Դաս 02: JS for, while, do while Ցիկլեր

Ցիկլերը (կամ կրկնողությունները) թույլ են տալիս միևնույն գործողությունները կրկնել մի քանի անգամ, մինչև որ սահմանված պայմանը դառնա սխալ: **JavaScript**-ում կան երեք հիմնական տեսակի ցիկլեր.

- **for ցիկլ** - օգտագործվում է, երբ պետք է նախապես հայտնի քանակով կրկնել գործողությունները, օրինակ՝ հաշվել թվերը 1-ից մինչև 10:
- **while ցիկլ** - օգտագործվում է, երբ պետք է գործողությունները կրկնել, մինչեւ որ որոշակի պայմանը ճիշտ է մնում:
- **do...while ցիկլ** - նման է **while** ցիկլին, բայց գործողությունները կատարում է գոնե մեկ անգամ, անկախ պայմանից:

for ցիկլով թվերի հաշվարկ

```
for (let i = 1; i <= 5; i++) {
    console.log(i);
}
// Արդյունքը: 1, 2, 3, 4, 5
```

while ցիկլով հաշվել, մինչեւ թիվը հասնի 10-ի

```
let count = 1;
while (count <= 10) {
  console.log(count);
  count++;
}
// Արդյունքը: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

do...while ցիկլով գուշակելու խաղ

```
let number;
do {
  number = Number(prompt("Մուտքագրեք թիվը (1-5):"));
```

```
} while (number < 1 || number > 5);
alert("Շնորհավորում եմ, դիշոս քիվն ես մուտֆագրել!");
```

Պարզ թվերի որոշում (for ցիկլ):

```
function isPrime(n) {
    if (n <= 1) return false;
    for (let i = 2; i < n; i++) {
      if (n % i === 0) return false;
    }
    return true;
  }
  console.log(isPrime(7)); // true
```

Քանակային բազմապատիկների համար (for ցիկլ):

```
function logMultiplesOf3And5(n) {
    for (let i = 1; i <= n; i++) {
      if (i % 3 === 0 && i % 5 === 0) {
        console.log(i);
      }
    }
  }
  logMultiplesOf3And5(15); // 15
```

Թվերի գումարը մինչև n (while ցիկլ):

```
function sumTo(n) {
    let sum = 0;
    let i = 1;
    while (i <= n) {
      sum += i;
      i++;
    }
    return sum;
  }
  console.log(sumTo(5)); // 15
```

Օգտագործել երկու փոփոխական for ցիկլում

```
for (let i = 0, j = 5; i < j; i++, j--) {
    console.log(i, j);
  }
  // Արդյունքը: 0 5, 1 4, 2 3
```

continue և break օգտագործումը for ցիկլում:

- **continue** թույլ է տալիս անցնել հաջորդ իտերացիային՝ բաց թողնելով ներկա քայլը:
- **break** դադարեցնում է ցիկլը:

```
for (let i = 1; i <= 10; i++) {
    if (i % 2 === 0) continue; // Բաց թողնել զույգ թվերը
    console.log(i); // Ցույց տալ միայն կենտ թվերը
  }
```

Փոփոխականների կուտակում ֆունկցիայի ներսում:

```
function countDigits(n) {
    let count = 0;
    while (n > 0) {
      n = Math.floor(n / 10);
      count++;
    }
    return count;
  }
  console.log(countDigits(1234)); // 4
```

Ֆիբոնաչիի թվերի հաշվում for ցիկլով

```
function fibonacci(n) {
    let a = 0, b = 1;
    for (let i = 0; i < n; i++) {
      console.log(a);
      let next = a + b;
      a = b;
```

```
      b = next;
    }
  }
fibonacci(5); // Արդյունքը: 0, 1, 1, 2, 3
```

Ֆիբոնաչիի n-րդ թիվը հաշվում while ցիկլով

```
function fibonacciNth(n) {
    let a = 0, b = 1, count = 2;
    if (n === 1) return a;
    if (n === 2) return b;
    while (count < n) {
      let next = a + b;
      a = b;
      b = next;
      count++;
    }
    return b;
  }
  console.log(fibonacciNth(5)); // 3
```

Հանդիպումների քանակի հաշվում

- Առաջադրանք: Հաշվեք, թե քանի անգամ է հանդիպում որոշակի թիվ n-ից փոքր թվերի շարքում:

```
function countOccurrences(n, digit) {
    let count = 0;
    for (let i = 1; i <= n; i++) {
      let num = i;
      while (num > 0) {
        if (num % 10 === digit) count++;
        num = Math.floor(num / 10);
      }
    }
    return count;
```

```
  }
  console.log(countOccurrences(100, 5)); // Օրինակ՝
1-ից մինչև 100-ը թվերը 5 է հանդիպում 20 անգամ:
```

Ռեկուրսիան ֆունկցիայի կիրառման մեթոդ է, երբ ֆունկցիան ինքն իրեն է կանչում որոշակի պայմանով: Հիմնականում ռեկուրսիան օգտագործվում է խնդիրները լուծելու համար, երբ հնարավոր է դրանք բաժանել ավելի փոքր նույնանման խնդիրների:

Ռեկուրսիայի Կարճ Նկարագրություն

Ռեկուրսիվ ֆունկցիան սովորաբար ունի երկու հիմնական մաս.

1. **Հիմնական (Base) Պայման**՝ որոշում է, թե երբ պետք է կանգ առնել:
2. **Ռեկուրսիվ Կանչ**՝ երբ ֆունկցիան ինքն իրեն կանչում է, մոտենալով հիմնական պայմանին:

Ռեկուրսիայի Օրինակներ

1. Ֆակտորիալ (Factorial) հաշվարկ

Ֆակտորիալը հաշվարկում է n թվերի բազմապատկումը մինչև 1. Օրինակ՝ 5! = 5 * 4 * 3 * 2 * 1 = 120.

```
function factorial(n) {
    if (n === 0) return 1; // Հիմնական պայմանը
    return n * factorial(n - 1); // Ռեկուրսիվ կանչ
}

console.log(factorial(5)); // Արդյունքը: 120
```

2. Ֆիբոնաչիի n-րդ Թվանիշի Հաշվում

Ֆիբոնաչիի շարքը՝ 0, 1, 1, 2, 3, 5, 8,... յուրաքանչյուր հաջորդ թիվ հավասար է նախորդ երկու թվերի գումարին.

```
function fibonacci(n) {
    if (n <= 1) return n; // Հիմնական պայմանները n === 0 կամ n === 1 համար
    return fibonacci(n - 1) + fibonacci(n - 2); // Ռեկուրսիվ կանչ
}

console.log(fibonacci(5)); // Արդյունքը: 5
```

5. Թվերի Հակադարձում

Հակադարձում ենք թիվը՝ օրինակ 123 -> 321.

```
function reverseNumber(n, result = 0) {
    if (n === 0) return result; // Հիմնական պայմանը՝ եթե n-ը 0 է
    return reverseNumber(Math.floor(n / 10), result * 10 + (n % 10)); // Ռեկուրսիվ կանչ
}

console.log(reverseNumber(1234)); // Արդյունքը: 4321
```