

# Guía de Pseudoselectores en CSS

Los pseudoselectores en CSS son palabras clave que se agregan a los selectores para especificar un estado especial de los elementos seleccionados. Son muy útiles para estilizar elementos en base a condiciones específicas, como el estado de un enlace, el primer hijo de un elemento, o incluso elementos específicos en función de sus padres o hermanos.

A continuación, se presenta una explicación de los pseudoselectores más comunes con ejemplos:

---

## Pseudoselectores de Enlaces y Estados

**:link** Selecciona los enlaces que no han sido visitados.

```
a:link {  
  color: blue;
```

1. }

**:visited** Selecciona los enlaces que ya han sido visitados.

```
a:visited {  
  color: purple;
```

2. }

**:hover** Aplica estilos a un elemento mientras el puntero del ratón está sobre él.

```
a:hover {  
  color: red;
```

3. }

**:active** Selecciona un elemento mientras está siendo activado, por ejemplo, cuando un enlace está siendo clicado.

```
a:active {  
  color: orange;
```

4. }

---

## Pseudoselectores de Estructura

**:first-child** Selecciona el primer hijo de su padre.

```
p:first-child {  
  font-weight: bold;
```

1. }

**:last-child** Selecciona el último hijo de su padre.

```
p:last-child {  
    font-style: italic;  
}
```

2. }

**:nth-child(n)** Selecciona el enésimo hijo (n puede ser un número, una palabra clave como **odd** o **even**, o una expresión).

```
li:nth-child(odd) {  
    background-color: #f0f0f0;  
}
```

```
li:nth-child(3) {  
    color: red;  
}
```

3. }

**:nth-last-child(n)** Selecciona el enésimo hijo desde el final.

```
li:nth-last-child(1) {  
    font-size: 1.2em;  
}
```

4. }

**:only-child** Selecciona un elemento si es el único hijo de su padre.

```
p:only-child {  
    text-align: center;  
}
```

5. }

---

## Pseudoselectores de Tipo

**:first-of-type** Selecciona el primer elemento de su tipo.

```
h1:first-of-type {  
    color: navy;  
}
```

1. }

**:last-of-type** Selecciona el último elemento de su tipo.

```
h1:last-of-type {  
    color: teal;  
}
```

2. }

**:nth-of-type(n)** Selecciona el enésimo elemento de su tipo.

```
p:nth-of-type(2) {  
}
```

```
text-decoration: underline;
```

3. }

**:nth-last-of-type(n)** Selecciona el enésimo elemento de su tipo contando desde el final.

```
h2:nth-last-of-type(1) {  
  background-color: lightgray;
```

4. }

**:only-of-type** Selecciona un elemento si es el único de su tipo en su contenedor.

```
h1:only-of-type {  
  font-size: 2em;
```

5. }

---

## Pseudoselectores de Estados del Formulario

**:focus** Selecciona un elemento que tiene el foco.

```
input:focus {  
  border-color: blue;
```

1. }

**:checked** Selecciona un elemento de formulario que está marcado o seleccionado (como un checkbox o radio).

```
input:checked {  
  background-color: yellow;
```

2. }

**:disabled** Selecciona los elementos de formulario deshabilitados.

```
input:disabled {  
  color: gray;
```

3. }

**:enabled** Selecciona los elementos de formulario habilitados.

```
input:enabled {  
  color: black;
```

4. }

**:required** Selecciona los campos de formulario requeridos.

```
input:required {  
  border: 2px solid red;
```

5. }

**:optional** Selecciona los campos de formulario que no son requeridos.

```
input:optional {  
  border: 2px solid green;
```

6. }

---

## Pseudoselectores Misceláneos

**:not(selector)** Selecciona todos los elementos que no coincidan con el selector dado.

```
p:not(.important) {  
  color: gray;
```

1. }

**:root** Selecciona el elemento raíz del documento.

```
:root {  
  --main-color: #ff6600;
```

2. }

**:empty** Selecciona elementos que no tienen hijos (ni texto ni elementos).

```
div:empty {  
  display: none;
```

3. }

**:target** Selecciona un elemento cuyo ID coincide con el fragmento de la URL actual.

```
#section1:target {  
  background-color: yellow;
```

4. }

---

Este documento cubre los pseudoselectores principales de CSS con ejemplos que puedes probar en tus propios proyectos. Los pseudoselectores son herramientas poderosas para diseñar aplicaciones web modernas de manera más eficiente.

# Guía de Flexbox y Grid en CSS

CSS Flexbox y Grid son dos sistemas de diseño que permiten crear estructuras de diseño modernas y responsivas de forma sencilla.

## Flexbox

Flexbox (Flexible Box) es un modelo de diseño unidimensional que organiza elementos en filas o columnas.

### Propiedades del Contenedor Flex

**display: flex;** Define un contenedor flex.

```
.flex-container {  
  display: flex;
```

1. }

**flex-direction** Especifica la dirección de los elementos hijos.

```
.flex-container {  
  flex-direction: row; /* row, row-reverse, column, column-reverse */
```

2. }

**justify-content** Alinea los elementos en el eje principal.

```
.flex-container {  
  justify-content: center; /* flex-start, flex-end, center, space-between, space-around,  
  space-evenly */
```

3. }

**align-items** Alinea los elementos en el eje transversal.

```
.flex-container {  
  align-items: stretch; /* flex-start, flex-end, center, baseline, stretch */
```

4. }

**flex-wrap** Define si los elementos se ajustan o no.

```
.flex-container {  
  flex-wrap: wrap; /* nowrap, wrap, wrap-reverse */
```

5. }

### Propiedades de los Elementos Flex

**flex** Establece el tamaño flexible de un elemento.

```
.flex-item {  
  flex: 1; /* grow shrink basis */
```

1. }

**align-self** Alinea un elemento en el eje transversal, sobrescribiendo **align-items**.

```
.flex-item {  
    align-self: center; /* auto, flex-start, flex-end, center, baseline, stretch */
```

2. }

---

## Grid

CSS Grid es un modelo de diseño bidimensional que organiza elementos en filas y columnas.

### Propiedades del Contenedor Grid

**display: grid;** Define un contenedor grid.

```
.grid-container {  
    display: grid;
```

1. }

**grid-template-rows y grid-template-columns** Define el tamaño de las filas y columnas.

```
.grid-container {  
    grid-template-rows: 100px 200px;  
    grid-template-columns: 1fr 2fr;
```

2. }

**gap** Establece el espacio entre filas y columnas.

```
.grid-container {  
    gap: 10px; /* row-gap column-gap */
```

3. }

**justify-items y align-items** Alinea los elementos dentro de sus celdas.

```
.grid-container {  
    justify-items: center; /* start, end, center, stretch */  
    align-items: center;
```

4. }

### Propiedades de los Elementos Grid

**grid-column y grid-row** Define en qué filas o columnas se coloca un elemento.

```
.grid-item {
```

```
grid-column: 1 / 3; /* inicio / fin */  
grid-row: 2 / 4;
```

1. }

**justify-self y align-self** Alinea un elemento dentro de su celda.

```
.grid-item {  
  justify-self: end; /* start, end, center, stretch */  
  align-self: stretch;
```

2. }

---

Este documento proporciona una visión general de Flexbox y Grid, junto con ejemplos básicos para ayudarte a comenzar a usar estos potentes sistemas de diseño en tus proyectos web.