

BEM یا Block Element Modifier یک روش نام‌گذاری کلاس‌های CSS است که به کاربران این امکان را می‌دهد تا با استفاده از نام‌گذاری مناسب، کدهای CSS خود را بهبود بخشند.

1. **Block**: بلاک یک کامپوننت یا یک بخش از DOM است قسمت اصلی صفحه وب شماست. به عنوان مثال، یک بلاک می‌تواند **"header"** برای هدر سایت یا **"button"** برای یک دکمه باشد. یا **div** یا **ul**

2. **Element\*\*** (المان): \*\*المان‌ها به عنوان بخش‌های داخلی بلاک تعریف می‌شوند. معمولاً المان‌ها در داخل یک بلاک قرار می‌گیرند و به کلاس بلاک اضافه می‌شوند. برای مثال، برای بلاک **"button"**، یک المان می‌تواند **"text"** باشد.

3. **Modifier\*\*** (اصلاحگر): \*\*اصلاحگرها به شما امکان می‌دهند ویژگی‌های مختلف یک بلاک یا المان را تغییر دهید. اصلاحگرها به عنوان کلاس‌های اضافی به المان‌ها یا بلاک‌ها اضافه می‌شوند. مثلاً، یک اصلاحگر می‌تواند **"large"** و یا ویژگی یا حالت خاص یک بلوک یا عنصر است، مانند **active** یا **disabled**. برای یک **دکمه بزرگ** اضافه کند. حالا برای مثال، نگاهی چگونگی استفاده از **BEM** در کدهای **HTML** و **CSS** بیندازیم:

[block]\_\_[Element]--[Modifier]

**button\_\_primary--active**

```
<div class="button button--large">
  <span class="button__text">Click me</span>
</div>
```

```
.button {
```

```
  /* اندازه و استایل پیش‌فرض دکمه */
```

```
}
```

```
.button--large {
```

```
  /* استایل برای دکمه‌های بزرگ با استفاده از اصلاحگر */
```

```
}
```

```
.button__text {
```

```
  برای جدا سازی بلوک از المنت - استفاده میشود
```

به شما کمک می‌کند تا ساختار مناسبی در کدهای وب خود داشته باشید و به **BEM** استفاده از راحتی تغییرات را اعمال کنید.

در این مثال، **button** نام بلوک است و **primary--** و **large--** نام مدیفایرها هستند.

در اینجا چند مثال دیگر از نام‌گذاری متغیرهای معنادار BEM آورده شده است:

- \* `**input-email-focus**` برای یک متغیر CSS که به یک ورودی متنی ایمیل متمرکز شده اشاره می‌کند.
- \* `**nav-bar-active**` برای یک متغیر CSS که به یک نوار پیمایش فعال اشاره می‌کند.
- \* `**modal-close-button**` برای یک متغیر CSS که به یک دکمه بستن پنجره مودال اشاره می‌کند.

با استفاده از نام‌گذاری متغیرهای معنادار BEM، می‌توانید کد CSS خود را واضح‌تر، قابل نگهداری‌تر و قابل درک‌تر کنید.

نام‌گذاری متغیرهای معنادار BEM مزایای زیادی دارد، از جمله:

- خوانایی و درک کد را بهبود می‌بخشد. با استفاده از این روش نام‌گذاری، می‌توان به راحتی تشخیص داد که هر متغیر مربوط به چه بخش از کد است.
- به کاهش خطا کمک می‌کند. با استفاده از این روش نام‌گذاری، احتمال استفاده از متغیرهای اشتباه کاهش می‌یابد.
- به بهبود قابلیت نگهداری کد کمک می‌کند. با استفاده از این روش نام‌گذاری، کد راحت‌تر قابل درک و توسعه است.

در اینجا چند نکته برای استفاده از نام‌گذاری متغیرهای معنادار BEM آورده شده است:

- از حروف کوچک استفاده کنید. نام متغیرهای جاوا اسکریپت باید با حروف کوچک نوشته شوند.

از کاراکترهای غیر معمول استفاده نکنید. از کاراکترهای غیر معمول مانند فاصله، خط تیره و زیرخط در نام متغیرها استفاده نکنید.

از نام‌های طولانی استفاده نکنید. نام متغیرها نباید بیش از چند کلمه باشند. با استفاده از این نکات، می‌توانید از نام‌گذاری متغیرهای معنادار BEM برای بهبود خوانایی، درک، قابلیت نگهداری و کاهش خطا در کد خود استفاده کنید.

از نام‌های کوتاه و مختصر استفاده کنید.

\* از نام‌های توصیفی استفاده کنید که معنای متغیر را روشن کند.

\* از - برای جدا کردن بلوک‌ها، عناصر و اصلاح‌کننده‌ها استفاده کنید.

\* از حروف بزرگ برای بلوک‌ها و حروف کوچک برای عناصر و اصلاح‌کننده‌ها استفاده کنید. با پیروی از این نکات، می‌توانید کد CSS خود را واضح‌تر و قابل نگهداری‌تر کنید.

در این مثال، `button` نام بلوک است

در این مثال، `myDiv` نام بلوک HTML است، `div` نام عنصر HTML است و `width` نام اصلاح‌کننده است

```
const myDivWidth = 100;
```

ثبت `<button class="button button--primary button--large">`  
`</button>` نام