

Javascript Kodları

Tıklanma Olayına Fonksiyon Bağlamak

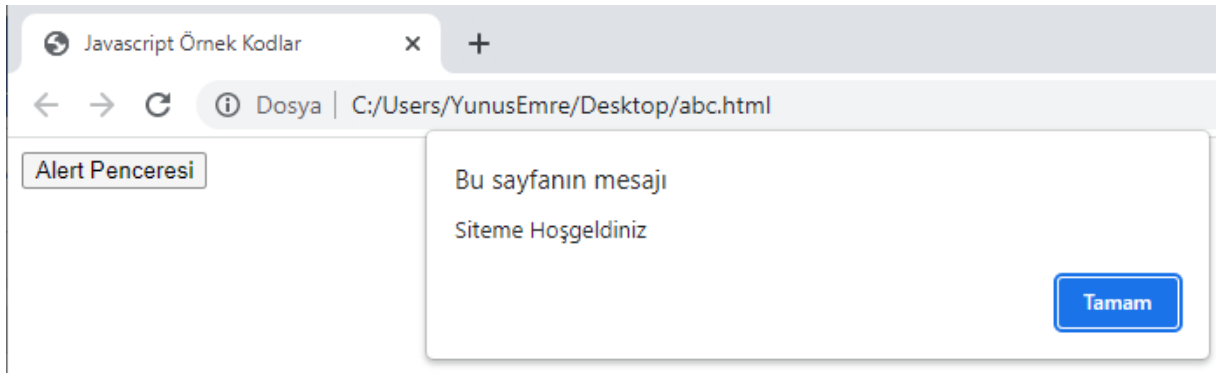
Form elemanlarına tıklandığında fonksiyon çalıştırılabilir. Aşağıdaki örnekte Butona tıklandığında alert penceresi açan bir fonksiyon tanımlandı:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Javascript Örnek Kodlar</title>
  </head>

  <body>
    <input type="button" name="button" id="selam" value="Alert Penceresi">

    <script>

      var btnSelam=document.getElementById("selam");
      btnSelam.onclick=function(){
        window.alert("Merhaba");
      }
    </script>
  </body>
</html>
```



Try-Catch Bloğu Tanımlamak

Try-catch bloğuyla hata kontrolü yaparak kullanıcıyı yönlendirebiliriz. Aşağıdaki kodda girisKodu isimli text elemanından ayrılma anında kullanıcı girişinin kontrol edilmesini sağlıyor:

onBlur : Seçili (aktif) olan elemanın terkedilmesi (odağın çıkması - seçilmiş olma özelliğini kaybetmesi) olayıdır. Metin kutusuna tıklanıldığında onFocus olayı gerçekleşir (kutuya odaklanılmış olur ve kutu aktif hale gelir).

isNaN fonksiyonu : kullanıcıdan veya bir ifade içerisinde gelen değerin sayısal olup olmadığını kontrol eder. Değer sayısal bir ifade ise false, sayısal ifade değilse (NaN) true değerini döndürür.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Javascript Örnek Kodlar</title>
  </head>

  <body>
    <input type="text" id="kod">
    <div id="bilgi"></div>
    <script>
      var girisKodu =document.getElementById("kod");
      var aciklama =document.getElementById("bilgi");

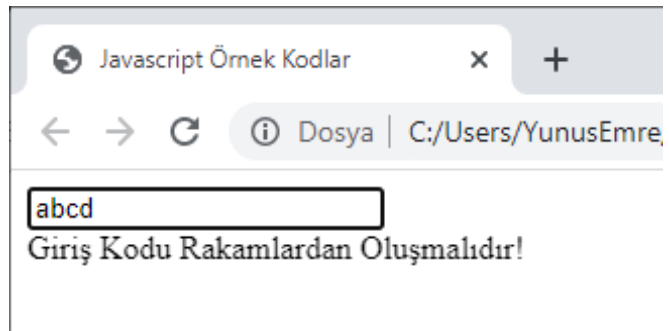
      girisKodu.onkeypress=function(){
        aciklama.textContent="";
      }

      girisKodu.onblur=function(){

        try{
          if(isNaN(girisKodu.value)) throw "Giriş Kodu Rakamlardan
Oluşmalıdır!";
          if(girisKodu.value.length<5) throw "5 karakterden kısa giriş
kodu olmaz!";
          if(girisKodu.value.length>5) throw "En fazla 5 haneli
olmalı!";
        }
        catch(e)
        {
          aciklama.textContent=e;
          girisKodu.focus();
        }

      }
    </script>
  </body>
```

```
</html>
```



Fare Koordinatlarını Almak ve Nesnelere Koordinat Atamak

Onmousemove olayı ile bir nesneye koordinat olarak anlık fare konumunu atayarak sürekli yerdeğiştirmesi sağlanıyor. Atanan konum bilgisi konsolda da listeleniyor.

```
<!doctype html>
<html>

<head>
  <meta charset="utf-8">
  <title>Farenin Peşinde</title>

  <style>
    body{
      background: #ca1e1e;
    }
    #cisim{
      background: #e2df30;
      width: 15px;
      height: 15px;
      border-radius: 50%;
      position: absolute;
    }
  </style>
</head>

<body>
  <div id="cisim"> </div>

  <script>
    var cisim=document.getElementById("cisim");
    cisim.style.position="absolute";
    window.onmousemove=function(konum){
      cisim.style.top=konum.clientY+20+"px";
      cisim.style.left=konum.clientX+10+"px";
      console.log(konum.offsetX);
    }
  </script>
</body>

</html>
```

Dizideki En küçük ve En büyük değeri bulmak

Aşağıdaki örnek dizi elemanları içerisinde en küçük ve en büyük değerleri bulmak için for döngüsünü kullanıyor.

innerHTML fonksiyonu ile bir elemanın içeriğini elde edebiliriz veya html elemanın içeriğini değiştirebiliriz.

```
<!doctype html>
<html>

<head>
  <meta charset="utf-8">
  <title>Farenin Peşinde</title>
</head>

<body>
  <h3>En Büyük / En Küçük Elemanı Bulmak</h3>
  <div id="bilgi"></div>

  <script>
    let arr = [1.45,8.5,4.56,0.927,5.4,78.9];
    var enb = arr[0]
    var enk = arr[0];

    for(var i=0; i<arr.length ; i++){
      if(enb < arr[i])
        enb = arr[i];

      if(enk > arr[i])
        enk = arr[i];
    }

    document.getElementById("bilgi").innerHTML = "En küçük eleman: " + enk
+ ", Ve en büyük eleman: " + enb;
  </script>
</body>

</html>
```

