Insider | USEINSIDER.COM

CodeCraft
Bootcamp'25

Javascript

Loops, Conditionals, Functions & Scope, Exception Handling

What is loop?

Döngü, belirli bir koşul sağlandığı sürece kodun tekrar çalışmasını sağlayan bir yapıdır. Tekrarlayan işlemleri otomatikleştirerek kod tekrarını azaltır ve programın daha verimli çalışmasını sağlar.

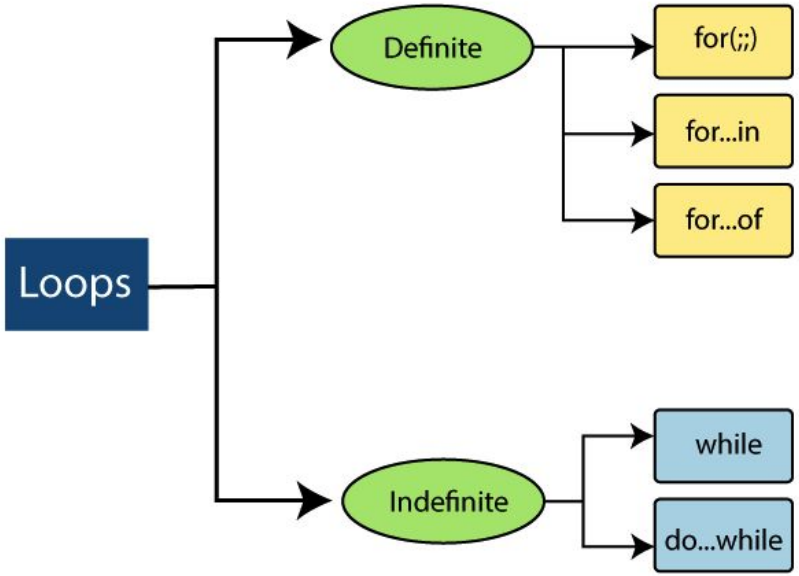
```
const students = ["Elena", "Baris", "Maria", "Dave"]

for(let index = 0; index < students.length; index++) {
  console.log('Name of student is: ' + students[index]);
}

// Name of student is: Elena
// Name of student is: Baris
// Name of student is: Maria
// Name of student is: Dave
```

Loop Types

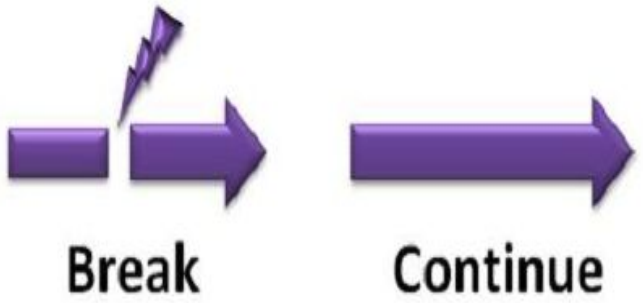
- **for:** Belirli bir sayıda işlem yapmak için
- **for..of:** Diziler ve iterable nesneler için
- **for...in:** Objelerin key'leri için
- **while:** Belirli bir koşul sağlanana kadar çalıştırmak için
- **do..while:** Koşul false bile olsa en az bir kez çalıştırmak için



Break & Continue

Break: komutu bir döngüyü belirli bir koşul gerçekleştiğinde anında sonlandırır. Bu sayede döngü, normal bitiş koşuluna ulaşmadan durdurulabilir.

Continue: komutu ise mevcut yinelemeyi (iteration) atlayarak döngünün bir sonraki turuna geçmesini sağlar. Böylece belirli bir koşul sağlandığında işlemler atlanabilir ancak döngü çalışmaya devam eder.



What are conditional statements?

Koşul ifadeleri, belirli bir koşulun doğru (true) veya yanlış (false) olup olmamasına bağlı olarak kodun farklı yollar izlemesini sağlayan yapılardır.

- if, else if, else Koşulları
- Ternary Operator (? :)
- Switch-Case Yapısı

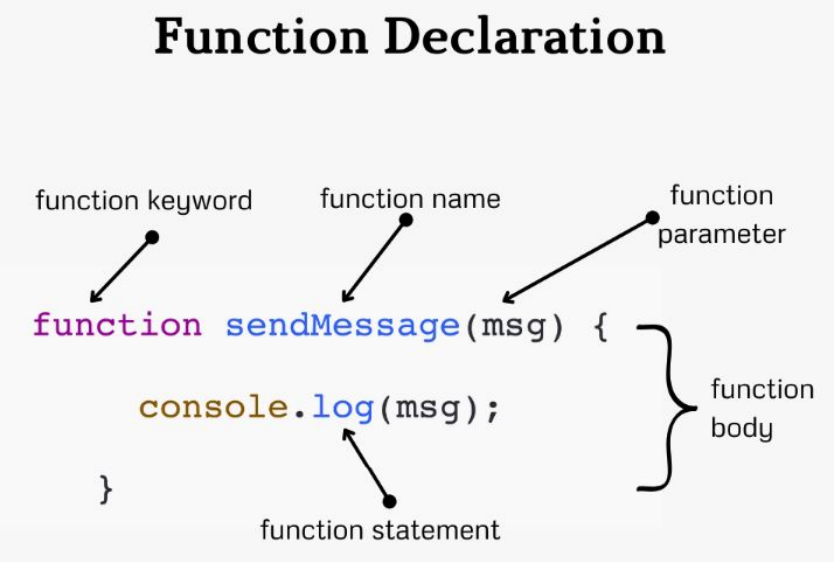
```
const hour = new Date().getHours();

if (hour < 12) {
  console.log("Good morning!");
} else if (hour < 18) {
  console.log("Good afternoon!");
} else {
  console.log("Good evening!");
}
```

What is function?

Fonksiyon, belirli bir görevi gerçekleştiren ve gerektiğinde tekrar kullanılabilen kod bloklarıdır.

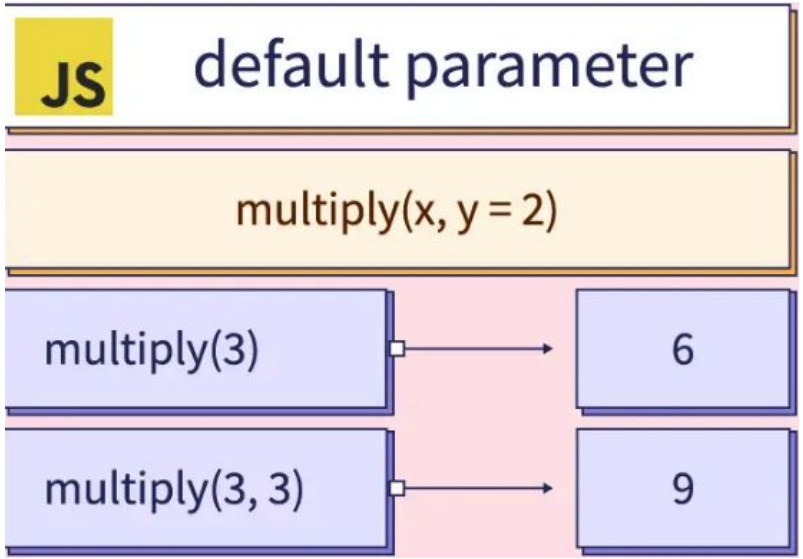
- Function Declaration
- Anonymous Function
- Arrow Function
- Immediately Invoked Function Expression (IIFE)
- Callback Function



Default Parameters and Return

Default parametreler, bir fonksiyon çağrıldığında eksik argümanlar için varsayılan değerler belirlemeye yarar. return ifadesi ise fonksiyonun çağrıldığı yere bir değer döndürmesini sağlar.

- Default parametreler, eksik argümanlar için varsayılan değerler belirler.
- return ifadesi, fonksiyonun bir değer döndürmesini sağlar.
- Eğer return kullanılmazsa, fonksiyon undefined döndürür.
- Default parametreler, diğer parametrelerle birlikte esnek kullanım sunar.



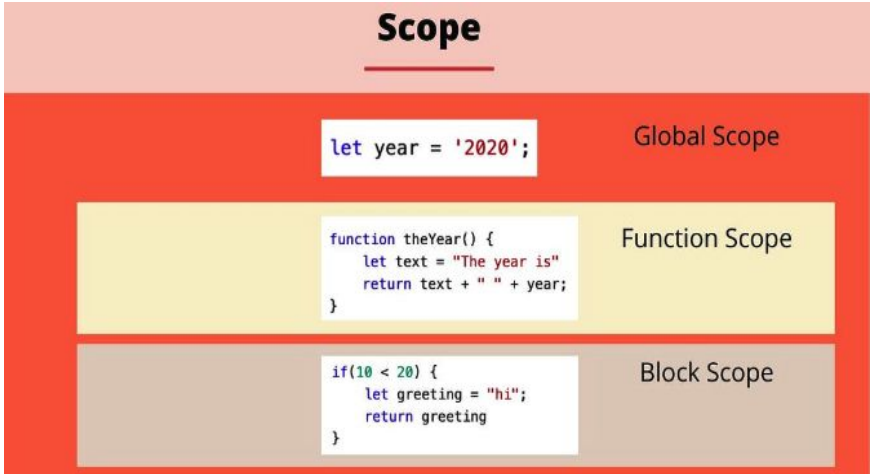
Scope (Kapsam)

Scope, bir değişkenin veya fonksiyonun hangi alanlarda erişilebilir olduğunu belirleyen kapsam kurallarıdır. JavaScript'te global, function (local) ve block scope olmak üzere üç temel scope türü bulunur.

var → Function Scope (Sadece tanımlandığı fonksiyon içinde geçerlidir, blok içinde tanımlansa bile dışarıdan erişilebilir.)

let → Block Scope (Tanımlandığı blok içinde geçerlidir, dışarıdan erişilemez.)

const → Block Scope (Tanımlandığı blok içinde geçerlidir ve değeri değiştirilemez.)



Exception Handling

JavaScript'te exception handling (hata yönetimi), kodda oluşabilecek hataları yakalamak ve uygun şekilde ele almak için try, catch ve finally bloklarını kullanma işlemidir.

try Bloğu: Hata oluşabilecek kodu içinde barındırır ve hata meydana geldiğinde kontrolü catch bloğuna aktarır.

catch Bloğu: try bloğunda oluşan hatayı yakalar ve hataya özgü işlem yapar.

finally Bloğu: Hata olsa da olmasa da her durumda çalışacak kodu içerir.

throw İfadesi: Kendi hata mesajınızı fırlatmak için kullanılır, bu sayede özel hata türleri oluşturulabilir.

```
try{
  if(age < 18) throw "You're too young"
  if (age > 95) throw "You're too old"
}
catch(error){
  console.log(error)
}
```

Homework - Longest Collatz Sequence

The following iterative sequence is defined for the set of positive integers:

$n \to n/2$ (n is even)
$n \to 3n + 1$ (n is odd)

Using the rule above and starting with 13, we generate the following sequence:

$$13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1.$$

It can be seen that this sequence (starting at 13 and finishing at 1) contains 10 terms. Although it has not been proved yet (Collatz Problem), it is thought that all starting numbers finish at 1.

Which starting number, under one million, produces the longest chain?

NOTE: Once the chain starts the terms are allowed to go above one million.