# 1 CSS Selector type

CSS selector dibagi menjadi 5 categories, yaitu: basic selector, complex selector, attribute selector,

pseudoclasses, dan pseudoelements.

**Basic Selector** adalah selector yang menunjuk satu (single) node dari html dengan identitas html tag

tersebut.

**Complex Selector** adalah selector yang menggunakan combinator (karakter penggabung) untuk

mengekspresikan hubungan suatu element dengan element lainnya.

**Attribute Selector** adalah selector yang fokus melakukan pilihannya dengan menspesifikasikan dan

memfilter attribute dan nama attribute yang dimiliki element tersebut.

**Pseudo-classes** adalah component penambah selector yang digunakan untuk menambah style dalam

situasi atau kondisi yang tidak bisa diekspresikan oleh basic, complex dan attribute selector. Penulisan pseudoclasses ditandai dengan : (colon).

**Pseudo-elements** adalah cara untuk melakukan styling pada content yang sesungguhnya tidak exist di

dalam html document, sehingga harus ditambah element tambahan/element semu (pseudo element).

Penulisan pseudoelements ditandai dengan :: (doublecolon).

**(Lihat table pada html file css-selector-references.html)**

# 2 Class vs ID vs Data Attribute

Class dan ID adalah 2 attribute html yang bisa digunakan untuk html tag manapun. Tujuan class dan id sama, yaitu memberikan sebuah identitas pada html tag, agar selector mudah untuk menemukan html tag yang ingin di style, di akses atau dimodifikasi.

Perbedaan class dan id adalah, class yang sama bisa digunakan berkali-kali pada satu halaman html yang sama, tetapi id tidak bisa digunakan berkali-kali. Sifat id sangat unik.

Note: pelanggaran menggunakan id lebih dari 1 kali memang bisa dilakukan dan tidak menyebabkan error, karena tidak ada error pada HTML.

Data attribute adalah attribute html yang bisa dicustom sendiri dengan menuliskan kata “data-” di depannya, contohnya:

<div data-bahasa=”indonesia”></div>

<div data-id=”ABC”></div>

Data attribute sifatnya sama seperti attribute biasa pada html.

# 3 Selector Specificity

Setiap selector memiliki specificity level atau tingkat spesifikasi. Mengerti tingkat spesifikasi adalah hal yang paling penting dalam mempelajari css selector, karena tingkat spesifikasi adalah hal yang menentukan selector mana yang akan di prioritaskan atau akan diambil properties dan valuenya.

Ini adalah prinsip yang paling penting di pahami dalam selector! Semakin specific akan semakin kuat selector anda! Selector yang lebih kuat akan meng-override property dari selector yang lebih lemah. Contohnya:

Ada sebuah DOM

|  |
| --- |
| <div class=”kakek”>  <div class=”ayah”>  <p id=”choosen” class=”anak”>Test</p>  </div>  </div> |

Maka seseorang web designer atau programmer ingin memilih paragraph tersebut, maka beginilah kurang contoh lebih urutan dari yang paling lemah ke yang paling kuat

|  |
| --- |
| **p**  **p.anak**  **div.ayah p.anak**  **.kakek div.ayah p.anak**  **div.kakek > div.ayah > p.anak**  **#choosen**  **p#choosen**  **.ayah p#choosen** |

**Untuk lebih memahami specificity level, cobalah web yang ada pada folder Selector Trial.**

# 4 Cross-browser test

Sebagian properties css tidak akan bekerja di sebagian browser dan bekerja di browser lainnya, atau bekerja di sebagian versi tapi tidak bekerja di versi lainnya. Terkadang ada beberapa property yang memiliki prefix, artinya penulisannya harus berbeda untuk di setiap browser, misalnya:

border-radius: untuk umum,

-webkit-border-radius: untuk chrome atau safari versi lama,

-ms-border-radius: untuk ie versi lama,

-moz-border-radius: untuk fire fox versi lama.

Tapi sekarang perbedaan prefix semakin sedikit, dan rata-rata browser bisa menerima penulisa property yang umum tanpa perlu tambahan prefix. Selalu lakukan test untuk setiap browser untuk mengetahui lebih pasti.

# 5 Nama ID dan Class yang dimulai dengan angka

Pernah membuat sebuah nama id atau class yang dimulai dengan angka? Cobalah buat sebuah tag class yang dimulai dengan angka, misalnya seperti ini:

|  |
| --- |
| <p class=”3susunan”>paragraph</p>  <div id=”23containers”></div> |

Lalu cobalah style kedua tag ini dengan cara seperti ini:

|  |
| --- |
| .3susunan { color:blue }  #23containers { width: 200px; height:200px; background-color:red; } |

Ini tidak akan bekerja! Anda akan sangat bingung, dan merasa apa ada yang salah dengan cssnya, htmlnya atau css anda tidak berhasil di insert ke dalam document. Lalu mencoba memperbaiki ini dengan membuang-buang waktu seharian.

**CSS rule, selector tidak akan bekerja dengan sebuah nama id atau class yang diawali dengan character numeric**. Tapi bila numeric characternya berada di tengah atau di akhir, itu tidak akan jadi masalah. Biarpun HTML5 tidak masalah dengan penamaan class dan id diawali dengan number, tetapi CSS tidak.

*In CSS, identifiers (including element names, classes, and IDs in selectors) can contain only the characters [a-zA-Z0-9] and ISO 10646 characters U+00A0 and higher, plus the hyphen (-) and the underscore (\_); they cannot start with a digit, two hyphens, or a hyphen followed by a digit. Identifiers can also contain escaped characters and any ISO 10646 character as a numeric code (see next item). For instance, the identifier “B&W?” may be written as “B\&W\?” or “B\26 W\3F”.*

Untuk mengatasi hal ini, kalian harus menulis selector dengan cara seperti berikut ini:

|  |
| --- |
| [class=”3susunan”] { color:blue }  [id=”23containers”] { width: 200px; height:200px; background-color:red; } |

Ini sama saja dengan trick menamai class dan idnya lewat attribute selector. Di dalam attribute selector, semua nama dianggap sebagai string dan dipilih melalui html attributenya.

Catatan! Trick untuk menggunakan selector dengan attribute [] hanyalah untuk emergency case. Untuk naming convention yang baik, jangan pernah membuat nama class dan id yang diawali dengan angka! Ini bisa membingungkan programmer atau web designer!

# 6 Naming convetion.

Untuk back-end programmer, naming convention yang popular kalian pakai adalah camelCase dan PascalCase. Untuk front-end developer case yang cukup terkenal adalah BEM (Block, Element Modifier).

BEM menggunakan dash(-) ,underscore (\_) dan lower case untuk penamaan. Kegunaan naming convention ini adalah untuk membedakan jelas antara server-side attribute atau client-side attribute.

BEM juga digunakan agar developer bisa dengan mudah men-track susunan DOM yang dibuat.

Penamaan BEM seperti ini

|  |
| --- |
| Untuk pengganti space (spasi) digunakan single dash (-)  <div class=”global-container”>  Untuk sebuah element, dimana merupakan anak dari sebuah container(block) adalah double underscore (\_\_).  <div class=”global-container”>  <p class=”global-container\_\_tebal”></p>  <p></p>  </div>  Dan untuk modifier adalah double dash (--)  <div class=”global-container global-container--fancy”> |

BEM dimaksud kan untuk membuat sebanyak mungkin kombinasi antara box dan element, sehingga mencegah penamaan class yang sama. Prinsipnya sama seperti nama orang, dimana kombinasi dari nama depan dan nama keluarga, sehingga kemungkinan nama seseorang yang sama persis akan lebih kecil.

# 7 Single Named Component

Strategi penamaan single named component adalah dengan mengirit sebisa mungkin pemberian class apa lagi ID kepada html tag yang mau di style atau mau dimanipulasi oleh javascript.

Sehingga apabila front-end developer membuat sebuah widget dimana widget tersebut terdiri dari banyak sekali html node di dalamnya, front developer hanya perlu membuat sebuah nama class pada top container dari widgetnya saja.

Ini akan meringankan beban back-end developer bila ingin membuat widget yang sama dengan style yang sama tapi dalam jumlah yang relative banyak, sehingga back-end developer tidak perlu menulis ulang nama classnya berkali-kali.