




BEM

Block Element Modifier

 2026-01   20 min.

¿Qué es BEM?

BEM - Block Element Modifier

BEM es una metodología de nombrado de clases CSS.

Objetivo: Resolver problemas de:

- Colisión de nombres de clases
- Especificidad excesiva
- Código difícil de mantener
- Falta de estructura en proyectos grandes

Resultado: Código CSS más predecible y organizado.

Estructura BEM

Los Tres Componentes

Block (Bloque):

- Componente independiente y reutilizable
- Ejemplo: `.card` , `.menu` , `.button`

Element (Elemento):

- Parte de un bloque que no tiene sentido por sí sola
- Se separa con doble guion bajo: `__`
- Ejemplo: `.card__title` , `.menu__item` , `.button__icon`

Modifier (Modificador):

- Variación de un bloque o elemento
- Se separa con doble guion: `--`
- Ejemplo: `.card--featured` , `.button--primary` , `.menu__item--active`

Sintaxis BEM

```
/* BLOCK */  
.block { }  
  
/* ELEMENT */  
.block__element { }  
  
/* MODIFIER de BLOCK */  
.block--modifier { }  
  
/* MODIFIER de ELEMENT */  
.block__element--modifier { }
```

Regla importante: Los elementos NO se anidan en la nomenclatura.

✗ **Incorrecto:** `.block__element__subelement`

✓ **Correcto:** `.block__subelement`

Block (Bloque)

Block - Componente Independiente

Un **bloque** es un componente independiente que puede existir por sí solo.

Características:

- Nombre describe su **propósito**, no su apariencia
- Puede contener otros bloques
- No debe tener margin o position (para ser reutilizable)

Ejemplos:

```
.card { }      /* Tarjeta */  
.menu { }      /* Menú */  
.button { }     /* Botón */  
.header { }     /* Encabezado */  
.sidebar { }    /* Barra lateral */  
.form { }       /* Formulario */
```


Block - Ejemplo

```
<div class="card">
  <!-- Contenido de la tarjeta -->
</div>
```

```
.card {
  background: white;
  border-radius: 8px;
  padding: 20px;
  /* NO margin o position aquí */
}
```

Reutilizable en cualquier lugar:

```
<div class="sidebar">
  <div class="card">...</div>
</div>

<main>
  <div class="card">...</div>
</main>
```

Element (Elemento)

Element - Parte del Bloque

Un **elemento** es una parte del bloque que no tiene sentido fuera de él.

Características:

- Separado por doble guion bajo: `__`
- No puede existir independientemente
- Describe una parte del bloque

Ejemplos:

```
.card__header { }    /* Encabezado de la tarjeta */  
.card__title { }     /* Título de la tarjeta */  
.card__body { }      /* Cuerpo de la tarjeta */  
.card__footer { }    /* Pie de la tarjeta */  
.card__image { }     /* Imagen de la tarjeta */  
.card__button { }    /* Botón de la tarjeta */
```

Element - Ejemplo

```
<div class="card">
  <div class="card__header">
    <h2 class="card__title">Título</h2>
  </div>
  <div class="card__body">
    <p class="card__text">Contenido</p>
  </div>
  <div class="card__footer">
    <button class="card__button">Acción</button>
  </div>
</div>
```

```
.card { }
.card__header { }
.card__title { font-size: 1.5rem; }
.card__body { padding: 15px; }
.card__footer { border-top: 1px solid #ccc; }
.card__button { }
```

Elements - NO Anidación en Nombres

✗ Incorrecto:

```
<div class="card">
  <div class="card__body">
    <p class="card__body__text">Texto</p>
  </div>
</div>
```

```
.card__body__text { } /* ✗ Muy específico */
```

✓ Correcto:

```
<div class="card">
  <div class="card__body">
    <p class="card__text">Texto</p>
  </div>
</div>
```

```
.card__text { } /* ✓ Simple y claro */
```

Modifier (Modificador)

Modifier - Variaciones

Un **modificador** define una variación, estado o apariencia diferente de un bloque o elemento.

Características:

- Separado por doble guion: `--`
- Describe **cómo se ve, cómo se comporta, o qué estado tiene**
- Siempre se usa junto con la clase base

Ejemplos:

```
.button--primary { }      /* Botón primario */  
.button--large { }        /* Botón grande */  
.button--disabled { }     /* Botón deshabilitado */  
.card--featured { }       /* Tarjeta destacada */  
.card--dark { }           /* Tarjeta oscura */  
.menu__item--active { }   /* Item activo */
```

Modifier - Ejemplo

```
<!-- Botón normal -->
<button class="button">Normal</button>

<!-- Botón primario (con modificador) -->
<button class="button button--primary">Primario</button>

<!-- Botón grande y primario (múltiples modificadores) -->
<button class="button button--primary button--large">Grande</button>
```

```
.button {
  padding: 10px 20px;
  border: none;
  background: gray;
}

.button--primary {
  background: blue;
  color: white;
}

.button--large {
  padding: 15px 30px;
  font-size: 1.2rem;
}
```


Modifier - Siempre con Clase Base

✗ Incorrecto (solo modificador):

```
<button class="button--primary">Botón</button>
```

✓ Correcto (base + modificador):

```
<button class="button button--primary">Botón</button>
```

Razón: El modificador solo agrega cambios incrementales, la clase base proporciona los estilos fundamentales.

```
.button {  
  /* Estilos base comunes a todos los botones */  
  padding: 10px 20px;  
  border-radius: 4px;  
}  
  
.button--primary {  
  /* Solo los cambios para la variante primaria */  
  background: blue;  
  color: white;  
}
```

Ejemplo Completo

Ejemplo Completo - Tarjeta de Producto

```
<article class="product-card product-card--featured">
  <div class="product-card__header">
    
    <span class="product-card__badge product-card__badge--new">Nuevo</span>
  </div>

  <div class="product-card__body">
    <h3 class="product-card__title">Nombre del Producto</h3>
    <p class="product-card__description">Descripción breve</p>
    <span class="product-card__price">$99.99</span>
  </div>

  <div class="product-card__footer">
    <button class="product-card__button product-card__button--primary">
      Comprar
    </button>
    <button class="product-card__button product-card__button--secondary">
      Ver más
    </button>
  </div>
</article>
```

Ejemplo Completo - CSS

```
/* Block */  
.product-card {  
  border: 1px solid #ddd;  
  border-radius: 8px;  
  background: white;  
}  
  
.product-card--featured {  
  border-color: gold;  
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
}
```

Ejemplo Completo - CSS

```
/* Elements */
.product-card__header { position: relative; }
.product-card__image { width: 100%; }
.product-card__badge {
  position: absolute;
  top: 10px;
  right: 10px;
}
.product-card__badge--new {
  background: green;
  color: white;
}
.product-card__title { font-size: 1.5rem; }
.product-card__description { color: #666; }
.product-card__price {
  font-size: 1.25rem;
  font-weight: bold;
}
```

Ejemplo Completo - CSS

```
.product-card__footer {  
  display: flex;  
  gap: 10px;  
}  
.product-card__button { padding: 10px 20px; }  
.product-card__button--primary {  
  background: blue;  
  color: white;  
}  
.product-card__button--secondary {  
  background: gray;  
}
```

Ventajas de BEM

Ventajas de Usar BEM

- ✓ **Nombres descriptivos y específicos** - Se entiende la estructura con solo leer las clases
- ✓ **Evita conflictos de nombres** - Cada clase es única y específica
- ✓ **Especificidad baja y plana** - No hay anidación profunda de selectores
- ✓ **Reutilización fácil** - Los bloques son independientes
- ✓ **Escalabilidad** - Funciona bien en proyectos grandes
- ✓ **Trabajo en equipo** - Convención clara que todos pueden seguir
- ✓ **Facilita encontrar y modificar estilos** - Estructura clara y predecible

Errores Comunes

Errores Comunes y Soluciones

✗ Anidar elementos en nombres:

```
.card__body__text { } /* MAL */
```

✓ `.card_text { }`

✗ Usar solo modificadores:

```
<div class="card--featured"> <!-- MAL -->
```

✓ `.card`

✗ Modificadores que no describen variaciones:

```
.card__title--red { } /* MAL - describe apariencia */
```

✓ `.card__title--error { }` /* Describe propósito */

✗ Crear bloques demasiado genéricos:

BEM vs Utility-First

BEM:

```
<button class="button button--primary button--large">  
  Comprar  
</button>
```

Utility-First (Tailwind):

```
<button class="bg-blue-500 text-white px-6 py-3 rounded-lg text-lg">  
  Comprar  
</button>
```

BEM = Componentes semánticos Utility = Utilidades atómicas

Mejores Prácticas

Mejores Prácticas con BEM

Recursos Adicionales

Documentación Oficial:

- BEM Official: <https://en.bem.info/>
- BEM Methodology: <https://en.bem.info/methodology/>

Artículos:

- CSS Tricks: BEM 101
- Smashing Magazine: BEM Guide

Resumen - BEM en 60 Segundos

Block → Componente independiente

```
.card { }
```

Element → Parte del bloque (__)

```
.card__title { }  
.card__button { }
```

Modifier → Variación o estado (-)

```
.card--featured { }  
.card__button--primary { }
```

Resumen - BEM en 60 Segundos

Reglas de oro:

1. Los elementos NO se anidan en nombres
2. Los modificadores siempre van con la clase base
3. Nombres descriptivos, no de apariencia
4. Especificidad plana

