



Düzenli İfadeler & JSON & Form Doğrulama

## Konu başlıkları

- Düzenli İfadeler Regex
- JavaScript Nesneler
- JavaScript Fonksiyonlar
- JSON
- Form Doğrulama

2



## Konuya Özgü Kavramlar

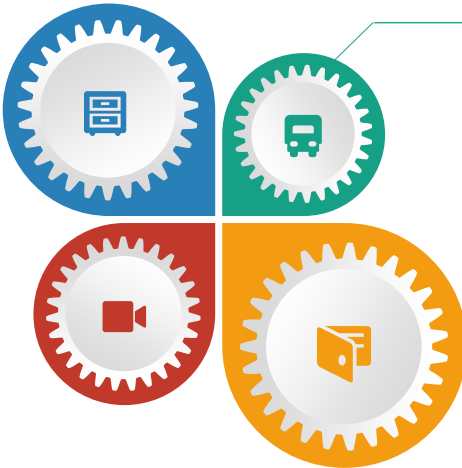
*JSON*

*Search, replace, case insensitive*

*Validation*



## Düzenli İfadeler



- Düzenli ifade belli bir biçime uyan karakterler dizesidir.
- Bir veriyi metin dokümanı içerisinde aratırken istediğimiz veri desenini tanımlarız.
- Düzenli ifade bir karakteri ya da daha karmaşık bir deseni (metin dizesini) temsil edebilir.
- Düzenli ifadeler metin veri arması veya değiştirilmesi konularında işlemler gerçekleştirebilir.

/desen/kurallar;

var desen= /w3schools/i;

## search Örnek

5

```
var str = "Visit W3Schools";  
var n = str.search(/w3schools/i); // 6
```

```
var str = "Visit W3Schools!";  
var n = str.search("W3Schools"); // 6
```

```
var str = "Visit Microsoft!";  
var res = str.replace(/microsoft/i, "W3Schools"); // Visit  
W3Schools
```

## Replace Örnek

6

```
var str = "Visit Microsoft!";  
var res = str.replace(/microsoft/i, "W3Schools"); // Visit W3Schools
```

```
var str = "Visit Microsoft!";  
var res = str.replace("Microsoft", "W3Schools"); // Visit W3Schools
```

- Stringler ve Düzenli ifadeler aynı fonksiyonlarla kullanılabilirler.
- Düzenli ifadeler daha güçlü aranacak ifade tanımlaması sağlarlar.

## Regex Kurallar Modifiers

7

Kural Modifier	Tanım
i	Büyük Küçük harf Duyarsız
g	Bütün eşleşmeleri bul (İlkini bulunca Durma!)
m	Çok satırlı eşlemeler yap

## Regex Desenleri

8

İfade	Tanım
[abc]	Sadece Parantez içinde geçen karakterleri bul
[0-9]	Sadece parantez içinde verilen aralıktaki sayılardan geçenleri bul
(x y)	Parantez içindeki iki ihtimalden birini bul

## Regex Meta Karakterler

9

Metacharacter	Tanım
\d	Sayıları Bul
\s	Boşluk karakterini bul
\b	Bir kelimenin en başında yada en sonunda eşleşen ifadeyi bul
\uxxxx	Heksadesimal olarak tanımlanan Unicode karakteri bul

## İçinde Ne kadar Gececeği

10

Quantifier	Description
n+	İçinde en azından 1 tane n geçenler
n*	İçinde n geçmeyen veya bir ya da birden fazla n geçenler
n?	İçinde hiç yada bir kez n geçenleri

## RegEx Nesnesi

- JavaScript de RegEx nesnesi düzenli ifadeler için ön tanımlı fonksiyonları içerir.
- `test()` metodu RegEx nesnesine ait bir metottur.

```
var desen= /e/;  
desen.test("Önce en iyi eğitim!"); // içinde e geçtiği için true
```

- Şöyle de yazılabilir  
`/e/.test("Önce en iyi eğitim!");`

## Exec() Kullanmak

`Exec()` metodu da bir regEx nesnesi metodudur.  
Bir deseni arar ve bulursa geri döndürür.  
Bulamazsa null döndürür.

Örn : `/e/.exec("Önce en iyi eğitim!"); // e`

## JSON Örnek

- Üç çalışan bilgisini JSON ile tanımlayalım

```
{  
  "calisanlar": [  
    {"ad": "John", "soyad": "Doe"},  
    {"ad": "Anna", "soyad": "Smith"},  
    {"ad": "Peter", "soyad": "Jones"}  
  ]  
}
```

- JSON JavaScript'teki nesne yazımı ile aynı olduğu için kolayca JS nesneleri olarak ifade edilebilirler.

## JSON Yazım Kuralları

- Veri, isim değer çiftleri olarak tutulur
- Veri, virgüller ile ayrılır
- Süslü parantezler nesneleri tutar
- Köşeli parantezler dizileri tutar

## JSON Verisi

15

JSON verisi aynen nesne gibi kodlanır

```
"ad":"UZEM"
```

JSON objeleri süslü parantezler içerisine yazılır

```
{"ad":"OMÜ", "program":"UZEM"}
```

JSON dizileri Köşeli Parantezler İçerisine Yazılır

```
"calisanlar":[  
  {"ad":"John", "soyad":"Doe"},  
  {"ad":"Anna", "soyad":"Smith"},  
  {"ad":"Peter", "soyad":"Jones"}  
]
```

Dizi elemanları nesnelerden oluşabilmektedir.

## JSON Metnini JS Nesnesine Dönüştürmek

16

JSON genel itibariyle web sunusundan alınan verilerin web sayfasında görüntülenmesi için kullanılır.

Basitçe bir metin dosyası için örneklersek

```
var text = '{"calisanlar":[' +  
  '{"ad":"John", "soyad":"Doe"}, ' +  
  '{"ad":"Anna", "soyad":"Smith"}, ' +  
  '{"ad":"Peter", "soyad":"Jones"} ]}';
```

```
var obj = JSON.parse(text); // Nesne olarak ayıklar
```



## JSON Örnek

17

```
<p id="demo"></p>
```

```
<script>  
document.getElementById("demo").innerHTML =  
obj.calisanlar[1].ad + " " + obj.calisanlar[1].soyad;  
</script>
```

Çıktısı : Anna Smith

## Form Doğrulama

18

JS ile form elemanları kontrol edilerek istediğimiz formata uygun değer girilmesi sağlanabilir.

Aksi takdirde form submit edilip gönderilmez. Alert verilebilir.

```
function formDogrula() {  
  var x = document.forms["form1"]["ad"].value;  
  if (x == null || x == "") {  
    alert("Ad alanı boş geçilemez!");  
    return false;  
  }  
}
```

## Form Doğrulama

```
<form name="form1" action="demo_form.asp" onsubmit="return formDogrula()" method="post">  
Name: <input type="text" name="ad">  
<input type="submit" value="Submit">  
</form>
```

## Form Doğrulama Numara mı Değil mi

```
<input id="numb">  
<button type="button" onclick="fonksiyon()"> Onayla </button>  
<p id="demo"></p><script>  
function fonksiyon() {  
    var x, text;  
    x = document.getElementById("numb").value;  
    if (isNaN(x) || x < 1 || x > 10) { text = "Giriş Geçeli Değil"; } else {  
        text = "Giriş Doğru";  
    }  
    document.getElementById("demo").innerHTML = text;  
}  
</script>
```

## Otomatik Form Doğrulama

21

```
<form action="demo_form.asp" method="post">  
  <input type="text" name="ad" required>  
  <input type="submit" value="Onayla">  
</form>
```

## JavaScript Validation API

22

Property	Tanım
checkValidity()	Input elementi geçerli veri içeriyorsa True döndürür.
setCustomValidity()	Bir input elemanın doğrulama mesajı özelliğini ayarlar

```
<form action="demo_form.asp" method="post">  
  <input type="text" name="ad" required>  
  <input type="submit" value="Onayla">  
</form>
```

## JavaScript Validation API

```
<input id="id1" type="number" min="100" max="300" required>
<button onclick="myFunction()">OK</button>
<p id="demo"></p>
<script>
function myFunction() {
  var inpObj = document.getElementById("id1");
  if (inpObj.checkValidity() == false) {
    document.getElementById("demo").innerHTML = inpObj.validationMessage;
  }
}
</script>
```

## Doğrulama Kısıtlamaları DOM Özellikleri

Property	Tanım
validity	Gerçeklik durumuna bağlı olarak boolean değer alır.
validationMessage	Yanlış olma durumunda gösterilebilecek mesajı içerir.
willValidate	Doğrulanacak bir input elemanına işaret eder.

## Doğru Olma (Valid) Özellikleri

25

Property	Tanım
customError	Özel hata mesajı atandı ise True
patternMismatch	Değer desen ile eşleşmezse True
rangeOverflow	Eğer değer maksimum değerden büyükse True
rangeUnderflow	Eğer değer maksimum değerden küçükse True
tooLong	Eğer elementin maxLength özelliğindeki değeri aşarsa True.
typeMismatch	Elementin içerdiği değer türü uyumsuzsa True
valueMissing	Required olan bir element boş geçilmişse True
valid	Eğer elementin değeri geçerlisi ise True

## rangeOverflow Örnek

26

```
<input id="id1" type="number" max="100">
<button onclick="myFunction()">OK</button>
```

```
<p id="demo"></p>
<script>
function myFunction() {
  var txt = "";
  if (document.getElementById("id1").validity.rangeOverflow) {
    txt = "Değer Çok Büyük";
  }
  document.getElementById("demo").innerHTML = txt;
}
</script>
```

## rangeUnderFlow Örnek

```
<input id="id1" type="number" min="100">  
<button onclick="myFunction()">OK</button>
```

```
<p id="demo"></p>  
<script>  
function myFunction() {  
    var txt = "";  
    if (document.getElementById("id1").validity.rangeUnderflow) {  
        txt = "Value too small";  
    }  
    document.getElementById("demo").innerHTML = txt;  
}  
</script>
```

## Öğrenenden Beklenenler

JSON

Regex

Validation

konularını anlamak ve kodlamak



## Teşekkürler

Öğr. Gör. Ömer SEVİNÇ

JavaScript

Hata Yakalama Try Catch

75 Ünite 11