۵ نکته در استفاده نکردن از حلقه For در جاوا اسکریپت : نه به تکرار (نسخه PDF)

یکی از مرسوم ترین روش ها برای کار کردن با مجموعه ها و آرایه ها این است که برای انجام عملیاتی مانند پیمایش و جستجو در آنها از حلقه تکرار یا loop استفاده کنیم. ولی در این مطلب می خواهیم بگوییم که در جاوا اسکریپت در موارد بسیاری از حلقه تکرار استفاده نکنیم. استفاده نکردن از حلقه ها مزایای زیادی دارد که از جمله آنها می توان به موارد زیر اشاره کرد:

۱. خوانایی کد بالا می رود.
۲. کد قابل فهم تر و قابل درک تر می شود.
۳. کد راحت تر خطایابی و debug می شود.

خب شاید سوال پیش آمده باشد که اگر از حلقه استفاده نکنیم کار حلقه را چگونه انجام دهیم که در ادامه به این سوال پاسخ می دهیم.

تغییر کل عناصر آرایه

مواردی وجود دارد که از حلقه استفاده می کنیم تا تغییری در عناصر آرایه به وجود بیاوریم. مانند کد زیر:

```
var names = ["Mehdi", "Adeli", "Hosein", "Ahmadi", "Java"];
var uperCases = [];
for (let i = 0; i < names.length; i++) {
  uperCases[i] = names[i].toUpperCase();
}
```

حال می توان کد بالا را بدون استفاده از حلقه نوشت که نتیجه کد زیر خواهد شد.

```
var names = ["Mehdi", "Adeli", "Hosein", "Ahmadi", "Java"];
var uperCases2 = names.map((name) => name.toUpperCase());
```

پیمایش عناصر آرایه

کد معمول و رایج پیمایش آرایه ها به شکل زیر است:

```
var names = ["Mehdi", "Adeli", "Hosein", "Ahmadi", "Java"];
for (let i = 0; i < names.length; i++) {
  console.log(names[i]);
}
```

حال شکل بدون حلقه این کد کد زیر خواهد بود

```
var names = ["Mehdi", "Adeli", "Hosein", "Ahmadi", "Java"];
names.forEach((name) => console.log(name));
```

فیلتر کردن عناصر آرایه

برای این که عناصـری از آرایه که شـرط خاصـی را دارند را فیلتر کنیم و بقیه عناصر را کنار بگذاریم معمولا از کدی مانند کد زیر استفاده می کنیم:

```
function isOdd(number) {
  return number % 2;
}
var numbers = [1, 2, 3, 4, 5, 6];
var oddNumbers = [];

for (let i = 0; i < numbers.length; i++) {
  if (isOdd(numbers[i])) {
    oddNumbers.push(numbers[i]);
  }
}
console.log(oddNumbers);
```

اما شکل بدون حلقه این کد که خیلی هم کار را راحت تر می کند در زیر آورده شده است:

```
function isOdd(number) {
  return number % 2;
}
var numbers = [1, 2, 3, 4, 5, 6];
var oddNumbers = [];
var oddNumbers2 = numbers.filter((n) => isOdd(n));
console.log(oddNumbers2);
```

انجام عملیاتی با استفاده از عناصر آرایه

تصور کنیـد که یک آرایه داریـد و می خواهید مجموع عناصـر آن را به دست بیاورید برای این کار به صورت معمول از کد زیر اسـتفاده می شود.

```
var result = 0;
for (let i = 0; i < numbers.length; i++) {
  result = numbers[i] + result;
}
```

دقت کنید که آرایه numbers را در کدهای قبل تعریف کرده ایم و در این کد دیگر آن را تعریف نکرده ایم. حال کد بالا را به شکل زیر بدون حلقه می نویسیم.

```
result = numbers.reduce((r, n) => r + n, 0);
```

جستجو در آرایه

برای جستجو در آرایه به روش حلقه تکرار کد را به شکل زیر می نویسیم:

```
var names = ["Mehdi", "Adeli", "Hosein", "Ahmadi", "Java"];
for (let i = 0; i < names.length; i++) {
  if (names[i] === "Mehdi") {
      console.log("%c %s found ","color:red",names[i]);
     break;
  }
}
```

حال اگر بخواهیم همین جستجو را بدون حلقه بنویسیم کد آن به شکل زیر خواهد بود:

```
var names = ["Mehdi", "Adeli", "Hosein", "Ahmadi", "tosinso"];
var isExists=names.some(a=>a==="Mehdi");
if(isExists){
    console.log("%c %s found ","color:red","Mehdi");
}
```

آیا همه عناصر آرایه یک شرط را دارند یا خیر؟

برای این که مشـخص کنیم که عناصـر آرایه یک شرط معین را دارند یا خیر نیز می توان هم از حلقه استفاده کرد و هم استفاده نکرد. این حالت هم شبیه به جسـتجو در آرایه است. مثلا برای این که مشخص کنیم که همه عناصر یک آرایه اعداد مثب هستند کد آن به شکل زیر خواهد بود:

```
or (let i = 0; i < numbers.length; i++) {
    if(numbers[i]<0){
        console.log("there is negative number in array");
    }
}
```

برای حالت بدون حلقه این کد از تابع every به شکل زیر استفاده می کنیم:

```
const positive = numbers.every(a => a > 0);
if(!positive){
    console.log("There is negative number in Array");
}
```

شایـد سوال پیش آمده باشد که در حلقه ها امکانی مانند continue , break داریم ولی توی این توابعی که گفته شـد چی؟ در جواب بـاید گفت که با استفاده از توابع some و every می توان همان نتیجه را گرفت.با وب سایت Tosinso همراه باشید

نویسنده: مهدی عادلی فر

مطلب اصلی