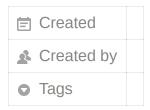
Metody na tablicach - array.map()

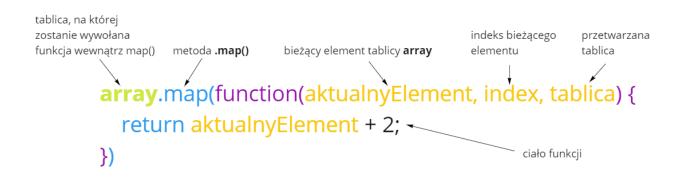


Wstęp

Dzisiaj przeanalizujemy w jak (łatwy!) sposób można dokonywać zmian na elementach tablicy za pomocą metody .map(). Nie musi Cię przerażać słowo "metoda" użyte w tytule, gdyż oznacza ono w tym przypadku funkcję.



Po krótkim słowie wstępu - metoda .map() jest funkcją, która poprzez iterację na elementach tablicy zwróci nową(zmienioną) tablicę. Co to oznacza w praktyce - pierwotna tablica pozostaje nienaruszona. Przejdźmy zatem do omówienia składni przy użyciu .map().



Omówienie składni:

- array czyli tablica na której robimy metodę .map(),
- .map() metoda dodana po kropce obok tablicy; pozwala na jej wykonanie w odniesieniu do określonej tablicy array,
- function jeśli tworzymy normalną funkcję to zawsze w środku musi być słowo kluczowe; równie dobrze można w tym miejscu użyć składni arrow function (parametr obowiązkowy),
- aktualnyElement jest to wartość aktualnie iterowanego elementu z tablicy array (parametr obowiązkowy),
- index index aktualnie przetwarzanego elementu tablicy (parametr opcjonalny),
- tablica jest to tablica na której użyliśmy naszej metody. Możemy dzięki temu mieć do niej dostęp w trakcie kolejnych wywołań metody (parametr opcjonalny),
- ciało funkcji to instrukcje, które będą wykonywane dla każdego elementu przetwarzanej tablicy array.



Funkcja użyta wewnątrz metody **.map()**, to tzw. *callback function* (funkcja z wywołaniem zwrotnym). Tą nazwą określa się funkcję przekazywaną wewnątrz innej funkcji jako parametr.

Jeżeli "cała ta wiedza" jeszcze nie do końca jest zrozumiała... to spokojnie - przypomnij sobie pętlę **for**, którą wykonywałeś na tablicy. Na początku przetwarzany był pierwszy element, można było na nim wykonać jakieś działanie, "wepchnąć" go do nowej tablicy, a po wykonaniu bloku kodu zwiększany był licznik wywołań pętli i instrukcje były wykonywane dla kolejnego elementu - tak samo jest w przypadku **.map()**, tylko krócej ;)

Przykład:

```
const array = [1, 2, 3, 4, 5]

const newArray = array.map(function(item) {
   return item * 2
```

```
})
console.log('array', array) // Wynik to "array", [1, 2, 3, 4, 5]
console.log('newArray', newArray) // Wynik to "newArray", [2, 4, 6, 8, 10]
```

Omówienie przykładu:

- array.map() na naszej zadeklarowanej wcześniej tablicy array wywołujemy funkcje map()
- function(item) { return item * 2 } funkcja map() przyjmuje jako pierwszy parametr inną funkcję, która wykona się podczas iterowania po elementach tablicy. W tym przypadku nasz "item" to jest każdy element tablicy, w składni jest to "aktualnyElement",
- const newArray = array.map() aby odczytać wynik przypisujemy to co nam da nasza funkcja do zmiennej,
- II Funkcja .map() tworzy <u>nową tablicę</u> przetwarzając jedynie elementy tablicy, na której została zastosowana. Pierwotna tablica pozostaje niezmieniona, czego dowodzi powyższy przykład.

WAŻNA INFORMACJA DO ZADAŃ!!!

Do każdego artykułu utwórzcie sobie na komputerze folder i w nim pliki z rozwiązaniami zadań. Zawsze, nawet jeśli pytanie jest otwarte i czysto teoretyczne to warto napisać na nie odpowiedź by nasz mózg lepiej zapamiętał to co piszemy.

Legenda:

bez gwiazdek - zadania podstawowe z artykułów oraz filmu,

- * na podstawie artykułu, ale nie znajdziesz tego na filmie,
- ** nie ma tego w artykule ani w filmiku ale dostajesz link gdzie znajduje się rozwiązanie,
- *** brak informacji w artykule, filmie oraz brak linku do rozwiązania. Są to najbardziej rozwijające zadania, które również uczą samodzielności :)
- 1. Utwórz zmienną, do której będzie przypisana tablica [5,10,15] a następnie za pomocą metody **map()** utwórz nową tablicę, której rezultatem będzie [10,15,20]. Wyświetl w consoli tablicę.
- 2. Stwórz tablicę zawierającą imiona ludzi z Twojej grupy na kursie Devstock. Dla każdej osoby wyświetl przywitanie w konsoli.
- 3. Do każdego z elementów tablicy [0,1,2,3,4] dodaj wartość odpowiądającego elementowi indeksu. Wyświetl w consoli przetworzoną tablicę.
- 4. Przetwórz tablicę [0, null, 'text'] w taki sposób, aby uzyskać informacje o tym, jakim typem danych jest każdy element z tablicy. Wyświetl w konsoli.
- 5. Wyświetl tablicę, która z tablicy [10,7,45,-34,23,-5] zwróci true, jeżeli liczba jest większa lub równa 10, a w każdym innym przypadku false.
- 6. Wyświetl tablicę, która zwracać będzie długość każdego z elementów tablicy ['Legolas', 'Gimli', 'Frodo', 'Aragorn','Sam'].
- 7. Dla tablicy celsjusz = [-20, -7, 0, 5, 12, 34, 56] stwórz odpowiednik w skali Fahrenheita.
- 8. * Zadeklaruj zmienną

```
array = [
{
   name: 'Andrzej',
   age: 32,
},
{
   name: 'Janek',
   age: 23,
},
{
   name: 'Witek',
   age: 35,
}
];
```

Następnie na tej tablicy użyj funkcji **map(),** aby uzyskać tablicę, której każdy element będzie wyglądać na wzór : "Andrzej jest w wieku 32 lat".

- 9. ** Za pomocą metody **map()** postaraj się uzyskać "odwróconą" tablicę w stosunku do tablicy [0,1,2,3], czyli [3,2,1,0].
- 10. ***Przy pomocy metody **.map()** spraw, aby tablica stringów array = ['0','1.44','2','3.14','4'] została zamieniona w tablicę numerów.