**JavaScript Değişkenler;**

* Değişken, içinde değerler saklayabileceğiniz bir programlama aracıdır. Değişkenlerin sakladığı değer kod sayfaları boyunca değişebilir, birbirine aktarılabilir. Değişkene başka bir değer atayabilirsiniz veya değişkenin sakladığı değeri program içinde kullanabilirsiniz. Değişkenler genellikle programın başlangıcında isimlendirir; böylece bu isimlerle değişkeni çağırabilir, değişkenin taşıdığı değeri kullanabilirsiniz.

**Not:** Değişkenin ismi program boyunca değişmez, ancak taşıdığı değer değişebilir.

**Değişken İsimleri**

Değişkenleri isimlendirirken dikkat etmemiz gereken bazı kurallar vardır:

* Değişken ismi harf, rakam veya "\_" (altçizgi) karakterlerinden oluşabilir.
* Değişken isminin ilk karakteri bir harf veya "\_" (alt çizgi) işareti olmalıdır. (**vol\_html23** gibi)
* JavaScript küçük harf, büyük harf ayrımı yapar. Mesela değişken isminiz DIAZ ise, bu değişkeni hep böyle büyük harflerle yazmak zorundasınız. **diaz** yazarsanız, gözatıcı bunu farklı bir değişken olarak okur.
* Değişken isimlerinin uzunluğu 255 karakter kadar olabilir.
* Değişken isimleri içinde diğer semboller (+, -, /, boşluk, vb.) yer alamaz.
* Değişken isimlerinde İngilizce'de olmayan karakterleri (ş, ç, ö, vb.) kullanmayınız. (Genellikle yeni kullanıcılar bu kuralı her zaman anımsamıyorlar.

**Değişkenlere Değer Verme**

Değişkenlere değer verirken, değişken her zaman sol tarafta yer alır. Mesela **sayi** değişkenine **15** değerini vereceksek, şöyle yazarız:

**sayi**= *15*

Değişkenler birbirinin değerini de alabilir. Her zaman sola yazılan ifade, sağdaki değeri alır. Örneğin şu iki değişkene bakalım:

**sayı1**= *5*

**sayı2**= *10*

**sayı1**= *sayı2*

Burada **sayi1** ve **sayi2** isimli iki değişken söz konusu. Önce **sayi1** isimli değişkene 5 değeri veriliyor. Sonra **sayi2** değişkenine 10 değeri veriliyor. Sonra **sayi1** değişkenine sayi2'nin değeri verilerek, sayi1'in değeri değiştiriliyor. Böylece **sayi1**= *10*, **sayi2**= *10* oluyor. Eğer **sayi2** = **sayi1** ifadesi olsaydı (**sayi2** solda yeralsaydı), **sayi1**'in değeri **sayi2**'ye verilmiş olacaktı. Bu durumda sonuç değişik olurdu: **sayi1**= *5*, **sayi2**= *5*

Yukarıdaki yazdıklarımızı bir uygulamayla gösterelim:

**<html>**

**<head>**

**<title>JavaScript Uygulaması</title>**

**</head>**

**<body>**

**<script language="JavaScript">**

**sayi1=5;**

**sayi2=10;**

**sayi1= sayi2;**

**document.write(sayi1);**

**document.write("<br>");**

**document.write(sayi2);**

**</script>**

**</body>**

**</html>**

**Değişken Tanımlama ve Değişken Türleri**

Yukarıdaki uygulamalarımızda değişkenlere değer verdik ama onları tanımlamadık, tanımlamak zorunda da değiliz. Ancak bu değişkenleri **var** kodu ile baştan tanımlamak doğru bir yaklaşımdır. (Başka program bilenlere not: Eğer baştan tanımlamazsak, değişkenler global olur. Başka program bilmeyenlere not: Önceden tanımlamakta büyük kerametler var.) Aşağıdaki üç durum da aynı şeyi ifade ediyor:

var sayi1

var sayi2

sayi1=5

sayi2=10

**veya,**

var sayi1, sayi2

sayi1=5

sayi2=10

**veya,**

var sayi1=10, sayı2=5

Bu şekillerde değişkenleri tanımlayabiliyoruz. Şimdiye kadar değişkenlere hep sayısal değerler verdik. Oysa JavaScript değişkenleri 5 çeşit değer alabiliyor:

|  |  |
| --- | --- |
| **Sayısal Değerler** | Aritmetik işlemler yapılabilen her türlü sayısal değer: tam sayı, ondalık sayı, negatif sayı. |
|  | **sayi = 32** |
| **Alfanümerik Değerler** | Aritmetik işlemler yapılmayan, metinsel özellikler taşıyan değerler. |
|  | **sev = "Onur UNURLU"** |
| **Onay Değerleri** | Sadece doğru veya yanlış değerlerini kabul eden değerler. Doğru ifadesi 1 ile, yanlış ifadesi 0 ile ifade ediliyor. (Eklenebiler boolean diyor.) |
|  | **cevap = 1** |
| **Fonksiyon Değeri** | Değişkenler tıpkı sayılar ve metinler gibi fonksiyonları da saklayabilmektedir. |
|  | **yap = selam()** |
| **Nesne Değeri** | Değişkenler HTML nesnelerini (düğme, metin, resim, vb.) değer olarak saklayabilmektedir. (Özellikle gelişmiş JavaScript örneklerimizde çok kullanacağız.) Unutmayalım ki JavaScript nesneye dayalı programdır. |
|  | **eleman = dugme** |

Değişkenler gerek yapısal gerek nesneye dayalı programlarda büyük önem taşır. Daha önce hiç programcılıkla uğraşmamışlar kişileri düşünerek anlattım bu sayfayı. Değişkenleri tanıyan arkadaşların sıkılmaması için olabildiğince sade yazdım, umarım başarmışımdır.