

## JavaScript'te selector kullanma:

Sizi teorik bilgi ile ya da bir çok tanımla sıkmaktansa konuyu birlikte oluşturacağımız bir çarpım tablosu eşliğinde anlatmaya çalışacağım.

Önce sayfamızı oluşturalım ve body içerisinde tablomuzu içerisine yerleştireceğimiz bir division (grid) tanımlayalım. Id olarak "tbl" verelim ki daha sonra yazacağımız script içerisinde ulaşabilelim.

```
<!doctype html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>Çarpım Tablosu</title>
  </head>

  <body>
    <div id="tbl" style="display: grid; grid-template-
columns: repeat(10, 50px); gap: 5px;"></div>
  </body>
</html>
```

Şimdi artık scriptimizi yazmaya başlayabiliriz. Burada dikkat etmemiz gereken şey, `<script></script>` taglarımızın yeri. Tagımızı, oluşturduğumuz gridimizden `<div>` etiketimizden sonra açmamız gerekiyor. Çünkü yazdığımız kodlar sıralı olarak okunacaklar. Yani scriptimiz çalıştığında gridimizin oluşturulmuş olması önemli.

Şimdi oluşturduğumuz grid stilindeki div içerisine satır ve sütunları oluşturacak yeni divler oluşturacağız ve tablomusdaki değerleri bu divler içerisine yazacağız. Bunu yaparken de döngüleri, for döngüsünü kullanacağız. İç içe geçmiş iki döngü. Biri sütunları, diğeri satırları oluşturacak.

```
<!doctype html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>Çarpım Tablosu</title>
  </head>

  <body>
```

```

<div id="tbl" style="display: grid; grid-template-
columns: repeat(10, 50px); gap: 5px;"></div>

<script>
  for(let i=1; i<=10; i++){
    for(let k=1; k<=10; k++){
      }
    }
  }
</script>
</body>
</html>

```

Artık içini doldurabiliriz. Bunun için pek çok yöntem mevcut. Biz bu yöntemlerden iki tanesini anlatarak işe başlayacağız. İlk yöntemimiz hepimizin yakından tanıdığı innerHTML (document nesnesinin innerHTML özelliği).

innerHTML, adından da anlaşılacağı üzere, hedef olarak verdiğimiz elementin içeriğini çağırmanızı (elde etmemizi) veya yeni bir içerik (content) oluşturmamızı sağlar. Bu içerik text şeklinde olacağı gibi biçimlendirilmiş metin, yani bir HTML metni de olabilir. Örnek:

```

document.getElementById("tbl").innerHTML =
  "<div>cell</div>";

```

Bu sayede “tbl” id’sine sahip div (grid) içerisinde bir div oluşturduk ve içerisine “cell” yazdık.

```

<script>
  for(let i=1; i<=10; i++){
    for(let k=1; k<=10; k++){
      document.getElementById("tbl").innerHTML =
        "<div>cell</div>";
    }
  }
</script>

```

Ama bir sorunumuz var. Tablomuzda tek bir hücre oluştu. Halbuki 10 satır ve 10 sütundan oluşan 100 hücre hayal etmiştik. Bunun sebebi, innerHTML komutunun her saferinde eski içeriği silerek yeniden içerik yazması. Bu bir sorun. Çünkü bizim eski içeriğimizin silinmemesi gerekiyor. Bunu aşmanın yolu yeni içeriğimizin eski içeriğimizin devamına eklenmesini sağlamak.

```

document.getElementById("tbl").innerHTML +=
  "<div>cell</div>";

```

Açılımı:

```
document.getElementById("tbl").innerHTML =  
    document.getElementById("tbl").innerHTML +  
    "<div>cell</div>";
```

İstedığınız söz dizimini kullanabilirsiniz. Tabi şu an tüm hücrelerimizin <div> içerisinde “cell” yazıyor. Bunu tablomuzun oluştuğunu görebilmek için yazdık. Daha sonra çarpım tablomuzun değerleri ile değiştireceğiz.

### **Şimdi gelelim tablomuzu oluşturmak için kullanabileceğimiz ikinci yönteme.**

Bunun için document nesnesi bize iki güzel özellik sunuyor. Yeni elementler oluşturabilmemiz için “createElement()” ve bu elementi web sayfamıza yerleştirebilmemiz için “appendChild()”.

```
<script>  
    for(let i=1; i<=10; i++){  
  
        for(let k=1; k<=10; k++){  
            var cell = document.createElement("div");  
            cell.innerHTML="cell";  
            document.getElementById("tbl").appendChild(cell);  
        }  
    }  
</script>
```

Yukarıdaki kodu incelersek, önce document.createElement(“div”); komutu ile document nesnesi (yani HTML sayfamız) içerisinde bir div oluşturduk ve bu divi onu temsil edecek olan “cell” adında bir değişkene atadık. Artık bu yeni oluşturduğumuz div üzerindeki her türlü işlemi “cell” değişkenimizi kullanarak yapabiliriz.

Daha sonra da yeni oluşturduğumuz bu divin içeriğini “cell” olarak belirledik. Ama hala sayfamızda görünen bir şey yok. Çünkü oluşturduğumuz elementin nereye yerleşeceğini belirlemedik. Henüz bir değişken içerisinde kayıtlı. Bu değişkeni HTML sayfamız içerisinde bir taga dönüştürmemiz gerekiyor. Bunun için yine selectorlerden (seçicilerden) yararlanıyoruz ve elementimizi sayfamızdaki grid içerisine yerleştiriyoruz.

```
document.getElementById("tbl").appendChild(cell);
```

Peki oluşturduğumuz hücrelere biraz şekil vermek istersek? Bunun için CSS kullanmak oldukça mantıklı bir yol. Ama bunu script içerisinden yapabilmek için de document nesnesi bize bir kolaylık sağlamış. “style” özelliği.

```
<script>  
    for(let i=1; i<=10; i++){  
  
        for(let k=1; k<=10; k++){  
            var cell = document.createElement("div");
```

```
        cell.style.backgroundColor = "yellow";
        cell.style.border = "solid 1px";
        cell.style.textAlign = "right";
        cell.innerHTML="Hücre";
        document.getElementById("tbl").appendChild(c11);
    }
}
</script>
```

Atrık son bir işimiz kaldı. Oluşturduğumuz hücrelerin içerisindeki “cell” yazısını çarpım tablomuzun değerleri ile değiştirmek. Yani sütun (i) ile satır (k) değerlerini çarpmak.

```
<script>
    for(let i=1; i<=10; i++){

        for(let k=1; k<=10; k++){
            var cell = document.createElement("div");
            cell.style.backgroundColor = "yellow";
            cell.style.border = "solid 1px";
            cell.style.textAlign = "right";
            cell.innerHTML= i * k;
            document.getElementById("tbl").appendChild(c11);
        }
    }
</script>
```