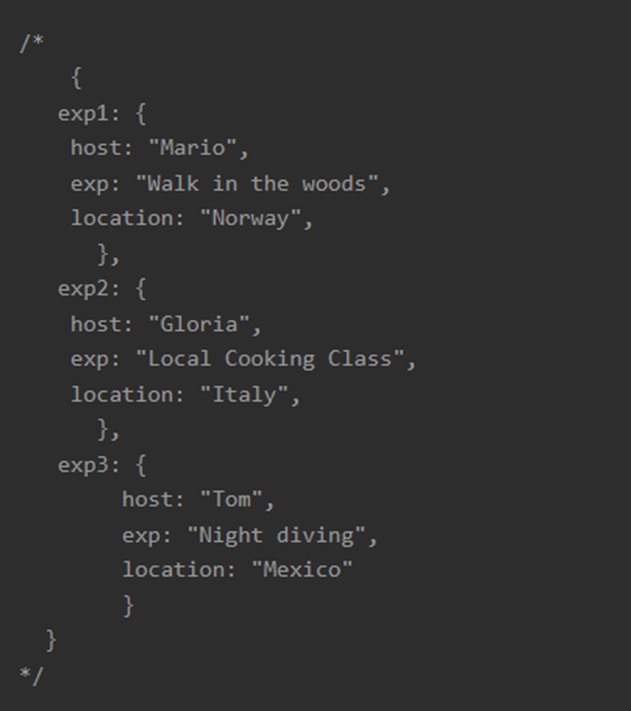
A developer often wants to concatenate multiple objects, arrays or any other property, for which the available approaches are either complex or time consuming, spread operator allows you to the join arrays or any other property with greater efficiency.



**Combine objects with spread**

Usually when a developer wants to add more elements or an array to ordinary array it occupies much more space and is less efficient as JavaScript does not make a copy of the object, it makes the new variable (or array item) contain the exact same object you assigned while the spread operator adds the element in the same array using its powerful spreading feature which ultimately makes your code more efficient, less complex and less time consuming.



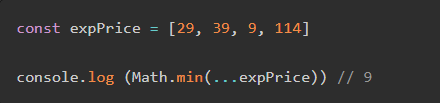


**Spread an array to function arguments**

When a developer is dealing statics in his/her code, they usually go for Math library which is efficient but to some extent since it cannot deal with handling function and array(s)



This where the efficiency of spread operator comes to use which can deal with the whole array and function as digits being used in the math library.



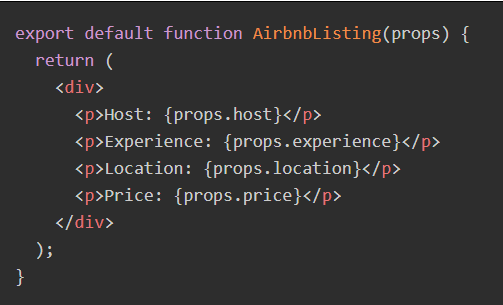
Now lets see the use of spread in React components

**How to use the spread operator to spread props in a React component**

Spread operator has great efficiency in many aspects although it can also be very useful when dealing with props. Props is dealt like an object which can also be spread using spread operator.

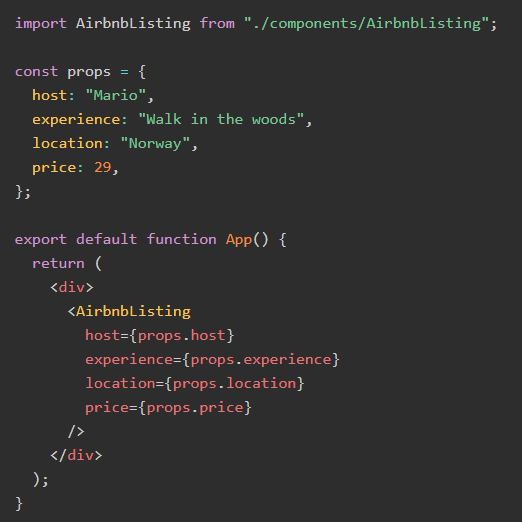
To understand its worth in this aspect we will build a component called AirbnbListing receiving props. Props is an object consisting of information about the user:

* **Host**
* **Experience**
* **Location**
* **Price**



Now to pass the props to the components we can either,

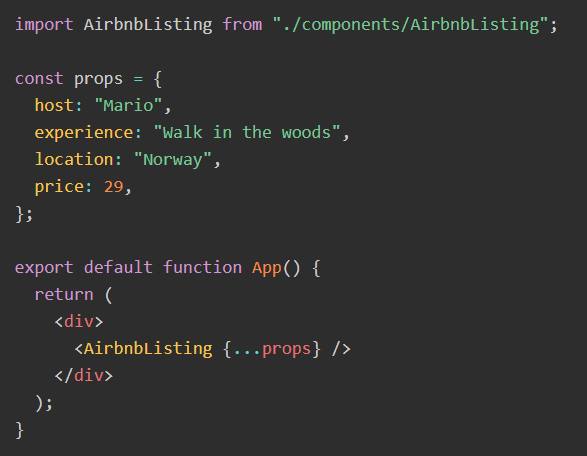
**1. Declaring the props one by one**



In this scenario the each entity about the user is being called into the AirbnbListing component which is time taking and put much more load on the server if the this information is coming in real time through API. And the prop. (…prop) Code is being written in the both files which increases the line of code doing the exact same thing.

Here spread operator comes and saves time and resources for the developer as well as the user.

**2. Using the spread syntax to pass the whole props object in one easy go**



Now in this second scenario, props is passed as a whole array itself by using the spread operator to the AirbnbListing function which is less complex, time consuming and a much more efficient approach.

Everything comes with its own pros and cons lets go through them now.

**Pros and cons of using the spread syntax to spread props**

**PROS**

* provides an efficient approach do the simple as well as complex tasks.
* Reduce the lines of code which saves time.
* Modify the components easily by just modifying the props.

**CONS**

* Makes the code difficult to understand at one glance.
* Lesser reusability of code.
* Less maintainability of code since it becomes hard to track which props a component is using when you use a spread operator

It all comes down to what approach you can use to make your code efficient and what you are most comfortable with while writing the code. However the use of this operator not only on itself but with the combination with other great operators/functions can open the door to new approaches to achieve the best written as well as usable code.