

2025-01-09

Masyvai

1. Sukurkite masyvą `colors`, kuriame bus tokie elementai:

'red', 'magenta', 'violet', 'blue', 'green', 'yellow', 'orange', 'purple', 'pink', 'brown', 'black', 'white', 'gray', 'cyan', 'teal', 'indigo', 'lime', 'navy', 'gold', 'silver'

2. Sukurkite funkciją `first(masyvas)`, kuri gražina **pirmojo** elemento masyve reikšmę;

TEST:

```
console.log(first(colors)); // 'red'
```

3. Sukurkite funkciją `last(masyvas)`, kuri gražina **paskutinio** elemento masyve reikšmę;

TEST:

```
console.log(last(colors)); // 'silver'
```

4. Sukurkite funkciją `secondLast(masyvas)`, kuri gražina priešpaskutinio elemento masyve reikšmę bei patikrina ar masyvas turi bent 2 elementus. Jei masyve yra mažiau nei 2 elementai, gražinkite reikšmę `false`, kitu atveju gražinkite priešpaskutinį elementą;

TEST1:

```
console.log(secondLast(colors)); // 'gold'
```

TEST2:

```
console.log(secondLast(['tekstas'])) // false
```

5. Sukurkite funkciją `doesExist(masyvas, reiksme)`, kuri atlieka patikrinimą, ar parametre masyvas egzistuoja reiksme. Priklausomai nuo to ar egzistuoja, gražinti tekstinę reikšmę *"Rastas elementas masyvo indeksu X"* arba *"Elementas masyve buvo nerastas"*

TEST1:

```
console.log(doesExist([1,2,3], 4)); // "Elementas masyve buvo nerastas"
```

TEST2:

```
console.log(doesExist([1,2,3], 2)); // "Rastas elementas masyvo indeksu 1"
```

TEST3:

```
console.log(doesExist(colors, 'yellow')); // "Rastas elementas masyvo indeksu 5"
```

TEST4:

```
console.log(doesExist(colors, 'infrared')); // "Elementas masyve buvo nerastas"
```

6. `console.log`inkite visas spalvas, atskirtas kableliu su tarpu, rezultatas konsolėje turėtų atrodyti taip:

'red, magenta, violet, blue, green, yellow, orange, purple, pink, brown, black, white, gray, cyan, teal, indigo, lime, navy, gold, silver'

7. `console.log`inkite darbuotojų masyvą `employees`, kurį sugeneruosite iš šio teksto:

```
'Alice_ . _Bob_ . _Charlie_ . _David_ . _Eve_ . _Frank_ . _Grace_ . _Hannah_ . _Isaac_ .  
_Jack_ . _Karen_ . _Liam_ . _Mia_ . _Nathan_ . _Olivia_ . _Peter_ . _Quinn_ . _Rachel_ .  
_Sophia_ . _Tom'
```

Rezultatas konsolėje turėtų atrodyti taip:

```
[  
  'Alice', 'Bob', 'Charlie', 'David', 'Eve',  
  'Frank', 'Grace', 'Hannah', 'Isaac', 'Jack',  
  'Karen', 'Liam', 'Mia', 'Nathan', 'Olivia',  
  'Peter', 'Quinn', 'Rachel', 'Sophia', 'Tom'  
]
```

8. Sukurkite funkciją `createHtmlList(masyvas)`, kuri kuria HTML kodą. Kuriamas sunumeruotas sąrašas iš visų masyve esančių elementų. Raskite būdą, kaip protingai panaudoti `.join()` metodą bei teksto sujungimą, kad sudaryti HTML kodą iš darbuotojų sąrašo su viena eilute:

return rezultatas:

```
'<ol><li>Alice</li><li>Bob</li><li>Charlie</li><li>David</li><li>Eve</li><li>Frank</li><li>Grace</li><li>Hannah</li><li>Isaac</li><li>Jack</li><li>Karen</li><li>Liam</li><li>Mia</li><li>Nathan</li><li>Olivia</li><li>Peter</li><li>Quinn</li><li>Rachel</li><li>Sophia</li><li>Tom</li></ol>'
```

HTML dokumente sukurkite `sąrašas1` klasę turintį `div` elementą, kuriame pridėsite šio sunumeruoto sąrašo HTML. Rezultatas turi matytis pačiame puslapyje.

9. Patobulinkite funkciją `createHtmlList` pridėdami jai papildomą parametą `rezimas`.

`createHtmlList(masyvas, rezimas)`. Jei `rezimas` yra

`undefined`: grąžinkite tą patį numeruotą sąrašą

`ne skaičius`: grąžinkite tuščią string reikšmę: `""`

1: grąžinkite nenumeruotą sąrašą.

2: grąžinkite numeruotą sąrašą.