

Scssで使える

@eachが

結構デキるヤツだった話

2016/12/02

@bozu\_is\_mine

Go Noji



# 自己紹介



※地球に優しい  
天然の地毛を使用しています

## 野地 剛

Comcent,inc エンジニア

フロント・バック間をうろうろしている  
エンジニア二年目

ドラマーだけどキーボード叩いています

Web や音楽、科学が中心のブログ

<http://noji.wpblog.jp/>

# 今回の主役

## @each

Scss に用意されている繰り返し用  
制御構文の一種

(他に @for や @while がある)

配列に対して繰り返し処理ができる

# こんな時、ありません？

各背景に写真を入れて、  
リストの2番目は枠線を緑に、  
6番目は青、7番目は赤。  
2番目と3番目は字を大きく……

```
<ul class="box">
  <li class="pic1"></li>
  <li class="pic2 greenBd bigTxt"></li>
  <li class="pic3 bigTxt"></li>
  <li class="pic4"></li>
  <li class="pic5"></li>
  <li class="pic6 blueBorder"></li>
  <li class="pic7 redBorder"></li>
</ul><!-- 管理しづらいですよね……-->
```

で、私の出番です

@each

```
$transparent: transparent;
```

```
$green: #388E3C;
```

```
$blue: #1976D2;
```

```
$red: #d32f2f;
```

```
$fontSize: 12px;
```

```
$bigFontSize: $fontSize + 2px;
```

```
$pics: 'pic1', 'pic2', 'pic3', 'pic4' (略);
```

```
$colors: $transparent, $green(略);
```

```
$fontSizes: $fontSize, $bigFontSize(略);
```

```
.box{
  li{
    @each $pic in $pics{
      $index: index($pics, $pic);
      &:nth-child(#{ $index }) {
        background-image: url(#{ $pic }.jpg);
        border: nth($colors, $index) solid 1px;
        font-size: nth($fontSizes, $index);
      }
    }
  }
}
```



```
.box{
```

```
  li{
```

```
    @each $pic in $pics{
```

```
      $index: index($pics, $pic);  
      &:nth-child(#{ $index });  
      追記する。
```

```
      background-image: url(#{ $pic }.jpg);  
      border: nth($colors, $index) solid 1px;  
      font-size: nth($fontSizes, $index);  
    }
```

```
  }
```

```
}
```

```
}
```

```
.box{  
  li{  
    @each $pic in $pics{
```

```
      $index: index($pics, $pic);
```

```
      &:nth-child(#{ $index }){  
        index( 要素の含まれる配列, 要素の値 );
```

```
        background-image: url(#{ $pic }.jpg);  
        は要素が配列の何番目かを返す。
```

```
        border: nth($colors, $index) solid 1px;
```

```
        これで現在何周目のループか参照できる。  
        font-size: nth($fontSizes, $index);
```

```
      }
```

```
    }
```

```
  }
```

```
}
```

```
.box{
```

```
  li{
```

```
    @each $pic in $pics{
```

先ほど得たインデックスを nth-child に適用。

```
      &:nth-child(#{ $index }){
```

しかしそのままの \$index は値の部分にしか;

使えないので、 nth-interpolation(#{ \$index } 1px;

で補完する。 nth(\$fontSizes, \$index);

```
    }
```

```
  }
```

```
}
```

```
}
```

```
.box{
  li{
    @each $pic in $pics{
      $index: index($pics, $pic);
      &:nth-child(#{ $index }){
        background-image: url(#{ $pic }.jpg);
        border: nth($colors, $index) solid 1px;
        font-size: nth($fontSizes, $index);
        url('pic1'.jpg);
      }
    }
  }
}
```

この部分も普通に書くと  
url('pic1'.jpg);  
のように「'」がついてしまうので  
インターポレーション。

```
.box{
  li{
    @each $pic in $pics{
      $index: index($pics, $pic);
      &:nth-child(#{ $index }) {
        他の配列を参照するため nth 関数を使用。;
        background-image: url(#{ $pic }.jpg);
        border: nth($colors, $index) solid 1px;
        font-size: nth($fontSizes, $index);
      }
    }
  }
}
```

nth(参照する配列, 取得したいインデックス);  
で他配列内の要素を取得できる。

// コンパイル後の css

```
.box li:nth-child(1) {  
    background-image: url(pic1.jpg);  
    border: transparent solid 1px;  
    font-size: 12px;  
}
```

```
.box li:nth-child(2) {  
    background-image: url(pic2.jpg);  
    border: #388E3C solid 1px;  
    font-size: 14px;  
}
```

(以下略)

# 実際にこんなモノを作ってみた

```
<ol class="periodic">  
  <li> 水素 </li>  
  <li> ヘリウム </li>  
  <li> リチウム </li>  
    ⋮  
</ol>
```

このように元素名が並んでいる  
HTMLにScssでスタイルを当て、  
周期表を作ってみる。







```
2985 ▶ .periodic li:nth-child(110):after { ... }
2997 ▶ .periodic li:nth-child(111) { ... }
3005 ▶ .periodic li:nth-child(111):before { ... }
3011 ▶ .periodic li:nth-child(111):after { ... }
3023 ▶ .periodic li:nth-child(112) { ... }
3031 ▶ .periodic li:nth-child(112):before { ... }
3037 ▶ .periodic li:nth-child(112):after { ... }
3049 ▶ .periodic li:nth-child(113) { ... }
3057 ▶ .periodic li:nth-child(113):before { ... }
3063 ▶ .periodic li:nth-child(113):after { ... }
3075 ▶ .periodic li:nth-child(114) { ... }
3083 ▶ .periodic li:nth-child(114):before { ... }
3089 ▶ .periodic li:nth-child(114):after { ... }
3101 ▶ .periodic li:nth-child(115) { ... }
3109 ▶ .periodic li:nth-child(115):before { ... }
3115 ▶ .periodic li:nth-child(115):after { ... }
3127 ▶ .periodic li:nth-child(116) { ... }
3135 ▶ .periodic li:nth-child(116):before { ... }
3141 ▶ .periodic li:nth-child(116):after { ... }
3153 ▶ .periodic li:nth-child(117) { ... }
3161 ▶ .periodic li:nth-child(117):before { ... }
3167 ▶ .periodic li:nth-child(117):after { ... }
3179 ▼ .periodic li:nth-child(118) {
3180     left: 94.44444%;
3181     margin-top: 33.33333%;
3182     color: #4527A0;
3183     border-bottom: #FFA000 solid 6px;
3184     border-left: transparent solid 15px;
3185     border-right: transparent solid 15px;
3186 }
3187 ▼ .periodic li:nth-child(118):before {
3188     content: "118";
3189     display: block;
3190     margin-top: 30%;
3191     font-size: 12px;
3192 }
3193 ▼ .periodic li:nth-child(118):after {
3194     content: "Uuo";
3195     display: block;
3196     width: 90%;
3197     height: 0;
3198     padding-bottom: 40%;
3199     position: absolute;
3200     margin-top: 30%;
3201     left: 5%;
3202     font-size: 24px;
3203     font-weight: 700;
3204 }
```

できました！

※HTMLはただの <ol> & <li> です

1 水素 <b>H</b>																	2 ヘリウム <b>He</b>														
3 リチウム <b>Li</b>	4 ベリリウム <b>Be</b>											5 ホウ素 <b>B</b>	6 炭素 <b>C</b>	7 窒素 <b>N</b>	8 酸素 <b>O</b>	9 フッ素 <b>F</b>	10 ネオン <b>Ne</b>														
11 ナトリウム <b>Na</b>	12 マグネシウム <b>Mg</b>											13 アルミニウム <b>Al</b>	14 ケイ素 <b>Si</b>	15 リン <b>P</b>	16 硫黄 <b>S</b>	17 塩素 <b>Cl</b>	18 アルゴン <b>Ar</b>														
19 カリウム <b>K</b>	20 カルシウム <b>Ca</b>	21 スカンジウム <b>Sc</b>	22 チタン <b>Ti</b>	23 バナジウム <b>V</b>	24 クロム <b>Cr</b>	25 マンガン <b>Mn</b>	26 鉄 <b>Fe</b>	27 コバルト <b>Co</b>	28 ニッケル <b>Ni</b>	29 銅 <b>Cu</b>	30 亜鉛 <b>Zn</b>	31 ガリウム <b>Ga</b>	32 ゲルマニウム <b>Ge</b>	33 ヒ素 <b>As</b>	34 セレン <b>Se</b>	35 臭素 <b>Br</b>	36 クリプトン <b>Kr</b>														
37 ルビジウム <b>Rb</b>	38 ストロンチウム <b>Sr</b>	39 イットリウム <b>Y</b>	40 ジルコニウム <b>Zr</b>	41 ニオブ <b>Nb</b>	42 モリブデン <b>Mo</b>	43 テクネチウム <b>Tc</b>	44 ルルチウム <b>Ru</b>	45 ロジウム <b>Rh</b>	46 パラジウム <b>Pd</b>	47 銀 <b>Ag</b>	48 カドミウム <b>Cd</b>	49 インジウム <b>In</b>	50 スズ <b>Sn</b>	51 アンチモン <b>Sb</b>	52 テルル <b>Te</b>	53 ヨウ素 <b>I</b>	54 キセノン <b>Xe</b>														
55 セシウム <b>Cs</b>	56 バリウム <b>Ba</b>	57 ランタン <b>La</b>	72 ハフニウム <b>Hf</b>	73 タンタル <b>Ta</b>	74 タングステン <b>W</b>	75 レニウム <b>Re</b>	76 オスミウム <b>Os</b>	77 イリジウム <b>Ir</b>	78 白金 <b>Pt</b>	79 金 <b>Au</b>	80 水銀 <b>Hg</b>	81 タリウム <b>Tl</b>	82 鉛 <b>Pb</b>	83 ビスマス <b>Bi</b>	84 ポロニウム <b>Po</b>	85 アスタチン <b>At</b>	86 ラドン <b>Rn</b>														
87 フランシウム <b>Fr</b>	88 ラジウム <b>Ra</b>	89 アクチニウム <b>Ac</b>	104 ラザホージウム <b>Rf</b>	105 ドブニウム <b>Db</b>	106 シーボーギウム <b>Sg</b>	107 ボーリウム <b>Bh</b>	108 ハッシウム <b>Hs</b>	109 マイトネリウム <b>Mt</b>	110 ダームスタチウム <b>Ds</b>	111 レントゲニウム <b>Rg</b>	112 コペルニシウム <b>Cn</b>	113 ウンウントリウム <b>Uut</b>	114 フレロビウム <b>Fl</b>	115 ウンウンペンチウム <b>Uup</b>	116 リバモリウム <b>Lv</b>	117 ウンウンセプチウム <b>Uus</b>	118 ウンウンオクテチウム <b>Uuo</b>														
																		58 セリウム <b>Ce</b>	59 プラセオジム <b>Pr</b>	60 ネオジム <b>Nd</b>	61 プロメチウム <b>Pm</b>	62 サマリウム <b>Sm</b>	63 ユウロピウム <b>Eu</b>	64 ガドリニウム <b>Gd</b>	65 テルビウム <b>Tb</b>	66 ジスプロシウム <b>Dy</b>	67 ホルミウム <b>Ho</b>	68 エルビウム <b>Er</b>	69 ツリウム <b>Tm</b>	70 イットルビウム <b>Yb</b>	71 ルルチウム <b>Lu</b>
																		90 トリウム <b>Th</b>	91 プロトアクチニウム <b>Pa</b>	92 ウラン <b>U</b>	93 ネプツニウム <b>Np</b>	94 プルトニウム <b>Pu</b>	95 アメリシウム <b>Am</b>	96 キュリウム <b>Cm</b>	97 バークリウム <b>Bk</b>	98 カリホルニウム <b>Cf</b>	99 アインスタイニウム <b>Es</b>	100 フェルミウム <b>Fm</b>	101 メンデレビウム <b>Md</b>	102 ノーベリウム <b>No</b>	103 ローレンシウム <b>Lr</b>

# まとめ

## @each は結構デキるヤツ

- 複雑なリストがある静的ページに最適
- 一度書いてしまえば修正が楽
- コンパイルが成功すると気持ちいい

# ありがとうございました！

今回の周期表のコード & Web ページ

[https://github.com/Go-Noji/Scss-each\\_periodicTable](https://github.com/Go-Noji/Scss-each_periodicTable)

[https://go-noji.github.io/Scss-each\\_periodicTable/](https://go-noji.github.io/Scss-each_periodicTable/)

お蔵入り each デモコード & Web ページ

[https://github.com/Go-Noji/Scss-each\\_aPsedou](https://github.com/Go-Noji/Scss-each_aPsedou)

[https://go-noji.github.io/Scss-each\\_aPsedou/](https://go-noji.github.io/Scss-each_aPsedou/)

[https://github.com/Go-Noji/Scss-each\\_classImg](https://github.com/Go-Noji/Scss-each_classImg)

[https://go-noji.github.io/Scss-each\\_classImg/](https://go-noji.github.io/Scss-each_classImg/)

[https://github.com/Go-Noji/Scss-each\\_pref](https://github.com/Go-Noji/Scss-each_pref)

[https://go-noji.github.io/Scss-each\\_pref/](https://go-noji.github.io/Scss-each_pref/)