



## 1.) Döngü deyimleri

\* for

\* while

\* do-while

\* for in, for of (non-primitive)  
dizi, obje

## 2.) Döngü Metotları (alt program)

\* forEach, map, filter, reduce, every ...

forEach(→) → kullanıyoruz  
↓  
döngü deyimleri

for  
100 kere

for (sayac ; koşul ; sayacın  
degisim  
miktarı) {

işlem 1  
işlem 2  
işlem 1

1 2 3 4 ... 100  
1 → 100

$\frac{1}{100-200}$   
 $\frac{i}{i <}$

1-100 kere  $\frac{0 \rightarrow 99}{true}$

for (let  $i=1$  ;  $i \leq 100$  ;  $i++$ ) {  
    clg("char+14")  
}

döngü  
göreb

$\frac{i}{1}$

koşul  
başlangıç  
bitiş → girilme  
true olduktan  
bu döngü çalışır.

her döngüye → Sonsuz  
döngü

$\frac{i}{23}$

true false  
 $(101 < 100)$

for (let  $i=1$  ;  $i \leq 100$  ;  $i++$ ) {  
    clg("char+14")  
}

clg("bye")

( $i=100$  ;  $i > 2$  ;  $i/=2$ )

$i--$        $i/=2$

$i*=3$

false       $i-=4$

1==50

for (let j=1 ; j==50 ; j+=2) {

Boolean (true)

}

for (let i=0; 1; i++) {

≡.      Soruşt

}

1-100

for (let i=1 ; i<=100; i++) {

≡ ~~break~~

if (i==49) { → conditions

} break; → içindeki değeri kırar.

if (i==3 || i==5) {

continue

}

log(i)

1 2 4 6 7 8 ... 48

$$\begin{array}{r} 5 \\ 1 + 2 + 3 + 4 + 5 \\ \hline \end{array}$$

$$\text{Sum} = 0$$

$$\begin{array}{r} \text{Sum} = 0 \\ 1 \quad 0+1 \\ \text{Sum} = \text{Sum} + 1 \\ 3 \quad 1+2 \\ 6 \quad 3+3 \\ 10 \quad 6+4 \\ 15 \quad 10+5 \end{array}$$

Dongu sayacı  
while ( kosul ) {

=====  
dongu sayacı, fonksiyon

}

① let sayac = 1 (Dongu Sayacı)

while ( sayac ② ≤ 100 ) {

clg ( "\$ {sayac} . ~~Kisi~~ , Ahmet"

③ sayac ++

}

for ( let sayac = 1; sayac < 100; sayac++ )

- Aktli (Rakamsal)  
daha pratik (✓)

- for syntax,

- Karmaşık koşulların iterasyonu için  
while döngüsü daha pratik.

while (koşul) {  
    ≡  
    ≡  
}

≡  
≡  
≡

run  
while

X

for

≡  
≡  
≡

}

while (koşul);

do {  
    ≡  
    ≡  
    ≡  
    ≡  
    ≡  
}

en az  
1 kez  
çalışır

run  
~~do-while~~ (E)