

# Les enfants et les voisins

Temps de lecture : 2 minutes



## Stylisation basée sur l'état d'un élément frère

Vous pouvez marquer un élément frère avec la classe `peer` et utilisez des modificateurs comme `peer-invalid` pour styliser un élément cible basé sur l'état de cet élément frère.

Cette technique fonctionne avec tous les modificateurs pseudo-classes et est limitée aux frères précédents en raison de la combinaison de sélecteurs en `CSS`.

En utilisant des noms uniques pour chaque pair (`peer/{nom}`), il est possible de styliser des éléments spécifiquement en fonction de l'état d'un pair nommé, en utilisant des classes comme `peer-checked/{nom}`. Ceci est identique à ce que nous avons vu dans la leçon précédente sur les groupes.

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="./output.css" rel="stylesheet" />
  </head>
  <body>
    <div class="flex h-screen flex-col items-center justify-center gap-6">
      <div class="px-6 pb-5 pt-6 shadow">
        <form>
          <div>
            <label for="email" class="block text-sm font-medium text-slate-700">
              Email</label>
            <div class="mt-1">
              <input
                type="email"
                id="email"
                class="peer w-full rounded-md border border-slate-300 px
```

```

-3 py-2 placeholder-slate-400 shadow-sm invalid:border-pink-500 invalid:
text-pink-600 focus:border-sky-500 focus:outline-none focus:ring-1 focu
s:ring-sky-500 focus:invalid:border-pink-500 focus:invalid:ring-pink-50
0"

        placeholder="exemple@exemple.fr"
    />
    <p
        class="invisible mt-2 text-sm text-pink-600 peer-invali
d:visible"
    >
        Fournissez un email valide.
    </p>
</div>
</div>
</form>
</div>
</div>
</body>
</html>

```

Remarquez bien l'application de la classe `peer` sur le champ et l'utilisation de du modificateur `peer-invalid:visible` sur le paragraphe.

## Stylisation des enfants directs

Utilisez le modificateur `*` pour styliser les enfants directs d'un élément pour les situations où vous n'avez pas le contrôle direct sur ces éléments.

Notez bien que les styles appliqués directement sur les enfants ne peuvent pas être écrasés en ajoutant d'autres classes sur ces enfants directs.

```

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.
0" />
    <link href="./output.css" rel="stylesheet" />
  </head>
  <body>

```

```

<div class="flex h-screen flex-col items-center justify-center gap-
6">
  <div class="w-1/5 px-6 pb-5 pt-6 shadow">
    <h2 class="text-2xl text-blue-600 dark:text-blue-300">
      Langages de programmation
    </h2>
    <div
      class="flex flex-wrap gap-2 pt-4 text-sm text-blue-600 *:round
ed-full *:border *:border-blue-100 *:bg-blue-50 *:px-2 *:py-0.5"
    >
      <div>Java</div>
      <div>Python</div>
      <div>PHP</div>
      <div>JavaScript</div>
      <div>TypeScript</div>
      <div>HTML</div>
      <div>CSS</div>
      <div>SQL</div>
    </div>
  </div>
</div>
</body>
</html>

```

## Stylisation basée sur les descendants

Utilisez le modificateur `has-*` pour styliser un élément en fonction de l'état ou du contenu de ses descendants.

Cela fonctionne avec des pseudo-classes ou des sélecteurs d'éléments.

```

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.
0" />
    <link href="./output.css" rel="stylesheet" />
  </head>
  <body>

```

```

<div class="flex h-screen flex-col items-center justify-center gap-
6">
  <div class="w-1/5 px-6 pb-5 pt-6 shadow">
    <h2 class="text-lg font-bold text-indigo-900">Langage Favori</h2
  >
    <fieldset>
      <div class="my-4 space-y-2">
        <label
          for="javascript"
          class="flex items-center justify-between rounded-lg p-4 te
xt-slate-700 ring-1 ring-transparent hover:bg-slate-100 has-[:checked]:b
g-indigo-50 has-[:checked]:text-indigo-900 has-[:checked]:ring-indigo-20
0"
        >
          JavaScript
        <input
          id="javascript"
          name="langage"
          value="javascript"
          type="radio"
          class="box-content h-1.5 w-1.5 appearance-none rounded-f
ull border-[5px] border-white bg-white bg-clip-padding outline-none ring
-1 ring-gray-950/10 checked:border-indigo-500 checked:ring-indigo-500"
        />
      </label>
    </div>
    <div class="my-4 space-y-2">
      <label
        for="php"
        class="flex items-center justify-between rounded-lg p-4 te
xt-slate-700 ring-1 ring-transparent hover:bg-slate-100 has-[:checked]:b
g-indigo-50 has-[:checked]:text-indigo-900 has-[:checked]:ring-indigo-20
0"
      >
        PHP
      <input
        id="php"
        name="langage"
        value="php"
        type="radio"
        class="box-content h-1.5 w-1.5 appearance-none rounded-f
ull border-[5px] border-white bg-white bg-clip-padding outline-none ring

```

```

-1 ring-gray-950/10 checked:border-indigo-500 checked:ring-indigo-500"
    />
  </label>
</div>
<div class="my-4 space-y-2">
  <label
    for="python"
    class="flex items-center justify-between rounded-lg p-4 text-slate-700 ring-1 ring-transparent hover:bg-slate-100 has-[:checked]:bg-indigo-50 has-[:checked]:text-indigo-900 has-[:checked]:ring-indigo-200"
  >
    Python
    <input
      id="python"
      name="langage"
      value="python"
      type="radio"
      class="box-content h-1.5 w-1.5 appearance-none rounded-full border-[5px] border-white bg-white bg-clip-padding outline-none ring-1 ring-gray-950/10 checked:border-indigo-500 checked:ring-indigo-500"
    />
  </label>
</div>
</fieldset>
</div>
</div>
</body>
</html>

```

## Stylisation basée sur les descendants d'un groupe

Marquez un parent avec la classe `group` et utilisez `group-has-*` pour styliser un élément cible basé sur les descendants de ce parent.

## Stylisation basée sur les descendants d'un pair

Permet de styliser un élément en fonction des descendants d'un élément frère marqué avec la classe `peer`. Il faut utiliser `peer-has-*`.

# Code de la vidéo

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="./output.css" rel="stylesheet" />
  </head>
  <body>
    <div class="flex h-screen flex-col items-center justify-center gap-6">
      <ul class="w-1/3 hover:*:text-red-400">
        <li>
          Lorem ipsum dolor sit amet consectetur, adipisicing elit. Unde corrupti
        </li>
        <li>
          Qui nostrum. Possimus autem consequatur in id repudiandae quae
          dolorum voluptates officiis aperiam. Repellat veritatis asperiores
          error corporis omnis numquam.
        </li>
      </ul>
      <div class="flex has-[:checked]:bg-sky-500">
        <input class="peer" type="checkbox" />
        <p class="invisible peer-checked:visible">Bravo</p>
      </div>
      <div class="flex">
        <div class="peer">
          <input type="checkbox" />
        </div>
        <p class="invisible peer-has-[:checked]:visible">Bravo 2 !</p>
      </div>
    </div>
  </body>
</html>
```