**T.C.**

**BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ**

**SÖĞÜT MESLEK YÜKSEKOKULU**

**Programlamanın Temelleri Dersi**

**Araştırma Ödevi**

**Konu: JavaScript**

Ad: xxxxxx xxxxxxxxx

TC: xxxxxxxxxxxxx

Öğretim: I / II

2017

# JavaScript Nedir? Ne İşe Yarar?

JavaScript istemci ve sunucu tarafı internet uygulamaları geliştirmekte kullanılan nesne tabanlı bir script dilidir. Bir HTML içine eklenen JavaScript kodları herhangi bir derleme olmaksızın doğrudan tarayıcılar ( Microsoft Internet Explorer, Netscape Navigator gibi ) tarafından yorumlanarak gerekli işlemler yapılır.

Bir tarayıcı için bir istemcide, bir HTML sayfası içine eklenen JavaScript ifadeleri tanınır ve kullanıcı tarafından neden olunan fare tıklaması, form girişi ve sayfa değiştirmek gibi olaylara cevap verilir.

Örneğin, kullanıcının, bir form üzerinde text alanına girilen bir bilginin geçerli olup olmadığını sınaması için geçerlilik testi yapan bir fonksiyon yazabilirsiniz. Ağ üzerinde herhangi bir iletim olmaksızın, JavaScript kodları eklenmiş bir HTML sayfası üzerinde girilen bilginin geçerli olup olmadığı kontrol edilir ve ona göre kullanıcıya bir uyarı mesajı verilebilir veya siz kullanıcıya bir sayfayı açtığında veya kapattığında JavaScripti kullanarak ses dosyası çalma, bir Java appleti yürütme gibi işlemleri yapabilirsiniz.[1]

JavaScript bize etkileşimli ve dinamik web sayfaları hazırlama imkanı vermektedir. Sözdizimi Java’ya benzese de Java ve JavaScript birbirinden farklı teknolojilerdir. Ağırlıklı olarak Web Tasarım’da kullanılsa da başka birkaç kullanım alanı daha mevcuttur. Örnek olarak JavaScript ilk zamanlarda sadece istemci taraflı kullanım için tasarlanmış olsa da şimdilerde NodeJS ile birlikte sunucu taraflı kullanımına da başlanmıştır. Buna ek olarak Unity oyun motorunda da JavaScript ile bazı kodlamalar yapılabiliyor. JavaScript yardımı ile HTML kaynak kodları değiştirilebilir, bu sayede dinamik web sayfaları hazırlamamız mümkün oluyor. JavaScript kullanılan web sayfalarını düzgün görüntüleyebilmek için tarayıcınızın JavaScript desteğine sahip olması gerekmektedir. Günümüzde güncel tarayıcıların hepsi JavaScript desteklemektedir. [2]

## JavaScript Kodları Nereye ve Hangi Programlara Yazılır?

Javascript kodları, Notepad+++, Dreamweaver, Codeblocks, Visual Studio da .js dosyası formatında yazılır. Javascript dosyaları, web sitesine iki şekilde eklenir. Birincisi satıriçi (inline), ikincisi ayrı bir javascript dosyası hazırlayıp harici olarak eklemek.

## JavaScript <head> Etiketleri Arasına Yazmak

Bir javascript kodları HTML sayfasının <head> bölümüne yerleştirilir. HTML sayfasında javascript eklemek için <script> etiketi kullanılır.

<script> ve </script> Etiketleri JavaScript kodların başlangıç ve bitişin nerede olduğunu belirtir.

<html>

<head>

<title></title>

<script>

 //java kodları buraya yazılır.

</script>

</head>

<body>

</body>

<html>

## Ayrı Bir Javascript Dosyası Hazırlayıp Harici Olarak Yazmak

Ayrı bir javascript dosyası olarak hazırlayıp  .js uzantısı ile kaydedersek ve bunu html dokümana eklersek buda harici bir javascript dosyası ekleme kısmına girer.

ÖRNEK: deneme.js olarak kaydedin. [3]

<html>

<head>

<title></title>

<script type=”text/javascript” src=”deneme.js”></script>

</script>

</head>

<body>

</body>

<html>

## JavaScript Veri Tipleri

**Boolean:** Mantıksal varlığı ifade eder. İki tip değer içerebilir true ve false

**Null:** Tam olarak null değerini taşıyan veri türüdür.

**Undefined:**  Değişken tanımlanmış ama değer atanmamış ise değişken içindeki değer undefined dır.

**Number:** EcmaScript standartlarına göre 64bit işaretli tek sayı tipi vardır. Tam yada ondalıklı sayılar number tipi ile ifade edilmektedir. Ek olarak +Infinity, -Infinity, and NaN (not-a-number) olmak üzere 3 tane özel sembolik değer vardır.

**String:** JavaScript dizge üzerinde metinsel verileri temsil etmek için kullandığı veri tipidir. Dizgedeki her öğe 16 bitlik işaretsiz integer bir referansı vardır. Dizgedeki ilk elemanın index numarası 0dır. Dizgedeki diğer değerler için index numarası 1 artarak devam eder.

**Özetle:** Metinsel bir ifade karakterlerden oluşan bir dizi olarak nitelendirilir.

**Symbol:** EcmaScript 6 standartı ile tanımlanmış bir tipdir. Nesne değeri için benzersiz ve sabit bir değeri ifade eder.

**Object:** Bellekteki belirli bir nesnenin referansını(yolunu) tutmak için kullanılan tiptir. Dizi, Fonksiyon ve Nesneler bu tip ile ifade edilir.

## JavaScript İle Sabit Tanımlama

Sabit (constant) diye adlandırdığımız değeri değiştirilemeyen ifadeler const anahtar sözcüğüyle birlikte bildirilir. Bir sabit belirtecinin yazımı, değişken belirtecinin yazımıyla aynı olduğu için değişken yazım kuralları sabitler için de geçerlidir.

<html><head><title></title>

<script>

Const alanKodu: “322”;

</script>

</head><body></body></html>

## JavaScript Değişken Tanımlama ve Değişken Tipleri

JavaScriptte değişken tanımlamak için var deyimini kullanırız. Global değişken, ilgili fonksiyon içinde tanımlanan değişken gibi. Yani bir fonksiyon içinde tanımlanmış bir değişkeni yalnızca o fonksiyon içinde kullanabilirsiniz. Scirpt öğesinde direkt olarak tanımlanan değişkenler ise global değişkenlerdir ve bunlar tüm script boyunca alt fonksiyonlarda dahil olmak üzere kullanılabilirler.

Global değişken tanımlama için aşağıdaki örneği inceleyebilirsiniz:

<html>

<head>

<title></title>

<script>

var globalDegisken = “Galatasaray”;

function fonksiyonum (ad)

var lokaldegisken = “GÜRSOY”;

return /”Merhaba” + ad + “ ” + lokalDegisken + “nasılsın?”);

}

document.writeln(fonksiyonum(“İsmail”));

</script>Merhaba İsmail GÜRSOY nasılsın?

<script type= “text/javascript”>

document.writeln(“Sampiyon” + globalDegisken);

</script>

</head>

<body>

</body>

<html>

**String Tipler :** String değerleri tanımlarken çift tırnaklar arasında ya da tek tırnaklar arasında tanımlayabiliriz. Aşağıda örneğini görebilirsiniz :

<script type= “text/javascript”>

var birinciDegisken= ”İsmail Gürsoy”;

var ikinciDegisken= ‘JavaScript’

document.writeln(birinciDegisken);

document.writeln(ikinciDegisken);

<script>

Booelan Değişkenler: İki tip boolean değişken mevcuttur, True ve False. Boolean değişkenler özellikle if yapısı gibi karar yapılarında kullanılırlar. Aşağıdaki gibi tanımlanabilirler;

var birinciDegisken = true;

var ikinciDegisken = false;

Number Değişkenler: Number değişkenler sayı tipindeki değişkenlerdir. JavaScriptte tam sayı, ondalıklı sayı tipindeki değişkenler tanımlanabilir.

Aşağıdaki gibi sayıları tanımlayabiliyoruz.

var sayi = 7;

var pi = 3.14;

var hexadecimalSayi = 0xFFFF;

## JavaScript İle Operatör Tanımlama

### Atama Operatörü

JavaScript’in temel operatörü olan atama operatörü ‘=’ işaretiyle ifade edilir, işaretin sağındaki değeri, solundaki değişkene atar.

‘var sayi=34; var sayi2= sayi’

### Aritmetik Operatörler

‘+’ : Sayı değişkenleri toplamaya, string değişkenleri birbirine eklemeye yarar.

‘-’: Sayı değişkenlerde çıkartma işlemini yapar.

‘\*’ Sayı değişkenlerde çarpma işlemini yapar.

‘/’ Bölme işlemini yapar.

‘%’: Modülüs işlemi. ‘10%3’ ifedesinin matematikteki karşılığı ’10 mod 3’ ifadesidir ve sonuç 1 çıkar.

<script type=”text/javascript”>

var toplama= 11+2

var çıkarma= 10-4

var çarpma=(7,32)\*4

var bölme=(32,4)/(1,22)

var modül= 23%3

var hepsi= toplama+çıkarma+çarpma+bölme+çarpma+modül

document.write(toplama”<br>”, çıkarma”<br>”, carpma”<br>”, bölme”<br>”, modül”<br>”)

document.write(hepsi)

</script>


### Karşılaştırma Operatörleri

Karşılaştırma operatörleri, temsil ettikleri karşılaştırmanın sonucu doğruysa true, yanlışsa false değerini verirler.

'==' : Eşit operatörü, eğer iki tarafın değeri eşitse true sonucunu verir.

'!=' : Eşit değil operatörü, eğer iki tarafın değeri eşit değilse true sonucunu verir.

'< ' : Küçüktür operatörü, sol taraf küçükse true sonucunu verir.

'> ' : Büyüktür operatörü, sol taraf büyükse true sonucunu verir.

'<=' : Küçük eşittir operatörü, sol taraf küçükse veya iki taraf eşitse true sonucunu verir.

'>=' : Büyük eşittir operatörü, sol taraf büyükse veya iki taraf eşitse true sonucunu verir.

### Şart Operatörü

<script type= “text/javascript”>

var a =10, b=4;

document.writeln (a==b))

document.writeln (a!=b))

document.writeln(a<b))

document.writeln(a>b))

document.writeln(a<=b))

document.writeln(a>=b))

<script>

Bu operatörü bir örnek üzerinde öğrenelim: (a==b)? "Iki degisken esittir!":"Iki degisken esit degildir!" Bu operatörün kullanımında parantez içerisine bir şart yazılır, ki burada o şart x ve y değişkenlerinin eşit olmasıdır, eğer bu şart doğruysa soru işaretinden hemen sonraki işlem, aksi takdirde ':' işaretinden sonraki işlem yapılır.

### Mantıksal operatörler

Eğer karşılaştırmayı birden fazla durumu dikkate alarak yapacaksak mantıksal operatörleri kullanırız.

'&&' : 'And(ve)' operatörü: Koşulların hepsi doğru ise true değerini verir.

'|| ' : 'Or(veya)' operatörü: Koşulların en az birinin doğru olması halinde true değerini verir.

'! ' : 'Not' operatörü: Koşulların hepsinin yanlış olması durumunda true değerini verir.

## JavaScript İle Koşul Tanımlama

### IF-ELSE KOŞULU

Eğer bir şart doğru ise if kısmı çalışır. Eğer yanlış ise else kısmı çalışır.

<script>

Varsayi1=25;

 İf (sayi1<50)

{

document.write(“Bu sayı 50’den küçüktür.”);

}

Else

{

document.write(“Bu sayı 50’den büyüktür.”);

}

</script>

### ELSE IF KOŞULU

<script>

var sayi=100;

if(sayi1<50)

{

document.write(“Bu sayı 50’den küçüktür.”);

}

else if (sayi1==50)

{

document.write(“Bu sayı 50’ye eşittir.”);

}

Else if (sayi1>50)

{

document.write(“Bu sayı 50’den büyüktür.”);

}

</script>


### SWITCH KOŞULU

Bu koşul else if koşuluna benzemektedir. Ancak uzun uzun yazmak yerine switch kullanmak daha kısa olacaktır.

<script>

var sayi1=prombt (“Haftanın kaçıncı gününü seçersin: ”);

switch(sayi1)

}

case “1”;

document.write(Seçtiğiniz gün Pazartesi.”);

break;

case “2”;

document.write(Seçtiğiniz gün Salı.”);

break;

case “3”;

document.write(Seçtiğiniz gün Çarşamba.”);

break;

case “4”;

document.write(Seçtiğiniz gün Perşembe.”);

break;

case “5”;

document.write(Seçtiğiniz gün Cuma.”);

break;

case “6”;

document.write(Seçtiğiniz gün Cumartesi.”);

break;

case “7”;

document.write(Seçtiğiniz gün Pazar.”);

break;

default;

alert (Haftanın 7 günü olduğu için 1 den 7 ye kadar değer girmelisiniz.”);

</script>

Bu örnekte kullanıcıdan haftanın gününü sayı karşılığında girmesi isteniyor. Daha sonra girdiği sayının hangi güne denk geldiğini sistem kullanıcıya yansıtıyor. Ve kullanım şekli yukarıdaki gibidir.

Burada kullanıcı 5 sayısını girdiği için switch koşulunda case “5” kısmı çalışacaktır ve bu yüzden ekrana aşağıdaki gibi bir pencere çıkacaktır. Ancak aşağıdaki örnekte kullanıcı anlamsızca 8 sayısını girer halbuki haftanın 7 günü olduğunu bilmesi gerekirdi fakat switch koşulunda uygun bir case olmadığı için varsayılan yani default kısmı çalışacaktır.

Ve çıktı aşağıdaki gibi olacaktır.


## JavaScript İle Döngüler

Döngüler programlamanın olmazsa olmazlarından biridir ve programları kısaltmamızda oldukça faydalıdır. Örneğin; 1’den 1000’e kadar olan sayıları tek tek ekrana yazdırmak oldukça zahmetli ve uzun olacaktır. Fakat işte tam burada döngüler devreye giriyor.

### FOR DÖNGÜSÜ

var sayi1=prompt (“Bir sayı giriniz: ”);

 for (var i=1; i<=sayi1;i++)

{

document.write(i+”<br>”);

}

for(var i=1; i<=sayi1;i++)’ bu kısmın aslında anlamı döngü 1’den başlayıp kullanıcının girdiği sayıya kadar döngüyü birer birer artırarak döngüyü çalıştır demektir. Ve döngü her döndüğünde ekrana i’yi yazdırırız.


### WHILE DÖNGÜSÜ

While döngüsü, for ile aynı işlemleri yapmaktadır. Sadece yazılış şekilleri farklıdır. Aşağıdaki örnek while’ın kullanımını açıkça belirtmektedir.

<script>

var sayi1=promt (“Bir sayı giriniz: ”);

var i=1;

while(i:sayı1;)

{

document.write(i+”<br>”);

İ++

}

</script>

Burada döngünün kaçtan başlayacağını değişkeni tanımlarken ilk başta belirtiyoruz ve döngünün nereye kadar döneceğini while’ın yanında belirtiyoruz. For’dan farklı olarak döngünün değerini artırma işlemini bütün işlemler yapıldıktan sonra yapıyoruz.

### DO WHILE DÖNGÜSÜ

<script>

Var sayi1=prompt (“Bir sayıgiriniz”);

Var i=5;

Do

{

Document.write(i+”<br>”);

İ++

}while(i<=sayi1)

</script>

Do while, diğer döngülerle mantık olarak aynı olsa bile bazı farklılıkları vardır. Mesela; while’dan farkı koşula döngünün sonunda bakmasıdır. Yani koşulum sağlamasa bile bu döngüyü en az bir defa çalıştırır.

Örneğin; burada kullanıcı 4 sayısını girmektedir ancak do while döngüsü çalışırken döngü 5’ten başlıyor ve döngü bir defa çalışıp ekrana 5 yazdırıyor. Bundan sonra koşula bakıyor ve koşul sağlamadığı için döngüyü çalıştırmaya devam ettirmiyor.

Bu örnek ise döngünün koşulu sağladığı bir örnektir.

 [3]

ÖRNEK ÇALIŞMALAR:

1)Digital Saat Yapma (Bu örnekte bilgisayarın şuanki saatini ekrana yazdıracağız)

<html>

<head>

<meta charset="utf-8">

<title>Dijital Saat</title>

</head>

<body>

<h1>Bilgisayarın Şuanki Saati: <span id="zemin"> </span></h1>

<script>

var saatZemin = document.getElementById("zemin");

function format(gelen) {

var StrYap = gelen.toString();

if (StrYap.length === 1) {

return "0" + StrYap;

} else {

return StrYap;

}

}

function saatGuncelle() {

var dd = new Date();

var hh = dd.getHours();

var mm = dd.getMinutes();

var ss = dd.getSeconds();

saatZemin.textContent = format(hh) + ":" + format(mm) + ":" + format(ss);

}

setInterval(saatGuncelle, 1000);

</script>

</body>

</html>

2)Ekrana büyüyen font ile Türkiye yazdırma

<html>

<head>

<meta charset="utf-8">

<title>ekrana büyüyen font ile Türkiye yazdırma</title>

</head>

<style>

</style>

<body>

<script type="text/javascript">

for (var i = 0; i < 10; i++) {

document.write("<span style='font-size:" + i + 5 + "px;'> TÜRKİYE </span> <br/> ");}

</script>

</body>

</html>

3) 1 ile 10 arasındaki ÇİFT sayıları ekrana yazdırma

<html>

<head>

<title></title>

<script type="text/javascript">

for (var i = 0; i <= 10; i++) {

if(i%2==0) document.write(i+ "<br/>");

}

</script>

</head>

<body>

</body>

</html>

4)1-100 arası sayılardan 3’e ve 5’e tam bölünen sayıları listeleyen program

<html>

<head>

 <title></title>

 <script type="text/javascript">

 var toplam = 0;

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

 if (i % 3 == 0 && i % 5 == 0)

 {

 document.write(i + "<br/>");

 }

 }

 </script>

</head>

<body>

</body>

</html>

5)Yıldız ile ekrana üçgen şekli yazdırma

<html>

<head>

<title></title>

<script type="text/javascript">

for (var i = 0; i < 10; i++) {

for (var x = 0; x < i; x++) {

document.write("\*");

}

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

}

</script>

</head>

<body>

</body>

</html>

6)10’lar çarpım tablosunu ekranda bir tablo içinde yazdırma(bu örnekte table parametresi ile tablo yapacağız.)

<html>

<head>

 <title></title> <script type="text/javascript">

 document.write("<table border='3'><tr>");

 for (var x = 1; x <= 10; x++) {

 document.write("<td>");

 for (var i = 1; i <= 10; i++) {

 document.write(x+"x" + i + "=" + x \* i + "<br/>");

 }

 if(x==5) { document.write("</tr><tr>"); }

 document.write("</td>"); }

 document.write("</tr><table>");

 </script>

</head> <body> </body> </html>


# Kaynaklar

https://www.ahmetiscan.web.tr/javascript-nedir-nerelerde-kullanilir-nasil-ogrenilir/

http://www.webcebir.com/140-javascript-kodlari-nereye-yazilir-ve-aciklama-satiri-dersi.html

http://www.ismailgursoy.com.tr/javascript-degisken-tanimlama-ve-degisken-tipleri/