

Advanced Selectors

Important games: CSS Dinner

A. Combinators

- `space` - the descendant combinator
- `>` - the child combinator
- `+` - the adjacent sibling combinator
- `~` - the general sibling combinator

Example:

```
1 <main class="parent">
2   <div class="child group1">
3     <div class="grand-child group1"></div>
4   </div>
5   <div class="child group2">
6     <div class="grand-child group2"></div>
7   </div>
8   <div class="child group3">
9     <div class="grand-child group3"></div>
10  </div>
11 </main>
```

Following: muncul setelah

1) Descendant (keturunan)

```
1 main div {
```

This will select all 'child' and 'grand-child' <div> inside of <main>

2) Child (anak)

```
2 main > div {
```

This will only select direct children(of main, which is div with class '.child')

3) Adjacent sibling (saudara terdekat)

To select an element that is (following) adjacent on the same level of indentation.

```
2 .group1 + div {
```

This will only select the div with the class child group2

```
7 .group1 + div + div {
```

This will only select the div with the class child group3

4) General sibling (saudara umum)

To select all of the siblings following an element.

```
2 .group1 ~ div {
```

This will select all of .group1 siblings - in this case the 2nd and 3rd .child div

B. Pseudo-selectors

1. Pseudo-classes (:single-colon)

Untuk memilih elemen berdasarkan keadaan tertentu. Misal saat elemen dalam keadaan hover, fokus, atau saat elemen adalah anak pertama

1) Dynamic & user action

These can make your page feel much dynamic and interactive

`:focus` Menargetkan elemen saat elemen tersebut mendapat fokus, seperti input form yang sedang aktif.

`:hover` Menargetkan elemen saat pengguna mengarahkan kursor ke atasnya.

`:active` Menargetkan elemen saat sedang diklik

`:link` Menargetkan unvisited link

`:visited` Menargetkan visited link

2) Structural pseudo-classes

A powerful way to select elements based on their position within the DOM.

`:root` Menargetkan elemen root dari dokumen, yaitu `<html>`.

`:first-child` Menargetkan first child of any element

`:last-child` Menargetkan last child of any element

`p:only-child` Selects every `<p>` element that is the only child of its parent

`:empty` Menargetkan elemen yang tidak memiliki anak

`:nth-child(n)` Menargetkan elemen anak ke-n dalam parent

`:nth-of-type(n)` Menargetkan elemen(dengan tipe yg sama) ke-n dalam parent

Perbedaan `:nth-child(n)` & `:nth-of-type(n)`

```
<div>
  <p>Paragraf pertama</p>
  <span>Span pertama</span>
  <p>Paragraf kedua</p>
  <span>Span kedua</span>
  <p>Paragraf ketiga</p>
</div>
```

Salin kode

```
/* nth-child */
div p:nth-child(3) {
  color: red; /* Memilih paragraf kedua karena itu adalah elemen anak ketiga dalam parent */
}

/* nth-of-type */
div p:nth-of-type(2) {
  color: blue; /* Memilih paragraf kedua karena itu adalah elemen <p> kedua dalam parent */
}
```

Kesimpulan

- ``:nth-child(n)`` memilih elemen berdasarkan posisinya di antara semua anak elemen dari parent, tanpa memandang tipe elemen tersebut.
- ``:nth-of-type(n)`` memilih elemen berdasarkan posisinya di antara elemen-elemen dari tipe yang sama dalam parent.

2. Pseudo-elements (:double-colon)

Untuk menata bagian spesifik dari elemen yang sudah ada. Mereka memungkinkan developer untuk membuat elemen yang sebenarnya tidak ada dalam markup HTML, seperti menambahkan konten sebelum atau sesudah elemen, atau menata huruf pertama dari paragraf.

```
::before Menambahkan elemen sebelum elemen yg dipilih.  
  
::after Menambahkan elemen sesudah elemen yg dipilih.  
  
::first-letter Menargetkan huruf pertama dari elemen text.  
  
::first-line Menargetkan baris pertama dari elemen text.  
  
::marker Menargetkan penanda dari item <li> (seperti bullet atau angka dalam <li>).  
  
::selection Menargetkan bagian dari elemen yang dipilih oleh pengguna, seperti saat teks di-highlight.
```

C. Attribute Selectors

Attribute is anything in the opening tag of an HTML element, such as `src='picture.jpg'` or `href="www.web.com"`.

- `[attribute]` - This general selector will select anything where the given attribute exists. Its value doesn't matter.
- `[attribute="value"]` - To get really specific, we can use `=` to match a specific attribute with a specific value.
- `selector[attribute]` - Optionally we can combine our attribute selectors with other types of selectors, such as class or element selectors.
- `selector[attribute="value"]` - To get really specific

Based on how we access:

- `[attribute^="value"]` - `^=` Will match strings from the start.
- `[attribute$="value"]` - `$=` Will match strings from the end.
- `[attribute*="value"]` - `*=` The wildcard selector will match anywhere inside the string.

Example:

```
1  [class^='aus'] {  
2    /* Classes are attributes too!  
3     This will target any class that begins with 'aus':  
4     class='austria'  
5     class='australia'  
6     */  
7  }
```

```
9  [src$='.jpg'] {
10     /* This will target any src attribute that ends in '.jpg':
11     src='puppy.jpg'
12     src='kitten.jpg'
13     */
14 }
15
16 [for*='ill'] {
17     /* This will target any for attribute that has 'ill' anywhere
18     for="bill"
19     for="jill"
20     for="silly"
21     for="ill"
22     */
```