Tip Dönüşümleri

Çoğu zaman operatörler ve fonksiyonlar otomatik olarak değeri doğru tipe dönüştürürler. Buna "tip dönüştürme" denir.

Örneğin alert otomatik olarak verilen tüm değerleri karakter dizisine çevirir ve ekranda gösterir. Matematiksel operatörler ise değerleri sayılara çevirir.

Tabi bunun yanında doğrudan tipi sizin tarafınızdan değiştirilmesi gereken değişkenler vardır.

Objeler hakkında konuşulmayacak

Bu bölümde objeler hakkında bilgi verilmeyecektir. Önce ilkel tiplerin iyice öğrenilmesi lazım, sonra objelerin öğrenilmesi ve daha sonra Objelerin ilkel çevirileri bölümünde objelerin dönüştürülmesi öğrenilebilir.

1. toString

Bir değeri karakter dizisi olarak kullanmak istiyorsanız toString'i kullanabilirsiniz.

Örneğin alert(deger) değeri gösterir. Ayrıca String(deger) de kullanılabilir.

```
let value = true;

alert(typeof value); // boolean

value = String(value); // Şimdi değer karakter dizisi =  "true"

alert(typeof value); // karakter dizisi
```

Eğer false değeri karakter dizisi dönüştürme işlemine tabi tutarsanız "false", null'u tutarsanız "null" olur.

2. Number

Sayısal dönüştürme işlemleri matematiksel operasyonlarda otomatik olarak gerçekleşir.

Örneğin sayı olmayan iki değer / işlemine tutulduğunda:

```
alert( "6" / "2" ); // 3, karakterler sayılara dönüştürülür ve işlem öyle yapılır.
```

Eğer isterseniz Number(value) fonksiyonu ile değeri sayıya dönüştürebilirsiniz:

```
let str = "123";

alert(typeof str); // string

let num = Number(str); // sayı olan 123

alert(typeof num); // number
```

Bu şekilde fonksiyon ile değer dönüştürme işlemi genelde karakter dizi olarak aldığımız formlarda kullanılır. Aslında sayı kullanılmak istenmektedir. Fakat yazı kutusunun içeriğine sayı dahilinde yazılanları kontrol etmeniz gerekmektedir. Böyle bir fonksiyona sayı olmayan bir değer geldiğinde fonksiyon NaN değeri döndürür. Yani (**N**ot **a** **N**umber) sayı değil.

```
let age = Number("Bir sayı yerine herhangi bir yazı");

alert(age); // NaN, dönüştüremedi!
```

Sayısal dönüştürme kuralları:

Değer	Sonuç…
undefined	NaN
null	0
true ve false	1 veya 0
string	Önce başta ve sondaki whitespace'ler silinir. Sonra eğer kalan değerde hiçbir karakter yok ise sonuç 0. Eğer içerisinde sayısal olmayan bir değer var ise bu durumda NaN değeri alınır.

Örnekler:

```
alert( Number("   123   ") ); // 123

alert( Number("123z") );      // NaN (Hata "z" bir rakam değil)

alert( Number(true) );        // 1

alert( Number(false) );       // 0
```

Lütfen null ve undefined'ın aynı davranmadıklarını bilin. null 0 olurken undefined NaN yani sayı değildir.

Ekleme karakteri '+'

Neredeyse tüm matematiksel operasyonlar önce değerleri sayılara çevirmeye çalışır. Eğer bir taraf sayıya çevrilemediyse bu durumda karakter olarak diğeri ile birleştirme işlemi yapılır.

Bu birleştirme işlemine örnek:

```
alert( 1 + '2' ); // '12' (Sağ tarafta karakter var)

alert( '1' + 2 ); // '12' (Sol tarafta karakter var)
```

Gördüğünüz gibi sadece bir tarafın karakter olması yeterlidir. Eğer iki tarafta sayıya dönüşebiliyorsa bu durumda gerçek toplama işlemi yapılır.

3. Boolean

Boolean dönüştürme en kolay olanıdır.

Lojik operasyonlarda (durum testlerinde bu operasyonları işlenecek) otomatik olarak bu dönüştürme gerçekleşir. Bunun yanında gerekli olduğunda Boolean(value) da kullanılabilir.

Dönüştürücü kuralları:

- "boş" olan 0, veya boş karakter dizisi, null, undefined , Nan gibi değerler false olur.
- Diğer türlü değerler true olur.

Örneğin:

```
alert( Boolean(1) ); // true

alert( Boolean(0) ); // false

alert( Boolean("merhaba") ); // true

alert( Boolean("") ); // false
```

Dikkat: karakter olan "0" true'dur

PHP gibi bazı diller "0"'ı false olarak alırlar. Fakat JavaScript için boş olmayan karakter dizileri her zaman true olur.

```
alert( Boolean("0") ); // true

alert( Boolean(" ") ); // içerisinde boşluk olan karakter dizisi true olur.
```

Özet

Üç tane çok kullanılan tip dönüştürücü bulunmaktadır; karakter dizisine dönüştüren, sayıya dönüştüren ve boolean değere dönüştüren.

toString – Bir çıktı verildiğinde otomatik olarak bu fonksiyon çalışır. String(value) kullanılarak da dönüştürme işlemi yapılabilir.

Number – Matematiksel operasyonlar otomatik olarak yaparlar. Ayrıca Number(value) ile de dönüştürme işlemi yapılabilir.

Dönüştürme işlemi aşağıdaki kuralları kapsar:

Değer	Sonuç…
undefined	NaN
null	0
true / false	1 / 0
string	Önce başta ve sondaki whitespace'ler silinir. Sonra eğer kalan değerde hiçbir karakter yok ise sonuç 0. Eğer içerisinde sayısal olmayan bir değer var ise bu durumda NaN değeri alınır.

ToBoolean – Lojik operatörlerde otomatik çalışır ayrıca Boolean(value) ile de dönüştürme işlemi yapılabilir.

Kuralları şu şekildedir:

Değer	Sonuç…
0, null, undefined, NaN, ""	false
diğer her türlü değer	true

Bu kuralların çoğu akılda kalıcıdır. Genelde programcıların hata yaptıkları yer:

- undefined sayı olarak NaN'dır halbuki null sayı olarak 0 dır.
- "0" bu ve " " bu karakter dizisi boolean olarak ikisi de true olarak dönüşür.

Objeler bu bölüme konu edinmedi. Daha sonra Objelerin ilkel çevirileri konusunda özel olarak objeler hakkında bilgi verilecektir.

Görevler

Tip Dönüştürme

önem: 1

Aşağıdaki ifadelerin sonuçları nedir?

```
"" + 1 + 0
```

```
"" - 1 + 0
```

```
true + false
```

```
6 / "3"
```

```
"2" * "3"
```

```
4 + 5 + "px"
```

```
"$" + 4 + 5
```

```
"4" - 2
```

```
"4px" - 2
```

```
7 / 0
```

```
"  -9\n" + 5
```

```
"  -9\n" - 5
```

```
null + 1
```

```
undefined + 1
```

İyice düşünün, bir yere yazın ve sonra sonucunuzu doğru cevap ile karşılaştırın.