Spread Operator

এই অপারেটরটি ~~শুধু~~ যখন একটি অ্যারে বা অবজেক্টকে কপি করে তখন ~~শুধু~~ কেবলমাত্র 1st লেয়ার পর্যন্ত কপি করে।

≡ * আরও নেস্টেড অ্যারে বা অবজেক্ট থাকলে ~~সেগুলো কপি হয়~~, শুধু তাদের রেফারেন্স কপি হয়। একে শ্যালো কপি বলে।

>>> একই রেফারেন্স ধরে থাকে। একটির পরিবর্তন করলে অন্যটির পরিবর্তন হবে। নতুন কপিটি আগের রেফারেন্সের দিকেই পয়েন্ট করে থাকে।

Example:

```
const array1 = [[1,2,3],[4,5,6],[7,8,9]]
const array2 = [...array1]
array2[0][0] = 99;
console.log(array1)
console.log(array2)
```

এখানে দুটি অ্যারেই same আউটপুট দেয়।

ডিপ কপি ও শ্যালো কপির মধ্যে পার্থক্য:

(i) শ্যালো কপি: শুধু বাইরের লেয়ার কপি হয়, কিন্তু নেস্টেড অবজেক্ট বা অ্যারের রেফারেন্স একই থাকে।

(ii) ডিপ কপি: পুরো অবজেক্ট বা অ্যারে, তার নেস্টেড ডেটা সহ পুরোপুরি নতুন করে কপি হয়, তাই কোনো রেফারেন্স একই হয় না।

ডিপ কপির ২টি উপায়: (third party library দিয়েও করা যায়)

```
const obj1 = {
               name: "Mojnu",
               address: {
                          city: "Dhaka",
                          country: "Bangladesh"
                        }
             };

const obj2 = JSON.parse(JSON.stringify(obj1));
obj2.address.city = "Pabna";
Console.Log (obj1.address.city)  // Dhaka
```

*** প্রতিটি স্তরেই স্প্রেড অপারেটর ব্যবহার করে ডিপ কপি করতে পারি:

```
const obj1 = { name: "Asha Moni",
               address: {
                          city: "Kurigram",
                          country: "Bangladesh"
                        }
             }

const obj2 = { ~~name:~~ ...obj1,
               address: {
                          ...Obj1.address
                        }
             }
```

① প্রিমিটিভ টাইপ: নাম্বার, স্ট্রিং, বুলিয়ান, এই টাইপের ভেলু কপি করলে পুরোপুরি নতুন কপি হয়।

② রেফারেন্স টাইপ: অ্যারে, অবজেক্ট রেফারেন্স টাইপ। এই টাইপের ভেলু কপি করলে, একটি নতুন রেফারেন্স তৈরি হয়— যা একই অবজেক্ট বা অ্যারের দিকে নির্দেশ করে।

Rest Operator

...rest

- অজানা সংখ্যক আর্গুমেন্ট: যখন ~~আমা~~ আমরা জানি না, আমাদের ফাংশনে কতগুলো আর্গুমেন্ট পাস করা হবে, তখন (...rest) operator ব্যবহার করে সকল আর্গুমেন্টকে আমরা অ্যারেতে সংগ্রহ করে রাখতে পারি।

◇ rest operator দিয়ে আমরা অতিরিক্ত আর্গুমেন্টগুলোকে সহজে ম্যানেজ করতে পারি।

P.T.O.

- ☐ Spread Operator: অ্যারে বা অবজেক্টকে বিস্তার করে।
- ☐ rest Operator: একাধিক আর্গুমেন্টকে একটি অ্যারেতে সংগ্রহ করে রাখে।

Example - 1:

```
function   sum(...rest){
        const result = rest.reduce((total, currenElement)
                          ⇒ total + currentElement, 0);

        return result;

   }
   sum (1, 2, 3, 4, 7, 1, 3, etc)
```

Example - 2:

```
function  personDetails (firstName, LastName, ...otherInfo){
        ~~return~~   console.log(firstName + '  ' + lastName);
                     console.log(otherInfo);
   }

   personDetails ("Mojnu", "Miah", "Age: 25", "Lives
                   in Pabna", "wants a frontend job");
```

Example-3:

```
const {name,...otherDetails} = {
        name: 'Mojnu',
        age: 25,
        profession: "Developer",
        location: "Pabna",
}
```

Where you see in name and otherDetails variable?
comment...