

CSS Var() İşlevi ve Değişkenler

Değişken tanımlamaktaki amaç kendini tekrar eden kodları engellemektir. Örneğin bir web sayfasındaki birçok yerde kullanılan renk kodu gibi. İleride web sayfasının template değiştiğinde o renkleri değiştirmek için tek tek tanımlanan yerlere gelip değiştirmek bir hayli zahmetli olacak. Onun yerine renk adında bir değişken tanımlayıp renk kodunu ona atandığında; renk kodunun kullanılacak yerlere değişken adını koyarak ileride gerçekleşecek bir değişiklikte yapılacak olan tek şey değişkenin değerini değiştirmek olacaktır.

CSS'te değişken tanımlamak için -- başında çift tire kullanılır. örnek --main-color:blue; gibi. CSS değişken isimleri büyük ve küçük harfe duyarlıdır. --main-color ile --Main-Color değişkenleri tarayıcılar tarafından farklı değişken ismi olarak algılanmaktadır.

- Global Değişkenler
- :root Pseudo Class Kullanımı
- Yerel Değişkenler
- Responsive Media Query İle Kullanımı
- Calc() İşlevi İle Birlikte Kullanımı
- Yedek Değişken Değeri Tanımlama

Global Değişkenler

:root Pseudo Class Kullanımı

Global yani genel CSS değişkenleri :root{} içerisinde tanımlanır ve istenilen her yerde ve her öge için tanımlanabilir. :root{} içerisinde tanımlanan değişken istenilen yerde ögeye css-property: var(--degisken-ismi) şeklinde tanımlanır.

Aşağıdaki örnekte global bir değişkenin nasıl oluşturulduğunu ve div ögesi içerisinde nasıl tanımlandığını inceleyiniz.

```
:root{
    --main-color:blue;
}

/* Tanımlanan Değişken var() işlevi ile ögeye tanımlanır */
div{
    background-color: var(--main-color);
}
```

Not: `--color` ile `--Color` aynı şey değildir! Normalde css propertylerinde case-sensitive (büyük-küçük harf duyarlılığı) olayı yoktur. Fakat değişken adlarında bu geçerli değil. *Yani değişken adları case-sensitive'dir.*

Yerel Değişkenler

Yerel yani local CSS değişkenleri `:root{}` içerisinde değil de bir öğeye tanımlı olarak oluşturulan değişkenlerdir.

Örnek olarak `div.example{--main-color: pink}` şeklinde sadece `example` classname içeren `div` öğesi ve onun alt öğeleri için kullanıldığında değeri geçerli olacaktır. local değişkenler alt öğelere miras kalma özelliğine sahiptir.

```
:root{

    --main-color:blue;

}

/* Tanımlanan Değişken var() işlevi ile öğeye tanımlanır */

div.example{

    --main-color: pink;

    background-color: var(--main-color); /* Öğenin arka pla
renge pink olacaktır */

}
```

Responsive Media Query İle Kullanımı

Global ve yerel değişkenleri responsive media sorguları ile kullanabilirsiniz. Örnek olarak aşağıda tek bir yerel ve global değiştirilerek mobil ve desktop cihazlar için site genelinde tasarıma müdahale etmenize olanak sağlar.

```
:root{

    --main-font-size: 30px;

    --main-color: blue;

}

.container{

    font-size: var(--main-font-size);

    --main-color: green;
```

```
}

/* 450px ve altı görüntü alanında hem genel değişkeni hem
de yerel değişken değişiyor */

@media screen and (max-width: 480px){

  :root{

    --main-font-size: 18px;

  }

  .container{

    --main-Color: red;

  }

}
```

Calc() İşlevi İle Birlikte Kullanımı

CSS değişkenlerini [calc\(\)](#) işlevi ile birlikte tanımlayabilirsiniz. Aşağıdaki örnekte **margin-bottom** değerini **--spacing** değişkeninin 3 katı olarak tanımlayalım.

```
:root{

  --spacing: 1rem;

}

div.example{

  margin-bottom: calc(var(--spacing) * 3); /* Margin
Bottom Değeri = 3rem olacaktır */

}
```

Yedek Değişken Değeri Tanımlama

Yedek değişken değeri (fallback) tanımlamak henüz bir --degisken tanımlanmadı ise nasıl davranması gerektiğini belirtmek için kullanılır. Her değişken "," ile ayrılmalıdır ve ilk yazılan değişkenden sonraki değişkenler yine **var** içerisinde tanımlanmalıdır. Yoksa işlevi geçersiz olacaktır.

```
.container{

  color: var(--main-color, red); /* Eğer --main-color
tanımlı değil ise "red" olacak */

}
```

```

    }

    .container{

        color: var(--main-color, var(--main-color2, red)); /*
Eğer --main-color ve --main-color2 tanımlı değil ise "red"
olacak */

    }

    .container{

        color: var(--main-color,--main-color2, red); /* "--
main-color2, red" ifadesi geçersizdir yukarıdaki gibi var()
içerisinde yazılmalıdır. */

    }

```

Css Değişkenleri ve JavaScript

JavaScript ile css değişkenlerine erişebilir, okuyabilir, değişiklikler yapabilirsiniz.

```

/* JS */
const box = document.querySelector('.box');
// Classı box olanı seçtik.const boxStyles =
getComputedStyle(box);
// Box'ın style'ına eriştik.const boxColor =
boxStyles.getPropertyValue(--box-color);
// Eriştığımız style'lerden --box-color değerini
aldık.boxColor.style.setProperty('--box-color', #ccc);
// set ile yeni özellik eklemiş olduk.

```