

Ders17: JavaScript Giriş

Konular: JavaScript Giriş, Değişken, Değişken Tanımlama

JavaScript, etkileşimli bir şekilde web sayfası oluşturabilmek için kullanılan bir programlama dilidir. Web uygulamalarını daha dinamik hale getirebilmek için tarayıcı bazlı bir dildir. Tarayıcılar JavaScript ile kullanıcı etkileşimine yanıt verebilir ve içerik düzenini değiştirebilir.

JavaScript günümüzde hem istemci tarafında hem de sunucu tarafında kullanılmaktadır.

HTML sayfasında <body> etiketleri içerisine <script> etiketi kullanarak oluşturduğumuz .js uzantılı bir JavaScript dosyasını ekleyebiliriz. Script içerisinde src özelliğini kullanarak JavaScript dosyasının uzantısını vererek eklemesini yapabiliriz. Böylece HTML dosyası ile JavaScript dosyasının bağlantısını gerçekleştirmiş oluyoruz.

Farklı bir JavaScript dosyası yerine HTML dosyasında da JavaScript kodları yazmak mümkündür. Bu işlemi de yine <script> etiketi içerisine yazarak gerçekleştirmek mümkün hale geliyor. Ancak bu çok da tercih edilen bir durum değildir. Çünkü okunabilirlik açısından ve dosyalama açısından farklı bir dosyada JavaScript kodlarını bulundurmak daha kullanışlıdır. Çok fazla JavaScript kodu yazıldığı zaman da farklı bir dosyada bulunması daha faydalı olacaktır.

1. Console Kullanımı

Oluşturulan web sayfası içerisinde inspect yani incele kısmında Console adında bir kısım bulunmaktadır. JavaScript dosyası içerisinde console.log() yazarak parantez içerisinde consolda görünmesini istediğimiz şeyleri yazdırmamız mümkündür.

```
console.log("Merhaba JavaScript");
```

console kısmına bir hata mesajı yazdırmak da isteyebiliriz. Bunun için de console.error() kullanılmaktadır. Yine parantezler arasına yazılacak olan bilgiler konsolda hata mesajı olduğunu belirtir.

```
console.error("Yüklenirken Hata Oluşturdu");
```

console kısmında bir uyarı mesajı da vermek mümkündür. Bunun için de console.warn() kullanılır.

```
console.warn("Bu bir uyarı mesajıdır.");
```

console'da bulunan tüm mesajları temizlemek için de console.clear() kullanılır.

```
console.clear();
```

Console kullanımı önemlidir. Çünkü yapmış olduğumuz işlemleri test edebiliyoruz.

2. write Kullanımı

JavaScript dosyası üzerinden HTML dosyasına herhangi bir şey yazdırmak istediğimiz zaman document.write() kullanılmaktadır.

```
document.write("JavaScript'den gelen deneme yazısı");
```

```
document.write("<br>2.Deneme");
```

write içerişine HTML etiketleri de eklenebilir. Ancak HTML tagi "" (çift tırnak) içerisinde bulunmalıdır.

3. Değişkenler

Değişken oluşturmak için var komutu bulunmaktadır. Değişken değeri verilmediği zaman oluşturduğumuz değişken tanımlı değildir. Console kısmında da undefined şeklinde görünmektedir.

```
var password = "ahmetkanbaz"
```

Değişkenler türlerine göre console kısmında farklı renklerle gösterilebilirler.

4. Değişken Tanımlama Kuralları

a- Değişken oluştururken değişken ismi rakamla başlamamalıdır. Ancak ilk harfinden sonra rakam kullanılabilir.

```
var 1yas; //Bu şekilde kullanım yanlıştır.
```

```
var yas1 = 24;
```

b- Kullanılan komutlar değişken ismi olarak verilemez. if, for, while gibi. Çünkü bu komutların kendi görevi var.

```
var if; //Yanlıştır.
```

c- Değişken ismi en az iki kelimeden oluşuyorsa farklı kelimeler arasında boşluk bırakılamaz.

```
var ad soyad; // Yanlıştır.
```

d- Değişken isimlendirirken Türkçe karakter kullanmamalıyız.

```
var diğErAd; //Yanlıştır.
```

Sabit Değişken

Bir değişkeni sabit değişken olarak tanımlarsak tanımlamış olduğumuz değişkenin içeriğini, değerini değiştirmemize izin vermeyecektir. Değişkeni sabit olarak tanımlayabilmek için de değişken isminin başına "const" eklememiz yeterli olacaktır.



```
const emailAdress = "ahmetkanbaz@gmail.com";
console.log(emailAdress);

emailAdress = "1234567@gmail.com";
console.log(emailAdress);
```

Yukarıda bulunan kod ilk başta sabit olarak tanımlanan değişkenin değerini yazdırır. Ancak sabit değişkenin değeri değiştirildiği kısım için ise hata verecektir.