

CSSnITE

LPO2

Sass

2014.2.15

BASIC

Sass

CSS がもっとラクに書ける!

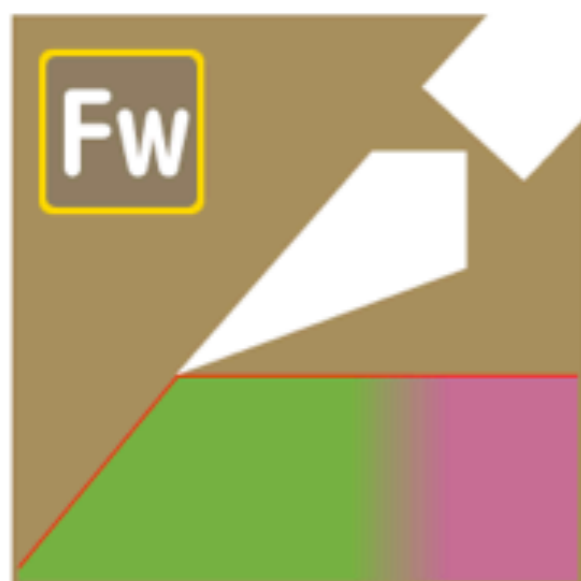
これから始める Sass の書き方

アンコピ 柴田 大樹



Fireworksにはスライスレイヤーに色をつけられる

2013年12月14日 · Fireworksの使い方 · comments(0)



まだ語られていない
とっておきの
スライス機能があるんだ…!

先日はFireworks以外のツールでのスライスを書きましたが、いかがだったでしょうか。「この背徳者！」と言われず、内心ホッとしております。

さて、今日はFireworksにしかできないところで地味なスライス機能を紹介します。

続きを読む

ダウンロード



+

フルバージョン

Pocket 20

Bookmark 1

Tweet 16

いいね!

27人が「いいね!」と言っています。「いいね!」をクリックして、友達に知らせましょう。



SassのことでCSS Niteでお話します

人気のエントリー

- [ネットで見られる提案書のまとめ](#)
- [Windowsのかゆいところに手が届く！便利なフリーソフト13選](#)
- [制作中の面倒を解決するFireworksの拡張機能のまとめ \(2011年7月版\)](#)
- [Dreamweaverのデザインビューを使い倒すために覚えてきたいこと](#)
- [PhotoshopやIllustratorを組み合わせると5分でも早く！Fireworks中心に制作している僕のAdobeソフトの使い方](#)

カテゴリー

- [Webデザイン](#) (117)
- [Fireworksの使い方](#) (37)
- [Dreamweaverの使い方](#) (18)
- [ランディングページ](#) (5)
- [twitter](#) (1)

2012



なぜ？

Sass

Sass

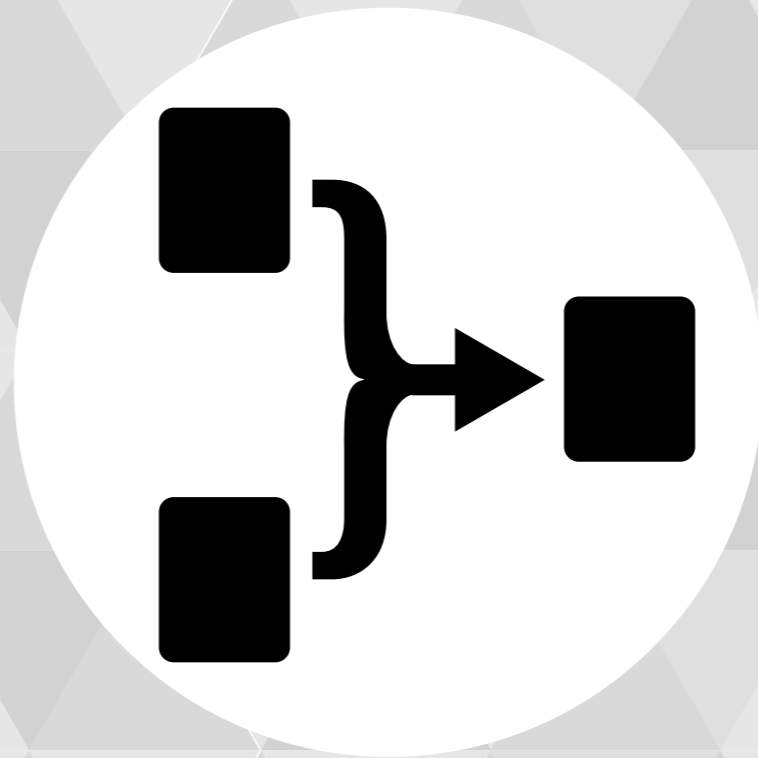
なら

ラクにCSSが書ける

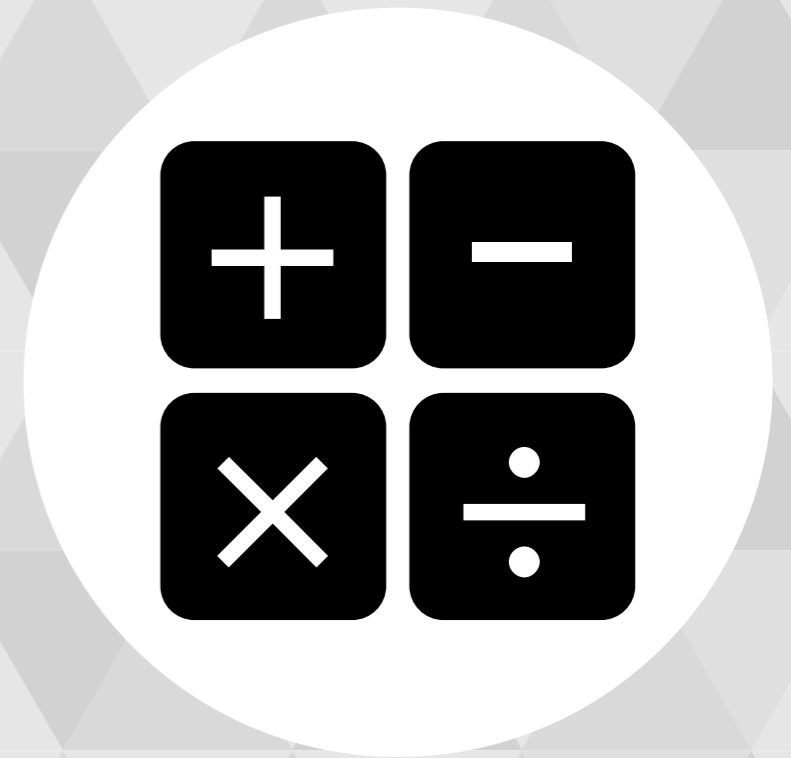
今日のお話



入力の
手間省く



CSSを
まとめる



計算
できる

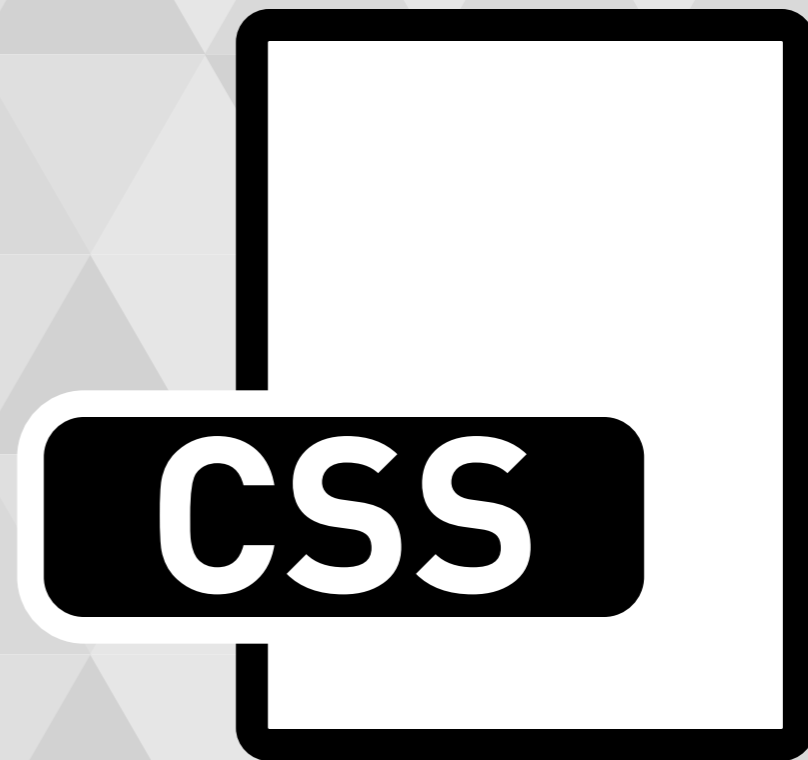




おさらしい



`style.scss`



`style.css`





PSD

を

し、

PSD

JPEG

書き出し

JPEGに

する



PSD

を

変換

し、

PSD

JPEG

書き出し

JPEGに

する



PSD

を

変換

し、

PSD

G

書き

JPEG

に

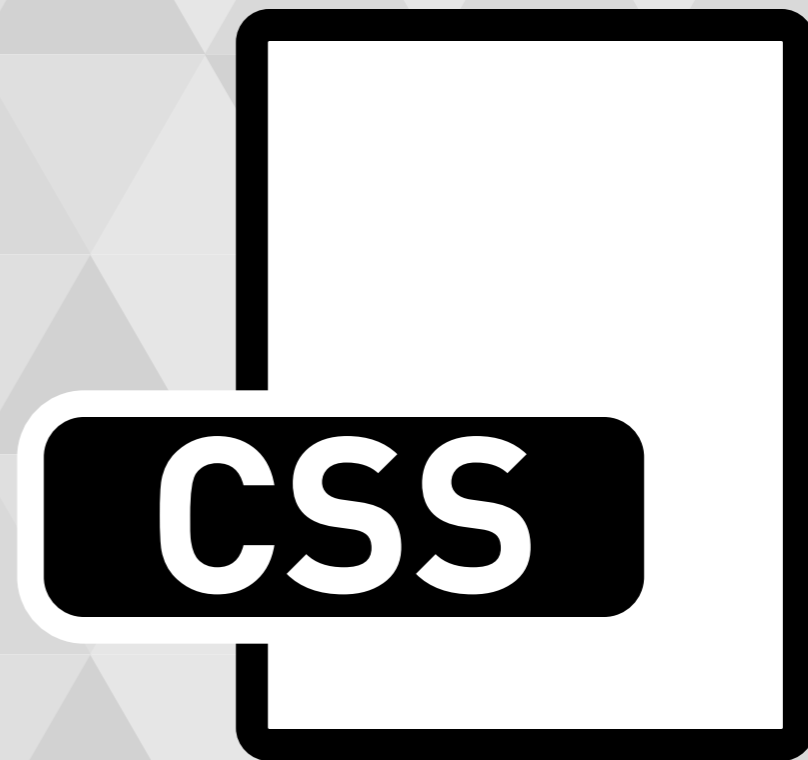
出力

する





`style.scss`



`style.css`



変換
コンパイル

Sass を

Sass

style.scss

し、

CSS

style.css

CSS

変換
に
コンパイル

する

Sass

を

変換

し、

Sass

style.scss

style.css

CSS

変換
に
コンパイル

する

Sass

を

変換

し、

Sass

style.scss

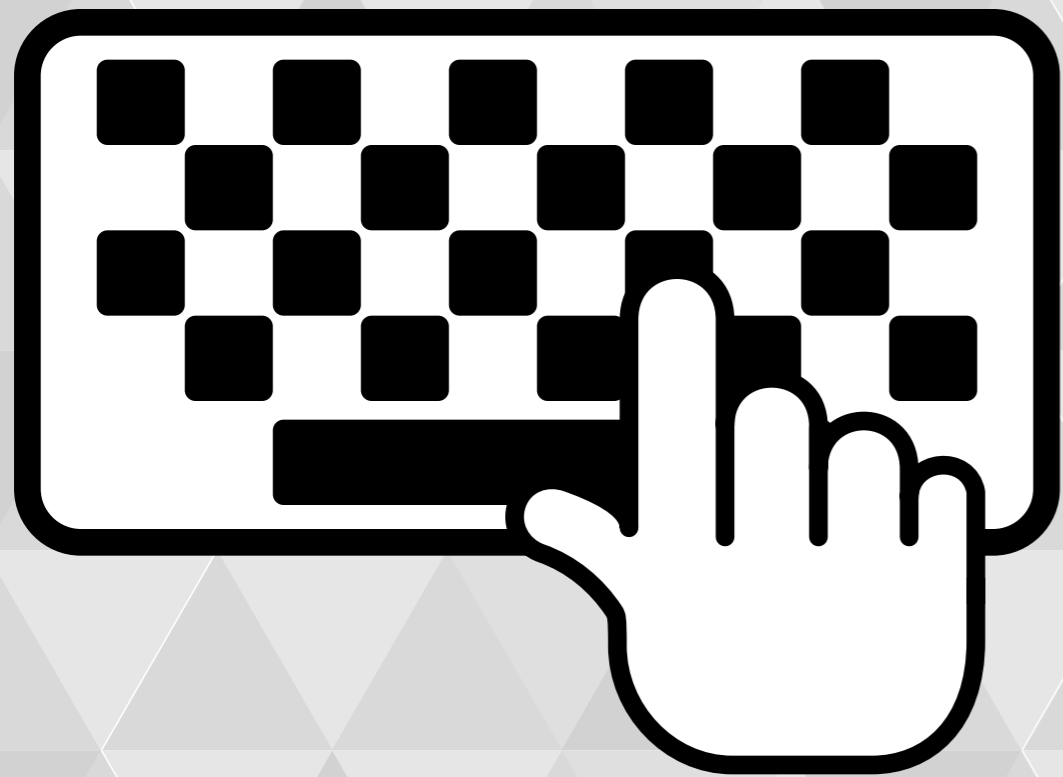
style.css

CSS

に

出力

する



入力の手間を
省略





+



Copy & Paste

$$\mathbf{C} + \mathbf{V}$$

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ



- **共通のセレクタ**
- **同じ値**
- **同じプロパティと値**
- **セレクタのグループ**

ここが大変CSS

共通セレクタを
1つ1つ書くのが大変

従来のCSS

```
#main h1 {  
  margin: 10px;  
}  
#main h2 {  
  margin: 10px;  
}  
#main p {  
  margin: 10px;  
}  
#main ul {  
  margin: 10px;  
}
```

従来のCSS

```
#main h1 {  
  margin: 10px;  
}  
#main h2 {  
  margin: 10px;  
}  
#main p {  
  margin: 10px;  
}  
#main ul {  
  margin: 10px;  
}
```

Sassにすると

```
#main {  
  h1 {  
    margin: 10px;  
  }  
  h2 {  
    margin: 10px;  
  }  
  p {  
    margin: 10px;  
  }  
  ul {  
    margin: 10px;  
  }  
}
```

Sass Meisterで見る

Sassにすると

```
#main {  
  h1 {  
    margin: 10px;  
  }  
  h2 {  
    margin: 10px;  
  }  
  p {  
    margin: 10px;  
  }  
  ul {  
    margin: 10px;  
  }  
}
```

Sass Meisterで見る

Sassにすると

```
#main {  
  h1 {  
    margin: 10px;  
  }  
  h2 {  
    margin: 10px;  
  }  
  p {  
    margin: 10px;  
  }  
  ul {  
    margin: 10px;  
  }  
}
```



```
#main h1 {  
  margin: 10px;  
}  
#main h2 {  
  margin: 10px;  
}  
#main p {  
  margin: 10px;  
}  
#main ul {  
  margin: 10px;  
}
```

Sass Meisterで見る

Sassにすると

```
#main {  
  h1 {  
    margin: 10px;  
  }  
  h2 {  
    margin: 10px;  
  }  
  p {  
    margin: 10px;  
  }  
  ul {  
    margin: 10px;  
  }  
}
```



```
#main h1 {  
  margin: 10px;  
}  
#main h2 {  
  margin: 10px;  
}  
#main p {  
  margin: 10px;  
}  
#main ul {  
  margin: 10px;  
}
```

Sass Meisterで見る

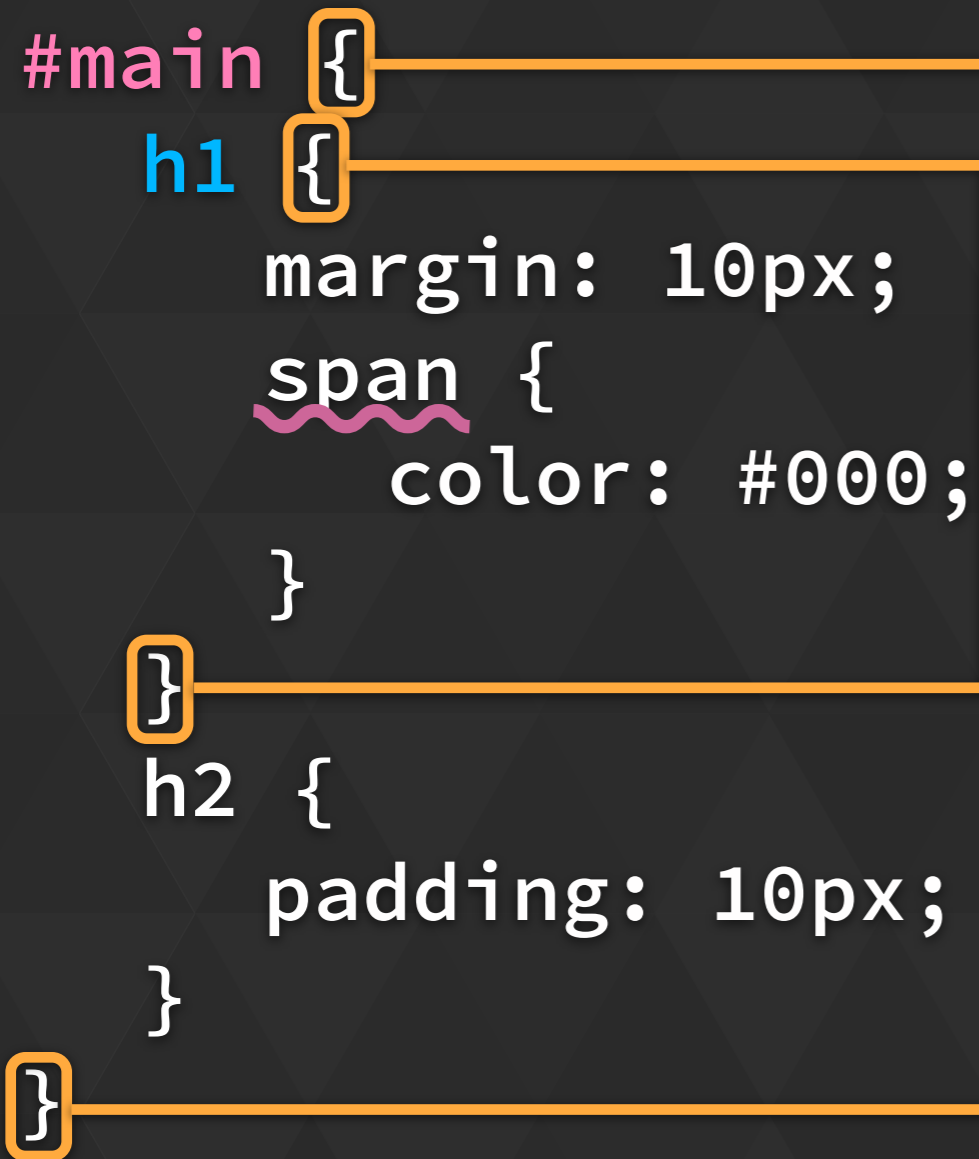
2重でも3重でもネスト（入れ子）OK

```
#main {  
  h1 {  
    margin: 10px;  
    span {  
      color: #000;  
    }  
  }  
  h2 {  
    padding: 10px;  
  }  
}
```

Sass Meisterで見る

2重でも3重でもネスト（入れ子）OK

```
#main {  
  h1 {  
    margin: 10px;  
    span {  
      color: #000;  
    }  
  }  
  h2 {  
    padding: 10px;  
  }  
}
```



Sass Meisterで見る

2重でも3重でもネスト（入れ子）OK

```
#main {  
  h1 {  
    margin: 10px;  
    span {  
      color: #000;  
    }  
  }  
  h2 {  
    padding: 10px;  
  }  
}
```

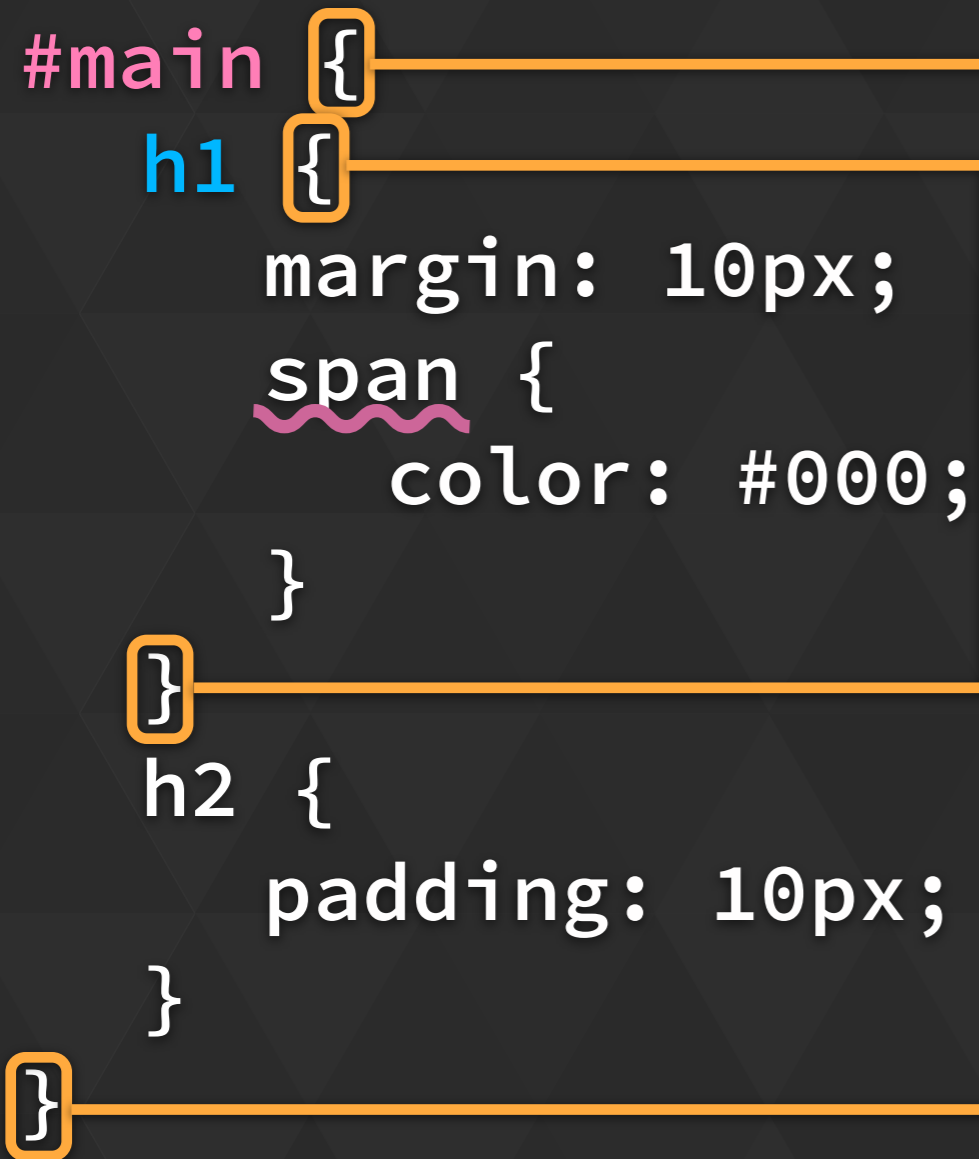


```
#main h1 {  
  margin: 10px;  
}  
#main h1 span {  
  color: #000;  
}  
#main h2 {  
  padding: 10px;  
}
```

Sass Meisterで見る

2重でも3重でもネスト（入れ子）OK

```
#main {  
  h1 {  
    margin: 10px;  
    span {  
      color: #000;  
    }  
  }  
  h2 {  
    padding: 10px;  
  }  
}
```



```
#main h1 {  
  margin: 10px;  
}  
#main h1 span {  
  color: #000;  
}  
#main h2 {  
  padding: 10px;  
}
```

Sass Meisterで見る

$$\mathbf{C} + \mathbf{V}$$

- **共通のセレクタ**
- **同じ値**
- **同じプロパティと値**
- **セレクタのグループ**

$$\mathbf{C} + \mathbf{V}$$

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ

ここが大変CSS

同じ値を書くのが大変

1dez

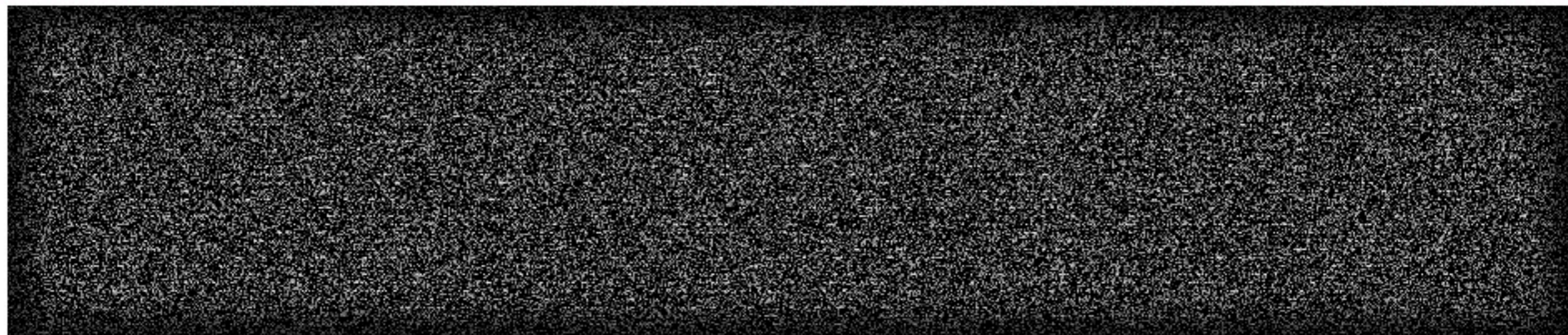
HOME

無料体験キット

受講価格一覧

ワンデズとは？

卒業生・在校生の声



無料体験
キット
全員プレゼント



全講座
登録OK

78×78

Webデザイン
専門講座

78×78

スクーリング
講座

78×78

BASIC COURSE

講座リスト

- ▶ [Webデザイン基礎講座](#)
- ▶ [IA基礎講座](#)
- ▶ [Fireworks基礎講座](#)
- ▶ [Photoshop基礎講座](#)
- ▶ [Illustrator基礎講座](#)
- ▶ [シルバーパソコン基礎講座](#)

安心サポート体制

- ▶ 専門学校の通信講座だから安心！

150×115

Webデザイン基礎講座

Webデザイン基礎と練習方法を中心にWebデザイナーになるための土台を目指します。

150×115

IA基礎講座

IAとは何か、IEとの違いを理解し、将来のIAとしておのリーダーシップを磨きます。

150×115

サークルホームページ講座

大学や社会人のサークルの

150×115

シルバーパソコン講座

これからの情報社会を生き

1dez

HOME

無料体験キット

受講価格一覧

ワンデズとは？

卒業生・在校生の声



無料体験
キット
全員プレゼント



全講座
登録OK

78 x 78

Webデザイン
専門講座

78 x 78

スクーリング
講座

78 x 78

BASIC COURSE

講座リスト

[ドッグトリマー基礎講座](#)

[ドッグトレーナー基礎講座](#)

[動物看護師養成専門講座](#)

[ドッグトリマー養成専門講座](#)

[ドッグトレーナー養成専門講座](#)

[ドッグブリーダー養成専門講座](#)

安心サポート体制

専門学校の通信講座だから安心！

150 x 115

Webデザイン基礎講座

Webデザイン基礎と練習方法を中心にWebデザイナーになるための土台を目指します。

150 x 115

IA基礎講座

IAとは何か、IEとの違いを理解し、将来のIAとしてぶのリーダーシップを磨きます。

150 x 115

サークルホームページ講座

大学や社会人のサークルのホームページに担当者に必要な知識と技術・調整力を習得します。

150 x 115

シルバーパソコン講座

これからの情報社会を生き抜くためのスキルを体系的にマスターできます。

1dez

HOME

#AD253A

一覧

ワンデズとは?

卒業生・在校生の声

#AD253A

無料体験
キット
全員プレゼント

全講座
登録OK

78 x 78

Webデザイン
専

スクーリング

78 x 78

#AD253A

BASIC COURSE

講座リスト

- [ドッグトリマー基礎講座](#)
- [ドッグトレーナー基礎講座](#)
- [動物看護師養成専門講座](#)
- [ドッグトリマー養成専門講座](#)
- [ドッグトレーナー養成専門講座](#)
- [ドッグブリーダー養成専門講座](#)

安心サポート体制

専門学校の通信講座だから安心!

Webデザイン基礎講座

150 x 115

Webデザイン基礎と練習方法を中心にWebデザイナーになるための土台を目指します。

IA基礎講座

150 x 115

IAとは何か、IEとの違いを理解し、将来のIAとしてのリーダーシップを磨きます。

シルバーパソコン講座

150 x 115

これからの情報社会を生き抜くためのスキルを体系的にマスターできます。

#AD253A

1dez

HOME

#AD253A

一覧

ワンデズと

全体を青に
できませんか？

ドッグトレーナー基礎講座

動物看護師養成専門講座

ドッグトリマー養成専門講座

ドッグトレーナー養成専門講座

ドッグブリーダー養成専門講座

150 x 115

We

#A

安心サポート体制

専門学校通信講座だから安心！



```
525 #main #featureList {
526     letter-spacing: -.4em;
527 }
528 #main #featureList li {
529     width: 222px;
530     display: inline-block;
531     letter-spacing: normal;
532     margin-right: 12px;
533 }
534 #main #featureList li:last-child {
535     margin-right: 0;
536 }
537 #main #featureList a {
538     -webkit-transition: all 0.1s ease-out;
539     -moz-transition: all 0.1s ease-out;
540     transition: all 0.1s ease-out;
541     display: table;
542     width: 214px;
543     border: 4px solid #333333;
544     background-color: #fff;
545     color: #ad253a;
546     text-decoration: none;
547 }
548 #main #featureList a:hover {
549     border: 4px solid #666666;
550     box-shadow: 0 0 15px #b22d00;
```



変数

#FF3300

10px

"Futura"



変数の代入と参照

```
$key_color: #DF5496;
```

```
.box {  
  background: $key_color;  
  p {  
    color: $key_color;  
    background: #FFF;  
  }  
}
```

変数の代入と参照

```
$key_color: #DF5496;
```

```
.box {  
  background: $key_color;  
  p {  
    color: $key_color;  
    background: #FFF;  
  }  
}
```

変数の代入と参照

#DF5496

```
$key_color: #DF5496;
```

```
.box {  
  background-color: $key_color;  
  p {  
    color: #FFF;  
    background-color: $key_color;  
  }  
}
```

\$key_color

変数の代入と参照

```
$key_color: #DF5496;
```

```
.box {  
  background: $key_color;  
  p {  
    color: $key_color;  
    background: #FFF;  
  }  
}
```


変数の代入と参照

```
$key_color: #DF5496;
```

変数を代入

\$変数名 : 値;

```
.box {  
  background: $key_color;  
  p {  
    color: $key_color;  
    background: #FFF;  
  }  
}
```

変数の代入と参照

`$key_color`: #DF5496;

変数を代入

\$変数名 : 値;

```
.box {  
  background: $key_color;  
  p {  
    color: $key_color;  
    background: #FFF;  
  }  
}
```

変数の代入と参照

`$key_color`: #DF5496;

変数を代入

`$変数名` : 値;

```
.box {  
  background: #DF5496 ;  
  p {  
    color: #DF5496 ;  
    background: #FFF;  
  }  
}
```

変数の代入と参照

```
$key_color: #DF5496;  
  
.box {  
  background: $key_color;  
  p {  
    color: $key_color;  
    background: #FFF;  
  }  
}
```

Sass Meisterで見る

変数の代入と参照

```
$key_color: #DF5496;

.box {
  background: $key_color;
  p {
    color: $key_color;
    background: #FFF;
  }
}
```



```
.box {
  background: #DF5496;
}
.box p {
  color: #DF5496;
  background: #FFF;
}
```

Sass Meisterで見る

#FF3300

10px

"Futura"



もちろん、数値や文字列もOK

```
$ff_Impact: Impact, Arial;
```

```
$m_value: 10px;
```

```
body {
```

```
  font-family: $ff_Impact;
```

```
}
```

```
input, textarea {
```

```
  font-family: $ff_Impact;
```

```
}
```

```
.box1 {
```

```
  margin: $m_value;
```

```
}
```

Sass Meisterで見る

もちろん、数値や文字列もOK

```
$ff_Impact: Impact, Arial;
$m_value: 10px;

body {
  font-family: $ff_Impact;
}
input, textarea {
  font-family: $ff_Impact;
}
.box1 {
  margin: $m_value;
}
```



```
body {
  font-family:
  Impact, Arial;
}

input, textarea {
  font-family:
  Impact, Arial;
}

.box1 {
  margin: 10px;
}
```

Sass Meisterで見る

1dez

HOME

#AD253A

一覧

ワンデズと

全体を青に
できませんか？

ドッグトレーナー基礎講座

動物看護師養成専門講座

ドッグトリマー養成専門講座

ドッグトレーナー養成専門講座

ドッグブリーダー養成専門講座

150 x 115

We

#A

安心サポート体制

専門学校通信講座だから安心！



1dez

HOME

#AD253A

一覧

ワンデズとは?

卒業生・在校生の声

#AD253A

無料体験
キット
全員プレゼント

全講座
登録OK

78 x 78

Webデザイン
専

スクーリング

78 x 78

#AD253A

BASIC COURSE

講座リスト

- [ドッグトリマー基礎講座](#)
- [ドッグトレーナー基礎講座](#)
- [動物看護師養成専門講座](#)
- [ドッグトリマー養成専門講座](#)
- [ドッグトレーナー養成専門講座](#)
- [ドッグブリーダー養成専門講座](#)

安心サポート体制

専門学校の通信講座だから安心!

Webデザイン基礎講座

150 x 115

Webデザイン基礎と練習方法を中心にWebデザイナーになるための土台を目指します。

IA基礎講座

150 x 115

IAとは何か、IEとの違いを理解し、将来のIAとしてのリーダーシップを磨きます。

#AD253A

シルバーパソコン講座

150 x 115

これからの情報社会を生き抜くためのスキルを体系的にマスターできます。

```
$color: #AD253A;  
body {  
  background: $color;  
}
```

```
#gNav {  
  color: $color;  
}
```

```
.ttl {  
  color: $color;  
}
```

```
#side {  
  background: $color;  
}
```

```
$color: #AD253A;  
body {  
  background: $color;  
}
```

```
#gNav {  
  color: $color;  
}
```

```
.ttl {  
  color: $color;  
}
```

```
#side {  
  background: $color;  
}
```



```
body {  
  background: #AD253A;  
}
```

```
#gNav {  
  color: #AD253A;  
}
```

```
.ttl {  
  color: #AD253A;  
}
```

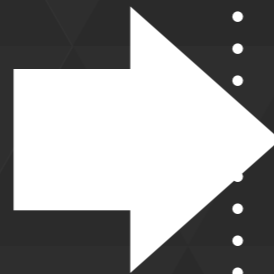
```
#side {  
  background: #AD253A;  
}
```

```
$color: #008ADE;  
body {  
  background: $color;  
}
```

```
#gNav {  
  color: $color;  
}
```

```
.ttl {  
  color: $color;  
}
```

```
#side {  
  background: $color;  
}
```



```
body {  
  background: #AD253A;  
}
```

```
#gNav {  
  color: #AD253A;  
}
```

```
.ttl {  
  color: #AD253A;  
}
```

```
#side {  
  background: #AD253A;  
}
```

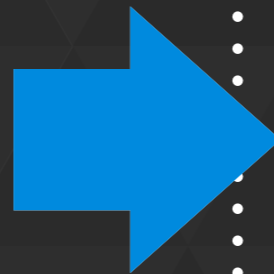


```
$color: #008ADE;  
body {  
  background: $color;  
}
```

```
#gNav {  
  color: $color;  
}
```

```
.ttl {  
  color: $color;  
}
```

```
#side {  
  background: $color;  
}
```



```
body {  
  background: #008ADE;  
}
```

```
#gNav {  
  color: #008ADE;  
}
```

```
.ttl {  
  color: #008ADE;  
}
```

```
#side {  
  background: #008ADE;  
}
```


1dez

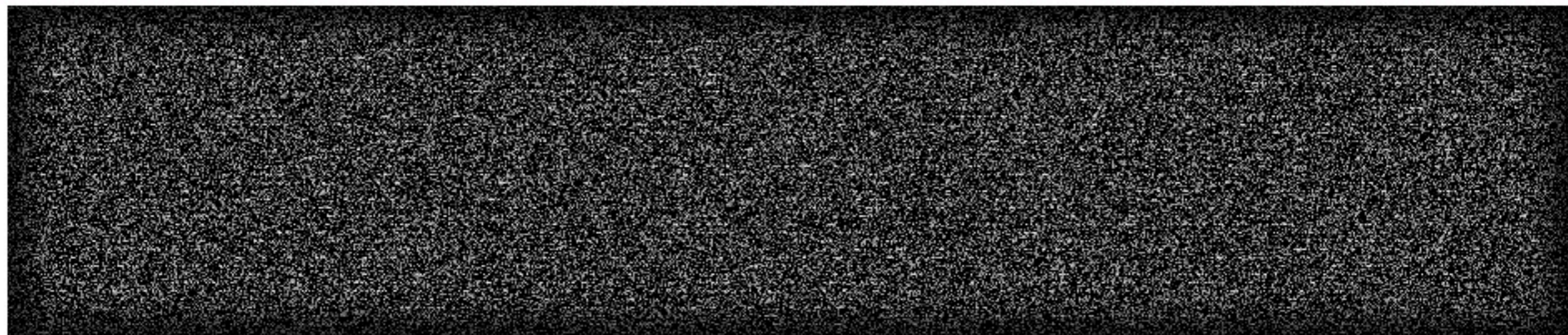
HOME

無料体験キット

受講価格一覧

ワンデズとは？

卒業生・在校生の声



無料体験
キット
全員プレゼント



全講座
登録OK

78×78

Webデザイン
専門講座

78×78

スクーリング
講座

78×78

BASIC COURSE

講座リスト

- ▶ [Webデザイン基礎講座](#)
- ▶ [IA基礎講座](#)
- ▶ [Fireworks基礎講座](#)
- ▶ [Photoshop基礎講座](#)
- ▶ [Illustrator基礎講座](#)
- ▶ [シルバーパソコン基礎講座](#)

安心サポート体制

- ▶ 専門学校通信講座だから安心！

150×115

Webデザイン基礎講座

Webデザイン基礎と練習方法を中心にWebデザイナーになるための土台を目指します。

150×115

IA基礎講座

IAとは何か、IEとの違いを理解し、将来のIAとしておのリーダーシップを磨きます。

150×115

サークルホームページ講座

大学や社会人のサークルの

150×115

シルバーパソコン講座

これからの情報社会を生き

1dez

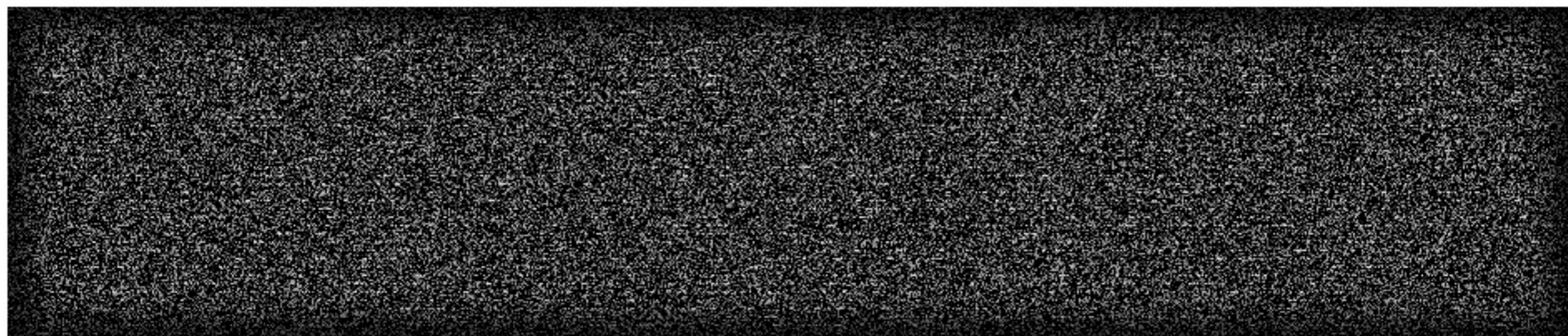
HOME

無料体験キット

受講価格一覧

ワンデズとは？

卒業生・在校生の声



無料体験
キット
全員プレゼント



全講座
登録OK

78×78

Webデザイン
専門講座

78×78

スクーリング
講座

78×78

BASIC COURSE

講座リスト

- ▶ [Webデザイン基礎講座](#)
- ▶ [IA基礎講座](#)
- ▶ [Fireworks基礎講座](#)
- ▶ [Photoshop基礎講座](#)
- ▶ [Illustrator基礎講座](#)
- ▶ [シルバーパソコン基礎講座](#)

安心サポート体制

- ▶ 専門学校通信講座だから安心！

150×115

Webデザイン基礎講座

Webデザイン基礎と練習方法を中心にWebデザイナーになるための土台を目指します。

150×115

IA基礎講座

IAとは何か、IEとの違いを理解し、将来のIAとしておのリーダーシップを磨きます。

150×115

サークルホームページ講座

大学や社会人のサークルの

150×115

シルバーパソコン講座

これからの情報社会を生き

変数の使いどころ

- サイト全体で共通して使う色
 - メインカラー、サブカラー、アクセントカラー
 - 通常のリンクカラー・ホバー時のリンクカラー
- ページ幅
- フォント名
 - bodyにfont-familyを指定しても、フォームには改めて指定する必要があるので変数にしておくと便利です。
- プロパティの値が他のプロパティにも影響するとき
 - 計算する際、同じ値を使うときに変数にしておくとし1箇所を編集するとCSSを書き換えられるので便利です。セッションでお話した960pxのボックスをpaddingやborderの値を引いて3分割する...といったときに使うと便利です。

$$\mathbf{C} + \mathbf{V}$$

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ

$$\mathbf{C} + \mathbf{V}$$

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ

ここが大変CSS

同じプロパティと値を
書くのが大変


```
.boxA p {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA ul {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
  line-height: 1.3;  
}
```

```
.boxA p {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA ul {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
  line-height: 1.3;  
}
```

```
.boxA p {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
.boxA ul {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
  line-height: 1.3;  
}
```

プロパティ: 値;
今度はこのセットをまとめたい



—
@amixin

ミックスイン

Sassで書くとこうなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```


Sassで書くとこうなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```

Sassで書くとこうなる

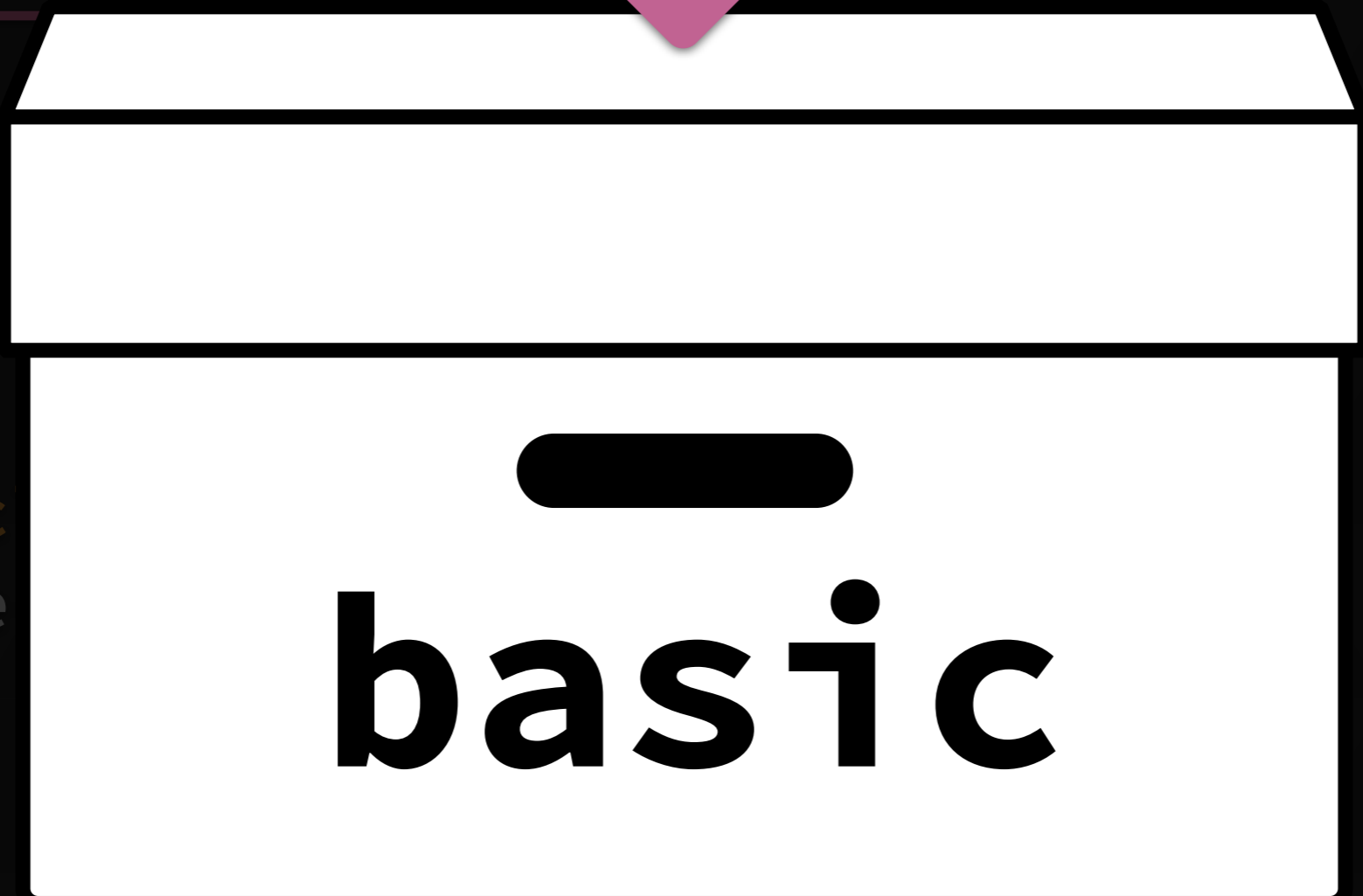
```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```

Sassで書

```
margin: 20px;  
padding: 10px;  
color: #595959;
```

```
@mixin basic  
margin  
padding  
color: #595959;  
}
```

```
.boxA {  
  p {  
    @inc  
  }  
  ul {  
    @inc  
    line  
  }  
}
```



Sassで書くとこうなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```

Sass Meisterで見る

Sassで書くところなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```

定義するとき

@mixin 名前{~}

Sass Meisterで見る

Sassで書くところなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```

定義するとき

@mixin 名前{~}

Sass Meisterで見る

Sassで書くところなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```

定義するとき

@mixin 名前{〜}

呼び出すとき

@include 名前;

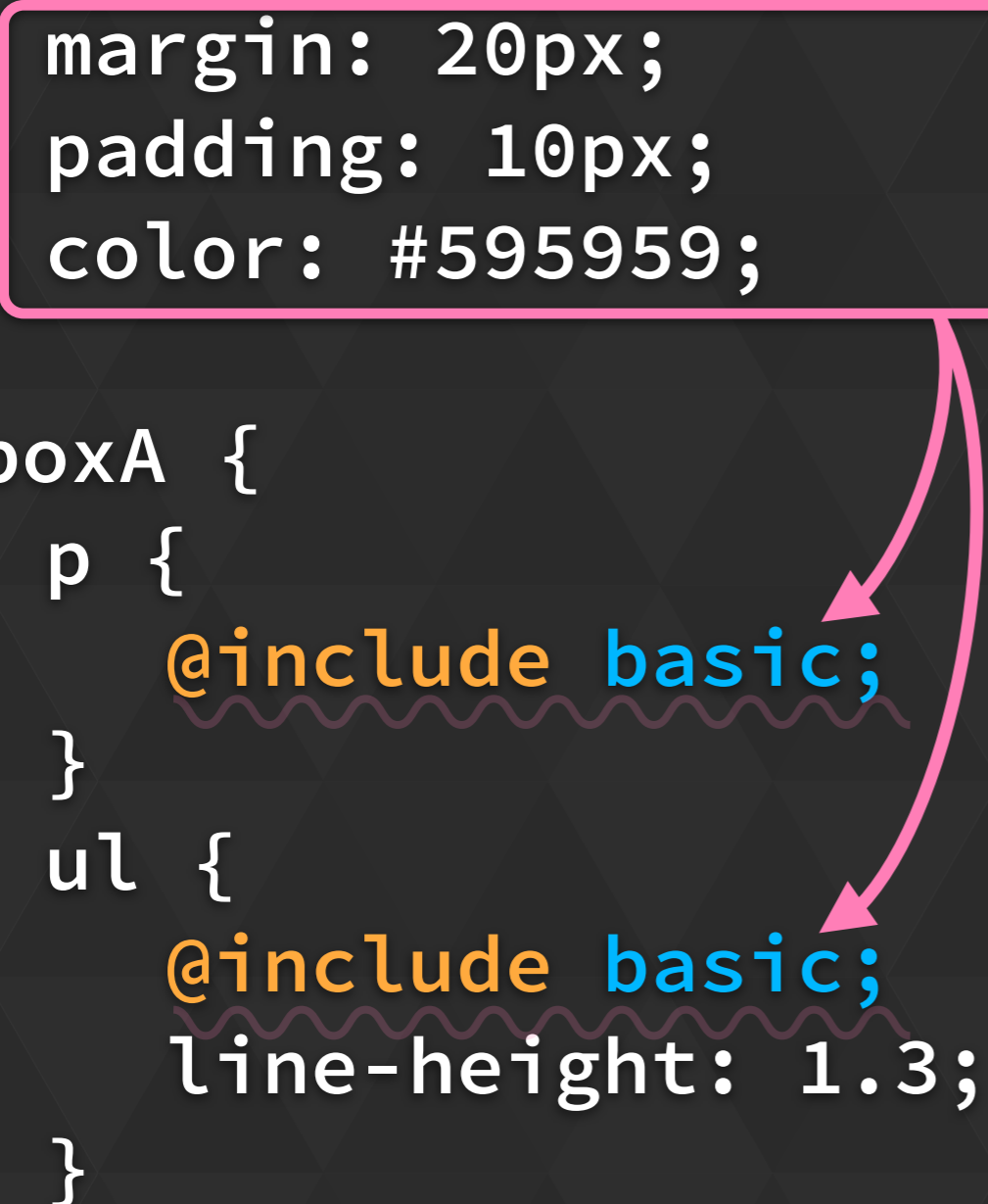
Sass Meisterで見る

Sassで書くとこうなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```

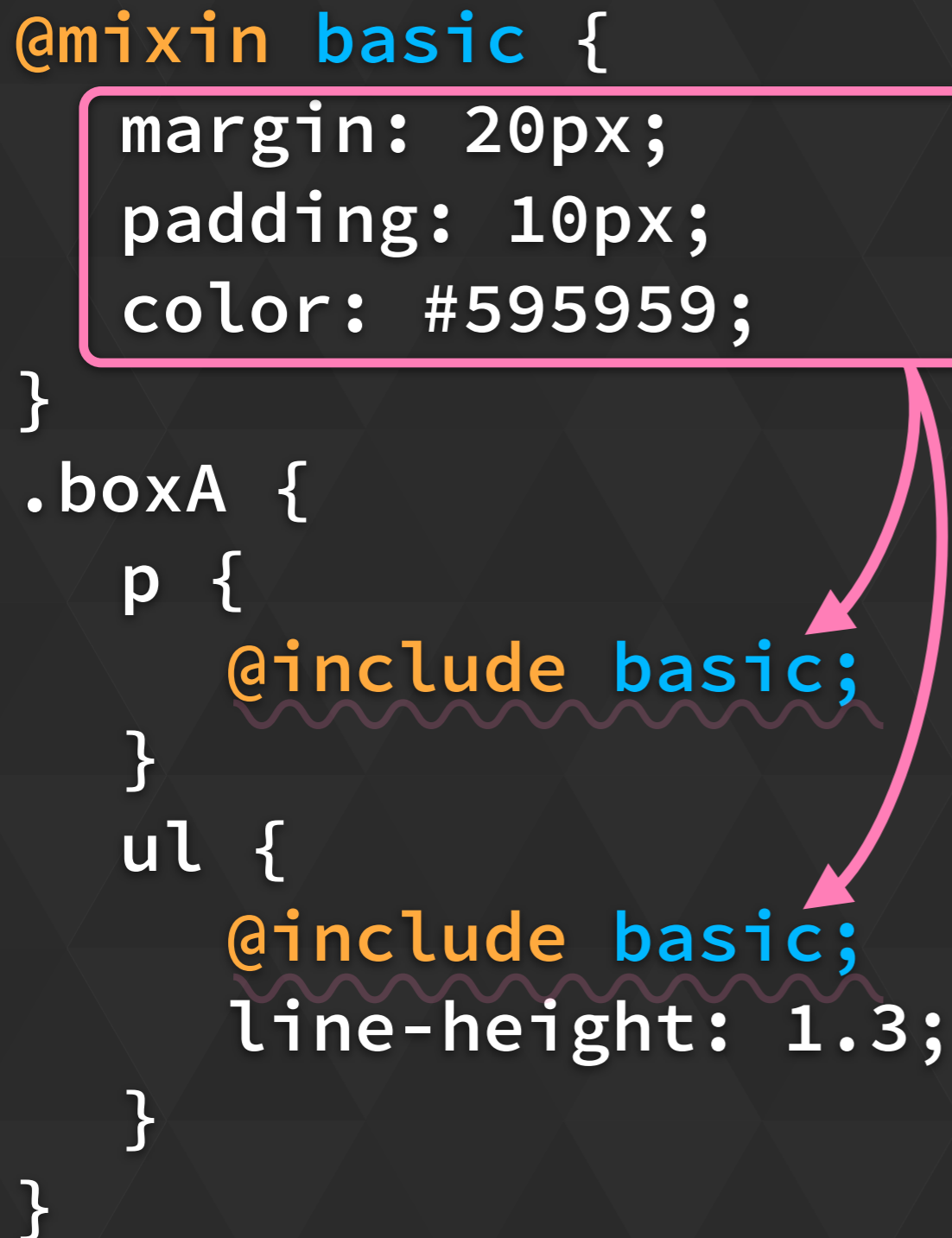
Sassで書くところなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```



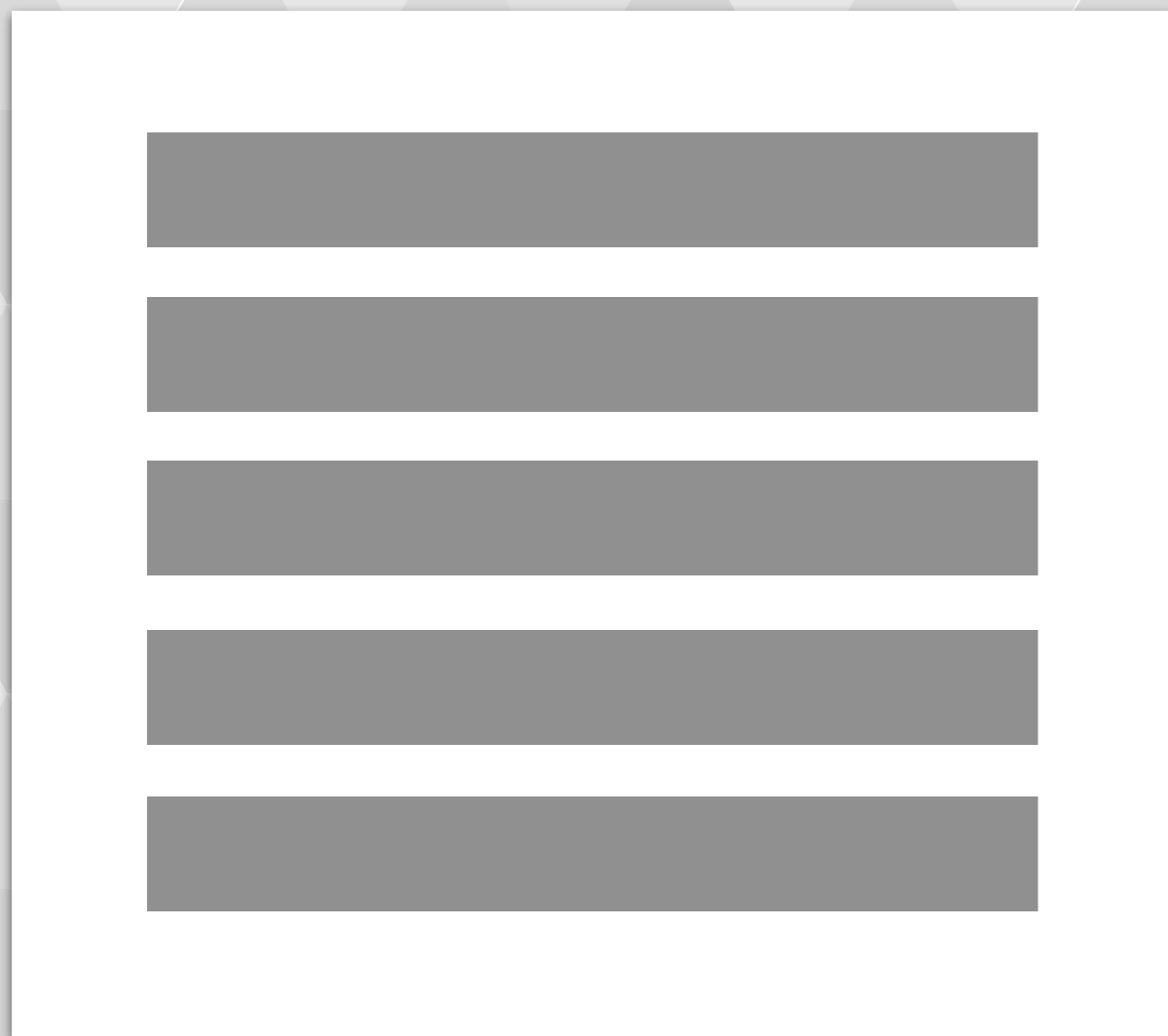
Sassで書くとこうなる

```
@mixin basic {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA {  
  p {  
    @include basic;  
  }  
  ul {  
    @include basic;  
    line-height: 1.3;  
  }  
}
```



```
.boxA p {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxA ul {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
  line-height: 1.3;  
}
```

今度是一部だけ違うとき

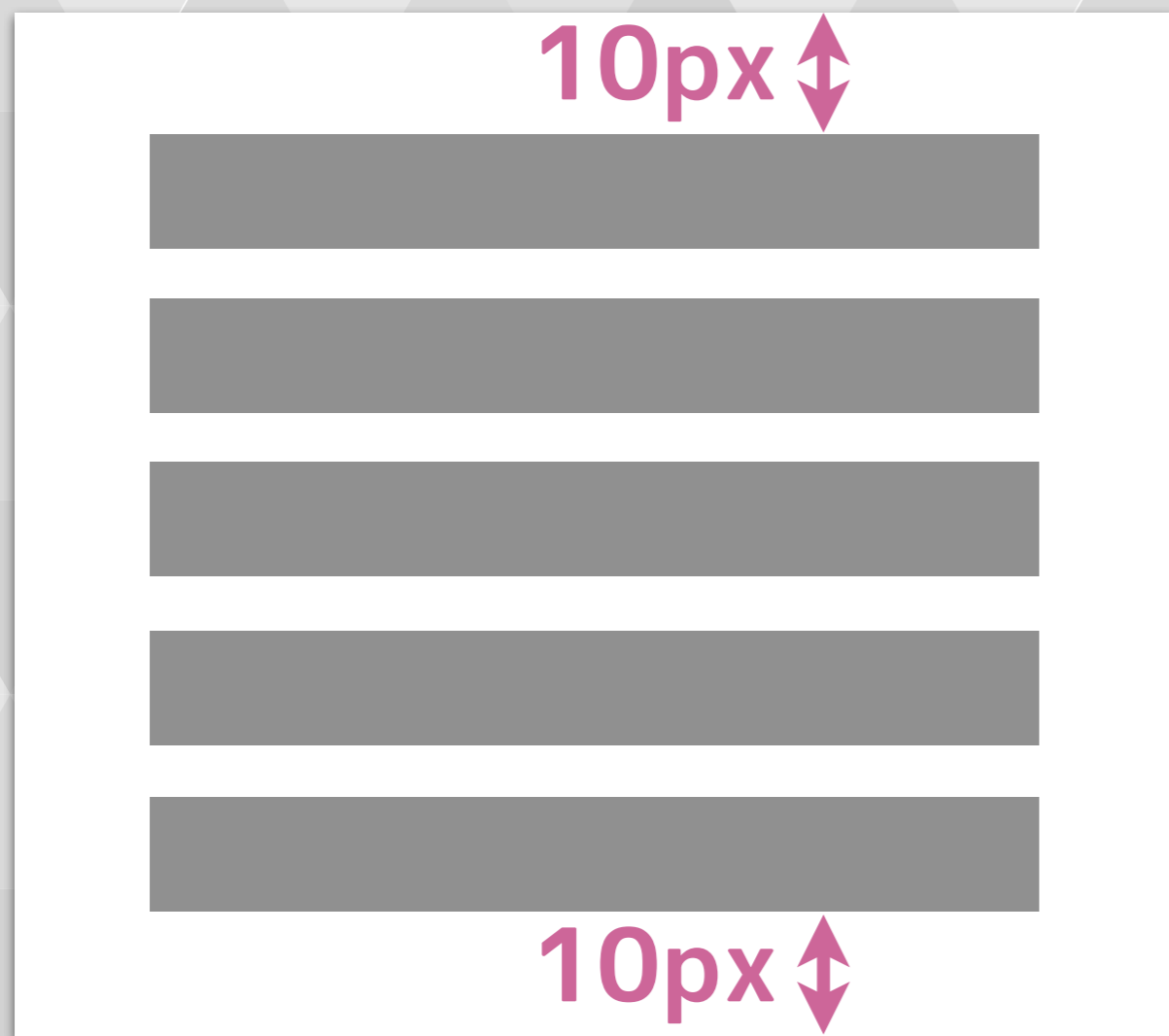


.boxA

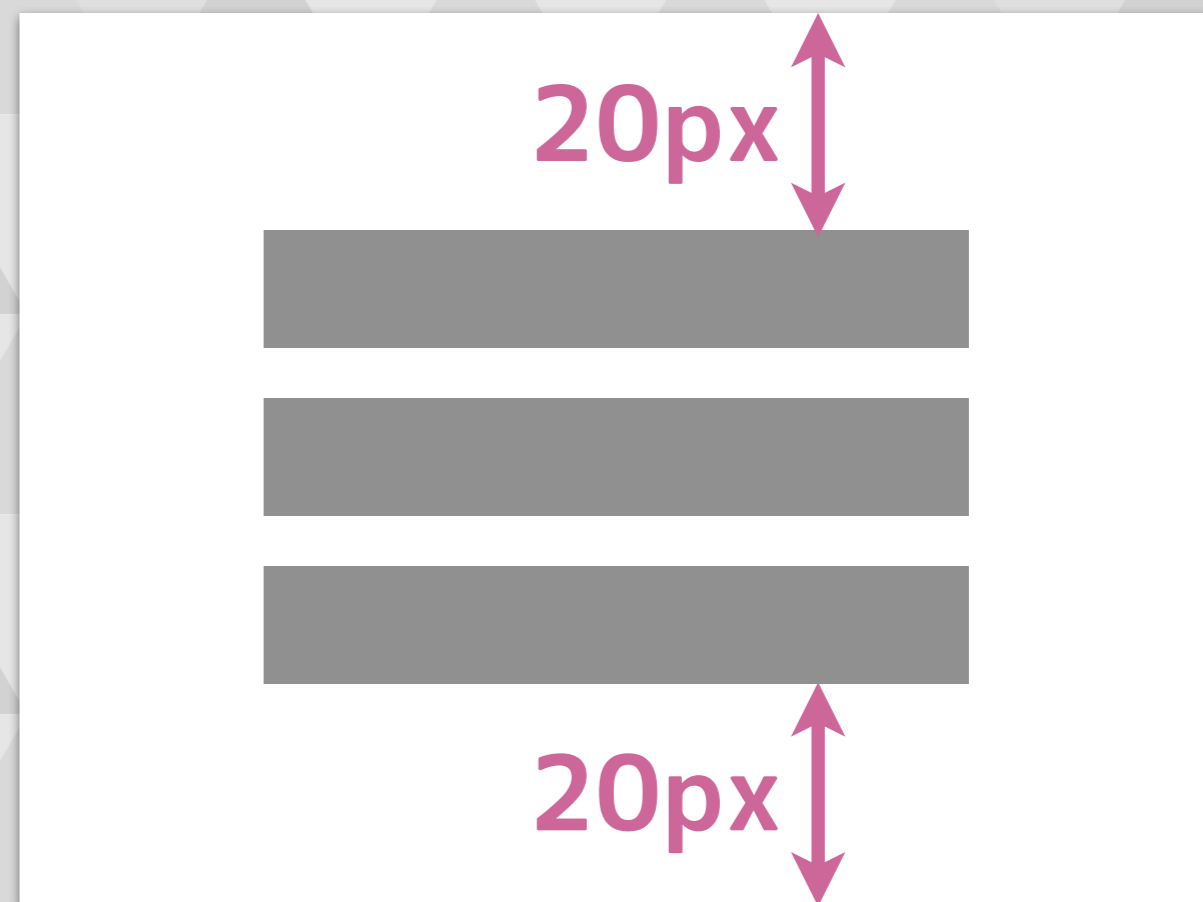


.boxB

今度は一部分だけ違うとき



.boxA

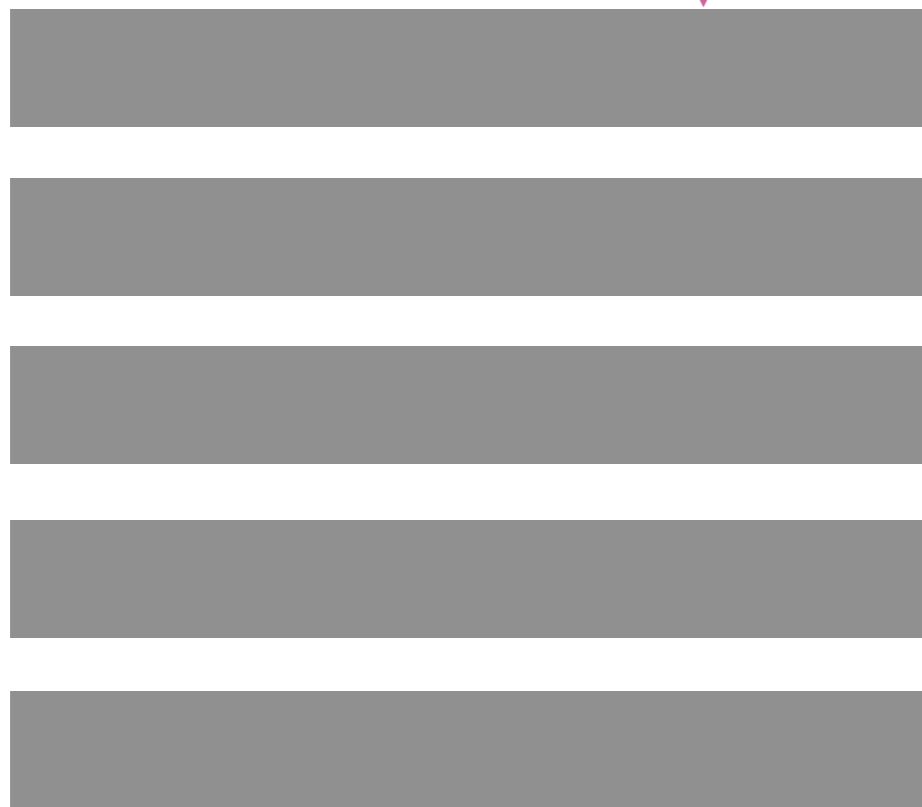


.boxB

今度は一部分だけ違うとき

paddingだけ違う

10px ↕



10px ↕

.boxA

20px ↕



20px ↕

.boxB

今度は一部だけ違うとき

```
.boxA {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxB {  
  margin: 20px;  
  padding: 20px;  
  color: #595959;  
}
```

Sassで書くとこうなる

```
@mixin basic($value) {  
  margin: 20px;  
  padding: $value;  
  color: #595959;  
}  
  
.boxA {  
  @include basic(10px);  
}  
  
.boxB {  
  @include basic(20px);  
}
```

Sassで書くところなる

```
@mixin basic($value)
  margin: 20px;
  padding: $value;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px);
}
```

定義するとき

@mixin 名前(\$変数名){}

Sassで書くところなる

```
@mixin basic($value)
  margin: 20px;
  padding: $value;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px);
}
```

定義するとき

```
@mixin 名前($変数名){}
```

Sassで書くところなる

```
@mixin basic($value)
  margin: 20px;
  padding: $value;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px)
}
```

定義するとき

@mixin 名前(\$変数名){}

呼び出すとき

@include 名前(値や変数);

Sassで書くところなる

```
@mixin basic($value)
  margin: 20px;
  padding: $value;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px)
}
```

定義するとき

@mixin 名前(\$変数名){}

呼び出すとき

@include 名前(値や変数);

Sassで書く **10px** る

```
@mixin basic($value)
  margin: 20px;
  padding: $value;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px)
}
```

定義するとき

@mixin 名前(**\$変数名**){}

呼び出すとき

@include 名前(**値や変数**);

Sassで書くところなる

```
@mixin basic($value)
  margin: 20px;
  padding: 10px;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px);
}
```

定義するとき

@mixin 名前(\$変数名){}

呼び出すとき

@include 名前(値や変数);

Sassで書くところなる

```
@mixin basic($value)
  margin: 20px;
  padding: $value;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px);
}
```

定義するとき

@mixin 名前(\$変数名){}

呼び出すとき

@include 名前(値や変数);

Sassで書く **20px** する

```
@mixin basic($value)
  margin: 20px;
  padding: 20px;
  color: #595959;
}

.boxA {
  @include basic(10px);
}

.boxB {
  @include basic(20px);
}
```

定義するとき

@mixin 名前(\$変数名){}

呼び出すとき

@include 名前(値や変数);

Sassで書くとこうなる

```
@mixin basic($value) {  
  margin: 20px;  
  padding: $value;  
  color: #595959;  
}  
.boxA {  
  @include basic(10px);  
}  
.boxB {  
  @include basic(20px);  
}
```

Sass Meisterで見る

Sassで書くとこうなる

```
@mixin basic($value) {  
  margin: 20px;  
  padding: $value;  
  color: #595959;  
}  
.boxA {  
  @include basic(10px);  
}  
.boxB {  
  @include basic(20px);  
}
```



```
.boxA {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
.boxB {  
  margin: 20px;  
  padding: 20px;  
  color: #595959;  
}
```

Sass Meisterで見る

Sassで書くとこうなる

```
@mixin basic($value) {  
  margin: 20px;  
  padding: $value;  
  color: #595959;  
}  
.boxA {  
  @include basic(10px);  
}  
.boxB {  
  @include basic(20px);  
}
```



```
.boxA {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
.boxB {  
  margin: 20px;  
  padding: 20px;  
  color: #595959;  
}
```

Sass Meisterで見る

Sassで書くところなる

```
@mixin basic($value) {  
  margin: 20px;  
  padding: $value;  
  color: #595959;  
}  
  
.boxA {  
  @include basic(10px);  
}  
  
.boxB {  
  @include basic(20px);  
}
```



```
.boxA {  
  margin: 20px;  
  padding: 10px;  
  color: #595959;  
}  
  
.boxB {  
  margin: 20px;  
  padding: 20px;  
  color: #595959;  
}
```

Sass Meisterで見る

$$\mathbf{C} + \mathbf{V}$$

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ

$$\mathbf{C} + \mathbf{V}$$

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ

ここが大変CSS

「,」でまとめると、
コードが把握しにくい

CSSでこういうコード書いたことありませんか？

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るといふ人のために、Sassの構文がコンパイ
ル（変換）でCSSがどのように変わるのか、
基本的な書き方を紹介します。

.noteA

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るといふ人のために、Sassの構文がコンパイ
ル（変換）でCSSがどのように変わるのか、
基本的な書き方を紹介します。

.noteB

CSSでこういうコード書いたことありませんか？

違いは線幅だけ

CSSで同じスタイルを何度も書くようなことがありませんか？CSSで幅の計算ができたなら？普段CSSを書く上で不便に感じていることはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみるという人のために、Sassの構文がコンパイル（変換）でCSSがどのように変わるのか、基本的な書き方を紹介します。

.noteA

CSSで同じスタイルを何度も書くようなことがありませんか？CSSで幅の計算ができたなら？普段CSSを書く上で不便に感じていることはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみるという人のために、Sassの構文がコンパイル（変換）でCSSがどのように変わるのか、基本的な書き方を紹介します。

.noteB

CSSでこういうコード書いたことありませんか？

```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}  
.noteB {  
  border-width: 4px;  
}
```

CSSでこういうコード書いたことありませんか？

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るといふ人のために、Sassの構文がコンパイ
ル（変換）でCSSがどのように変わるのか、
基本的な書き方を紹介します。

.noteA

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るといふ人のために、Sassの構文がコンパイ
ル（変換）でCSSがどのように変わるのか、
基本的な書き方を紹介します。

.noteB

CSSでこういうコード書いたことありませんか？

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るといふ人のために、Sassの構文がコンパイ
ル（変換）でCSSかどの上っに変わるのか、
基本的な書き方を紹介します。

.noteA

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るといふ人のために、Sassの構文がコンパイ
ル（変換）でCSSかどの上っに変わるのか、
基本的な書き方を紹介します。

.noteB

CSSでこういうコード書いたことありませんか？

CSSで同じスタイルを何度も書くようなことがありませんか？CSSで幅の計算ができたなら？普段CSSを書く上で不便に感じていることはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみるという人のために、Sassの構文がコンパイル（変換）でCSSかどのようになるのか、基本的な書き方を紹介します。

CSSで同じスタイルを何度も書くようなことがありませんか？CSSで幅の計算ができたなら？普段CSSを書く上で不便に感じていることはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみるという人のために、Sassの構文がコンパイル（変換）でCSSかどのようになるのか、基本的な書き方を紹介します。

• noteA

強調スタイルも
必要でした…。

• noteB

CSSでこういうコード書いたことありませんか？

```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}  
.noteB {  
  border-width: 4px;  
}
```

CSSでこういうコード書いたことありませんか？

```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}  
.noteA strong, .noteB strong {  
  color: #DF5496;  
}  
.noteB {  
  border-width: 4px;  
}
```

CSSでこういうコード書いたことありませんか？

```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}
```

見通しが悪い
||
ミスの原因

```
.noteA strong, .noteB strong {  
  color: #DF5496;  
}  
.noteB {  
  border-width: 4px;  
}
```

@extend





`@extend`

The diagram features a solid black rectangular box containing the text '@extend'. To its right and below is a larger dashed black rectangular box. A solid black arrow points from the bottom-right corner of the solid box towards the bottom-right corner of the dashed box.

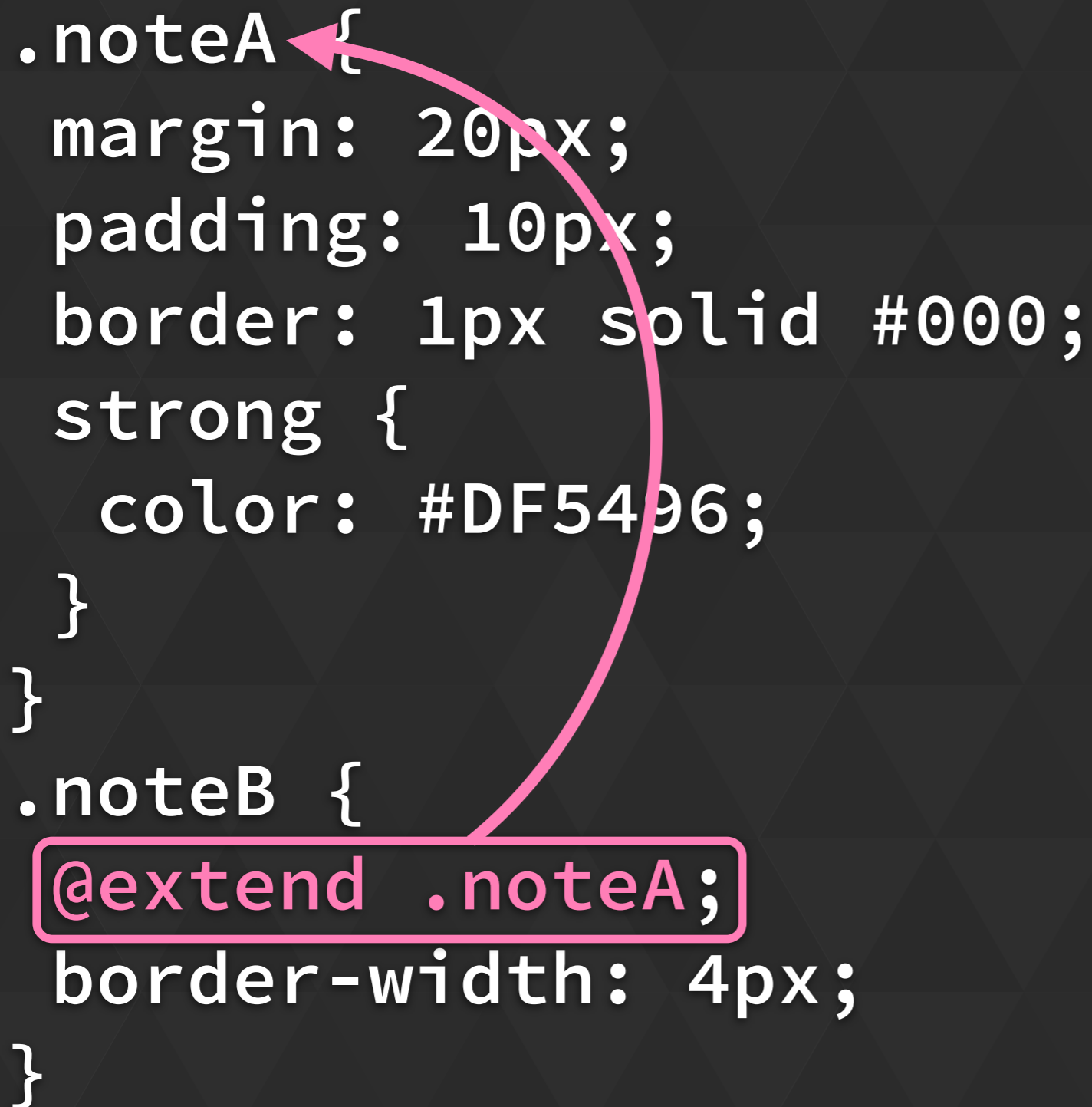
エクステンズ

Sassではこう書けます

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #DF5496;  
  }  
}  
  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```


Sassではこう書けます

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #DF5496;  
  }  
}  
  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```



Sassではこう書けます

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #DF5496;  
  }  
}  
  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```

セレクタを「,」でつなげる
@extend **セレクタ名**;

Sassではこう書けます

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #DF5496;  
  }  
}  
  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```

Sass Meisterで見る

Sassではこう書けます

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #DF5496;  
  }  
}  
  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```



```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}  
  
.noteA strong,  
.noteB strong {  
  color: #DF5496;  
}  
  
.noteB {  
  background-width: 4px;  
}
```

Sass Meisterで見る

Sassではこう書けます

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #DF5496;  
  }  
}  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```



```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}  
.noteA strong,  
.noteB strong {  
  color: #DF5496;  
}  
.noteB {  
  background-width: 4px;  
}
```

Sass Meisterで見る

CSSでこういうコード書いたことありませんか？

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るとい人のために、Sassの構文がコンパイ
ル（変換）でCSSがどのように変わるのか、
基本的な書き方を紹介します。

.noteA

CSSで同じスタイルを何度も書くようなこと
がありませんか？CSSで幅の計算ができた
ら？普段CSSを書く上で不便に感じているこ
とはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみ
るとい人のために、Sassの構文がコンパイ
ル（変換）でCSSがどのように変わるのか、
基本的な書き方を紹介します。

.noteB

.noteBが不要になりました！

CSSで同じスタイルを何度も書くようなことがありませんか？CSSで幅の計算ができたなら？普段CSSを書く上で不便に感じていることはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみるという人のために、Sassの構文がコンパイル（変換）でCSSがどのように変わるのか、基本的な書き方を紹介します。

.noteA

CSSで同じスタイルを何度も書くようなことがありませんか？CSSで幅の計算ができたなら？普段CSSを書く上で不便に感じていることはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみるという人のために、Sassの構文がコンパイル（変換）でCSSがどのように変わるのか、基本的な書き方を紹介します。

.noteB

.noteBが不要になりました！

CSSで同じスタイルを何度も書くようなことがありませんか？CSSで幅の計算ができたから？普段CSSを書く上で不便に感じていることはSassを使うことで解決できます。

本セッションでは、Sassをこれから使ってみるという人のために、Sassの構文がコンパイル（変換）でCSSがどのように変わるのか、基本的な書き方を紹介します。

.noteA

.noteBが不要になりました！

```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}  
  
.noteA strong, .noteB strong {  
  color: #DF5496;  
}  
  
.noteB {  
  border-width: 4px;  
}
```

.noteBが不要になりました！

```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}  
  
.noteA strong, .noteB strong {  
  color: #DF5496;  
}  
  
.noteB {  
  border-width: 4px;  
}
```

.noteBが不要になりました！

```
.noteA, .noteB {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
}
```

```
.noteA strong, .noteB strong {  
  color: #DF5496;  
}
```

```
.noteB {  
  border-width: 4px;  
}
```

3

箇所

Sassで書くと

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #CD6699;  
  }  
}  
  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```


Sassで書くと

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #ccc;  
  strong  
    color: red;  
}  
}  
  
.noteB {  
  @extend .noteA;  
  border-width: 4px;  
}
```



1
箇所

Sassで書くと

```
.noteA {  
  margin: 20px;  
  padding: 10px;  
  border: 1px solid #000;  
  strong {  
    color: #CD6699;  
  }  
}
```

削除範囲が近い

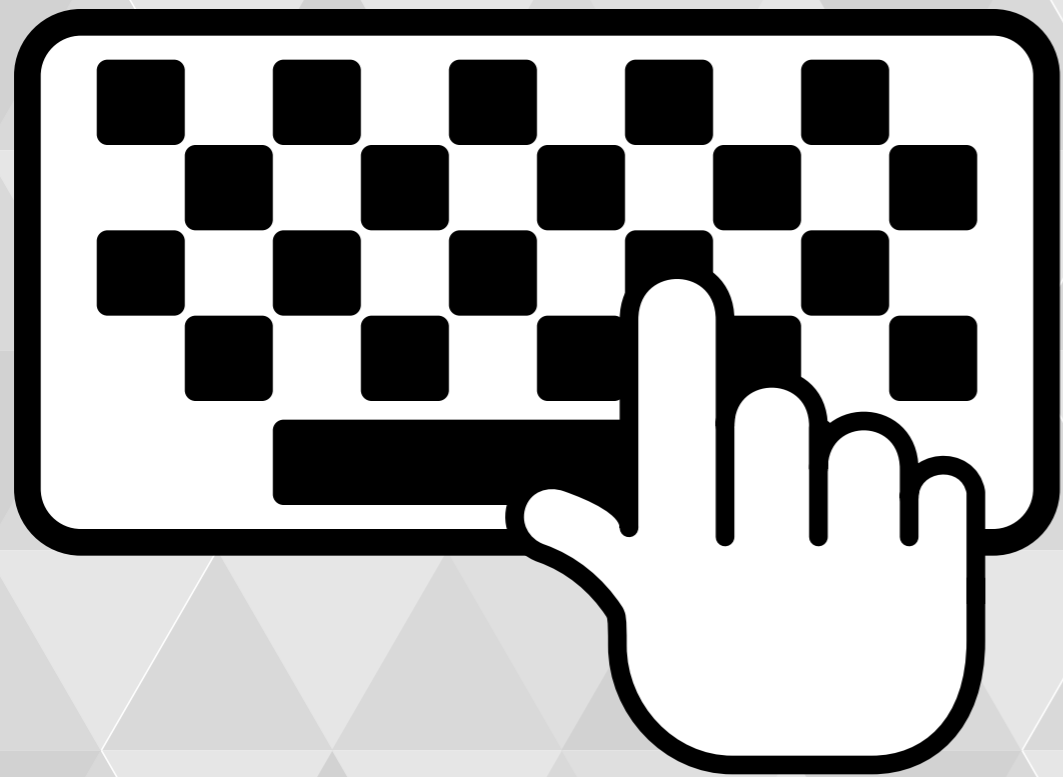
||

消し忘れが
起こりにくい

```
.noteB {  
@extend .noteA;  
border-width: 4px;  
}
```

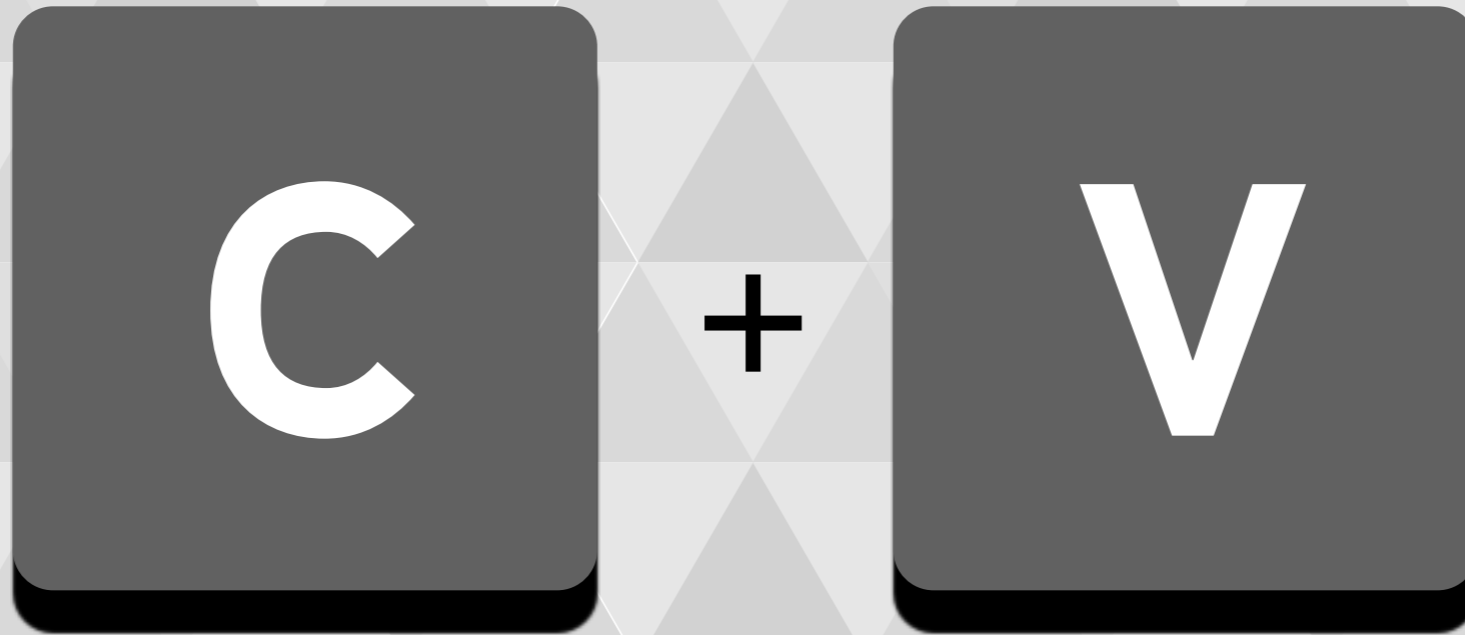
C + **V**

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ

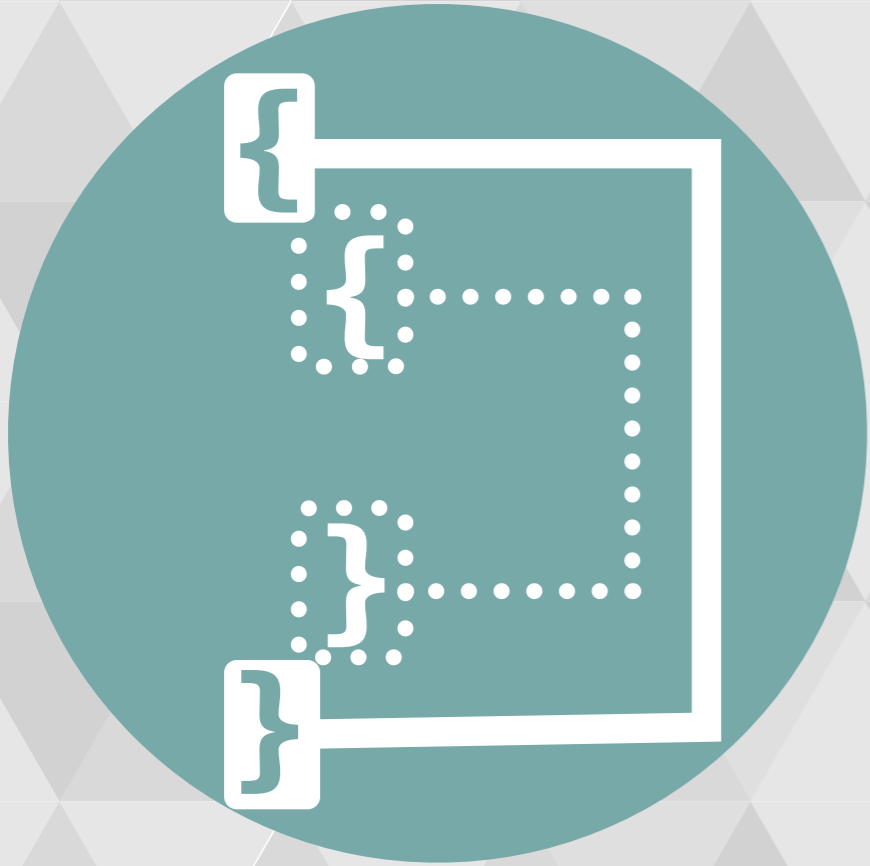


入力の手間を
省略





- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ



ネスト（入れ子）

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ



変数

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ



`@mixin`

`@mixin` で定義
`@include` で呼び出し

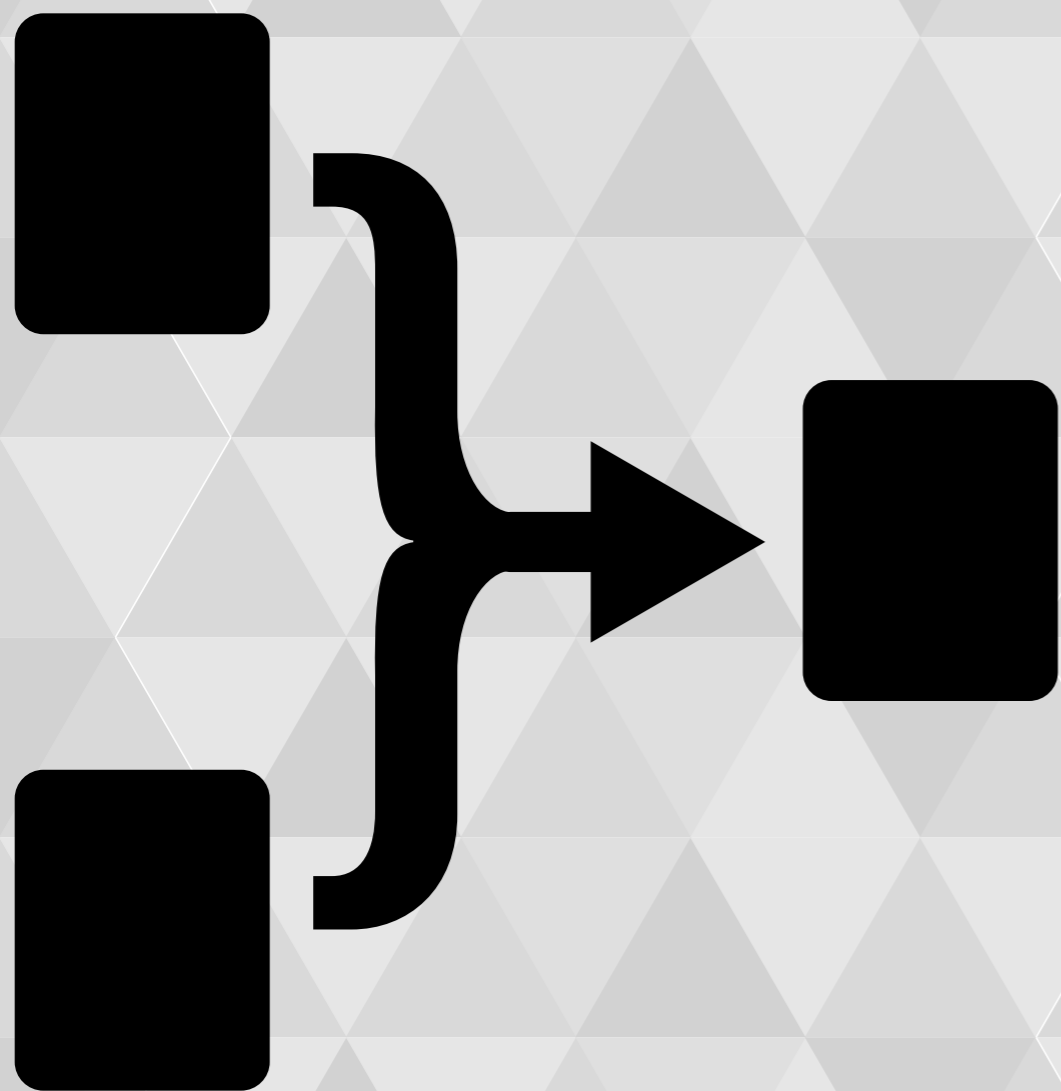
- 共通のセレクト
- 同じ値
- 同じプロパティと値
- セレクトのグループ



@extend

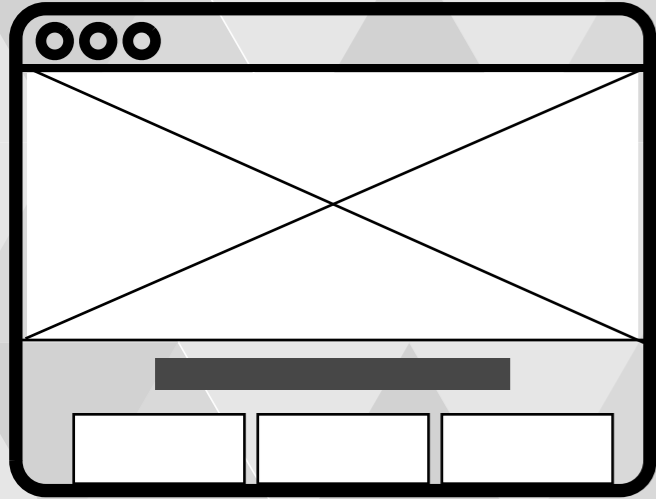
@extendで 「,」 でつなげる

- 共通のセレクタ
- 同じ値
- 同じプロパティと値
- セレクタのグループ



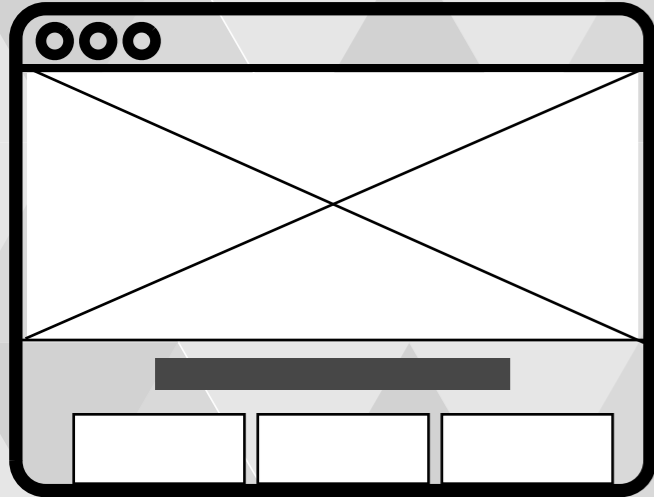
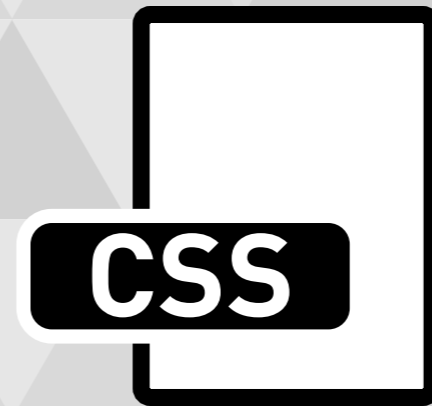
CSSを
まとめる



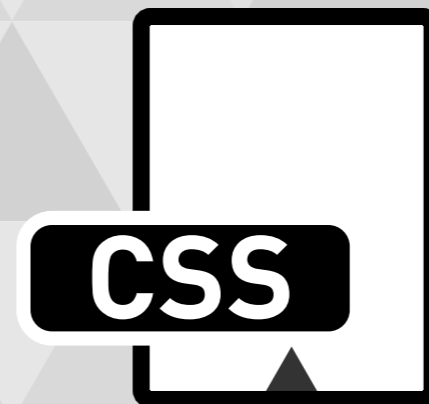




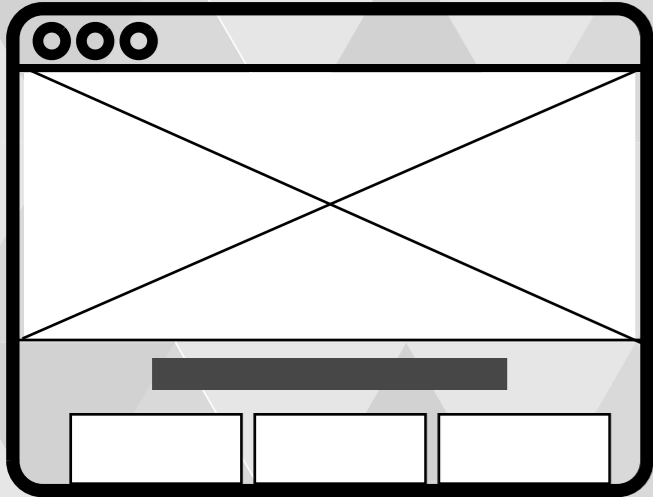
import.css



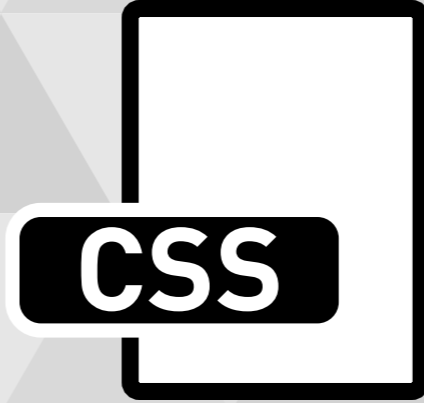
import.css



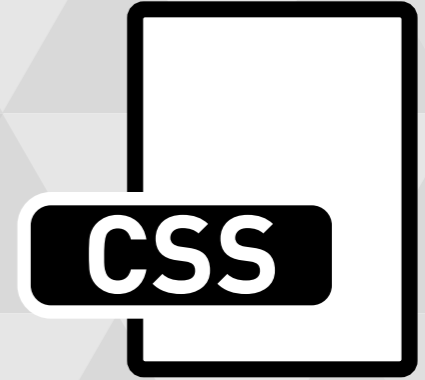
```
@import url("reset.css");  
@import url("layout.css");  
@import url("basic.css");
```



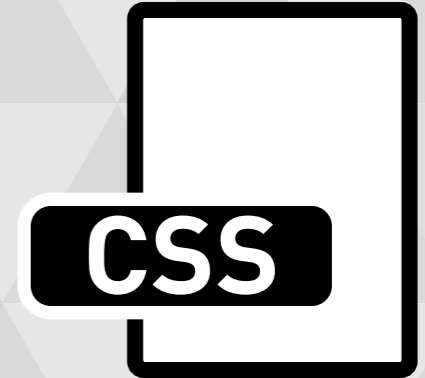
import.css



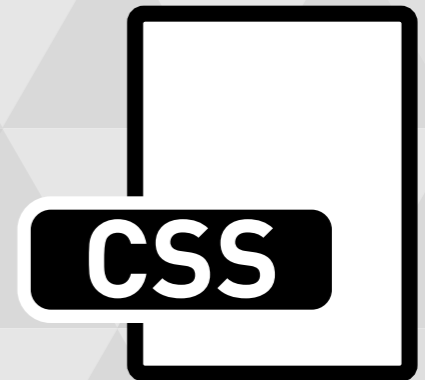
reset.css



layout.css



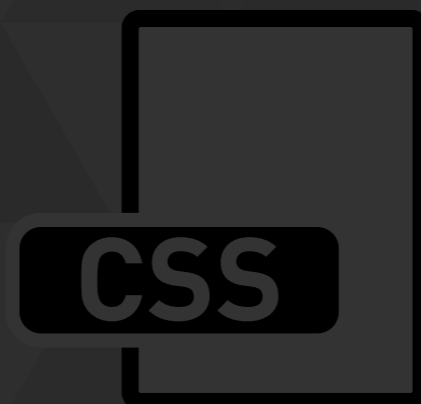
basic.css



4

requests

4回、サーバーからCSSをダウンロードしてこよう必要があった



layout.css



basic.css



import.css

reset.css



SPEED

4

HTML

import.css

CSS

reset.css

CSS

layout.css

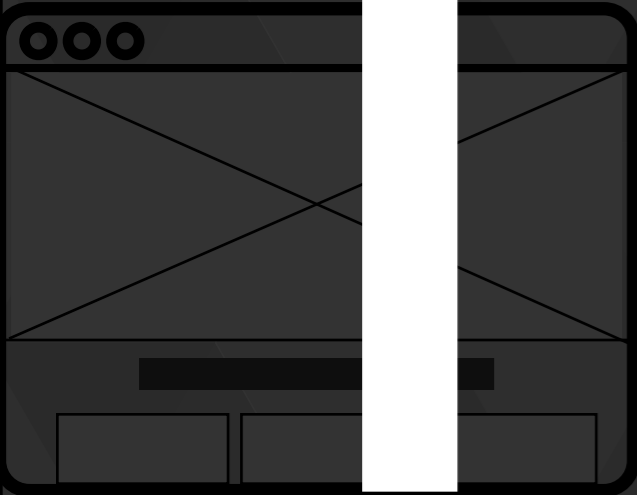
requests

basic.css

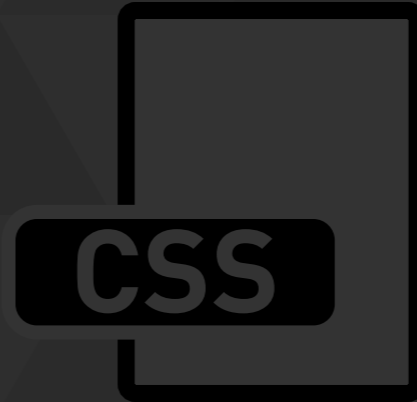
CSS

4回、サーバーからCSSをダウンロードしてこなければならない

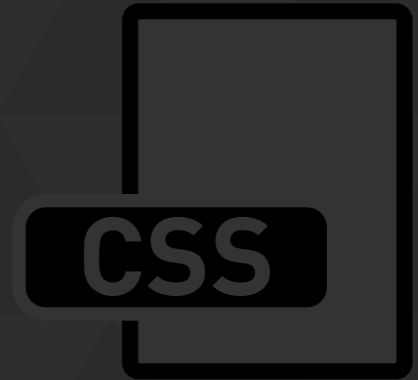
1



import.css



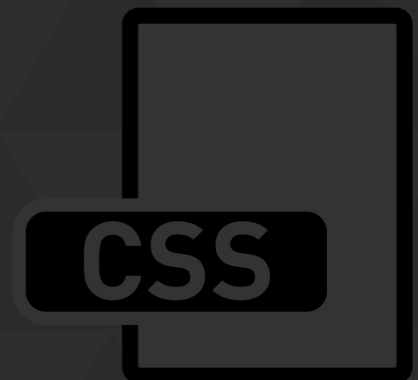
reset.css



layout.css



basic.css

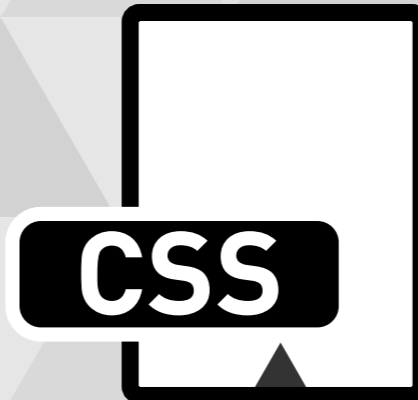


request

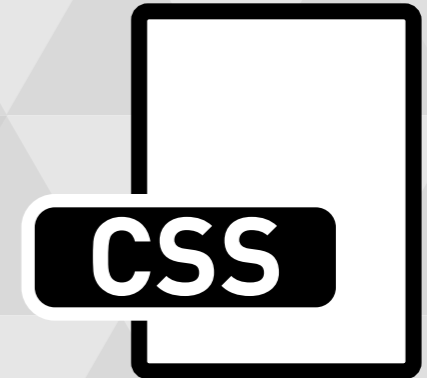
CSSを一つにまとめると高速化に結びつけやすい



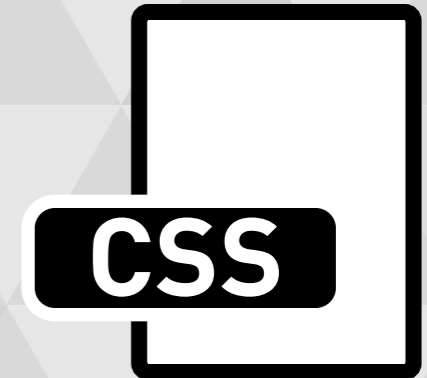
import.css



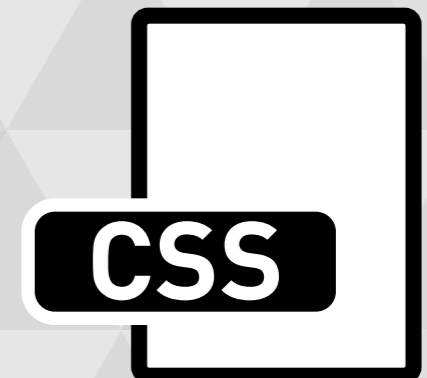
reset.css



layout.css

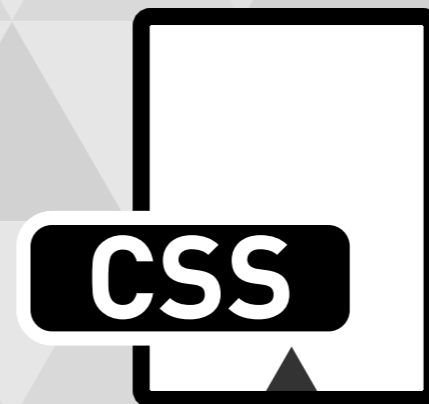


basic.css

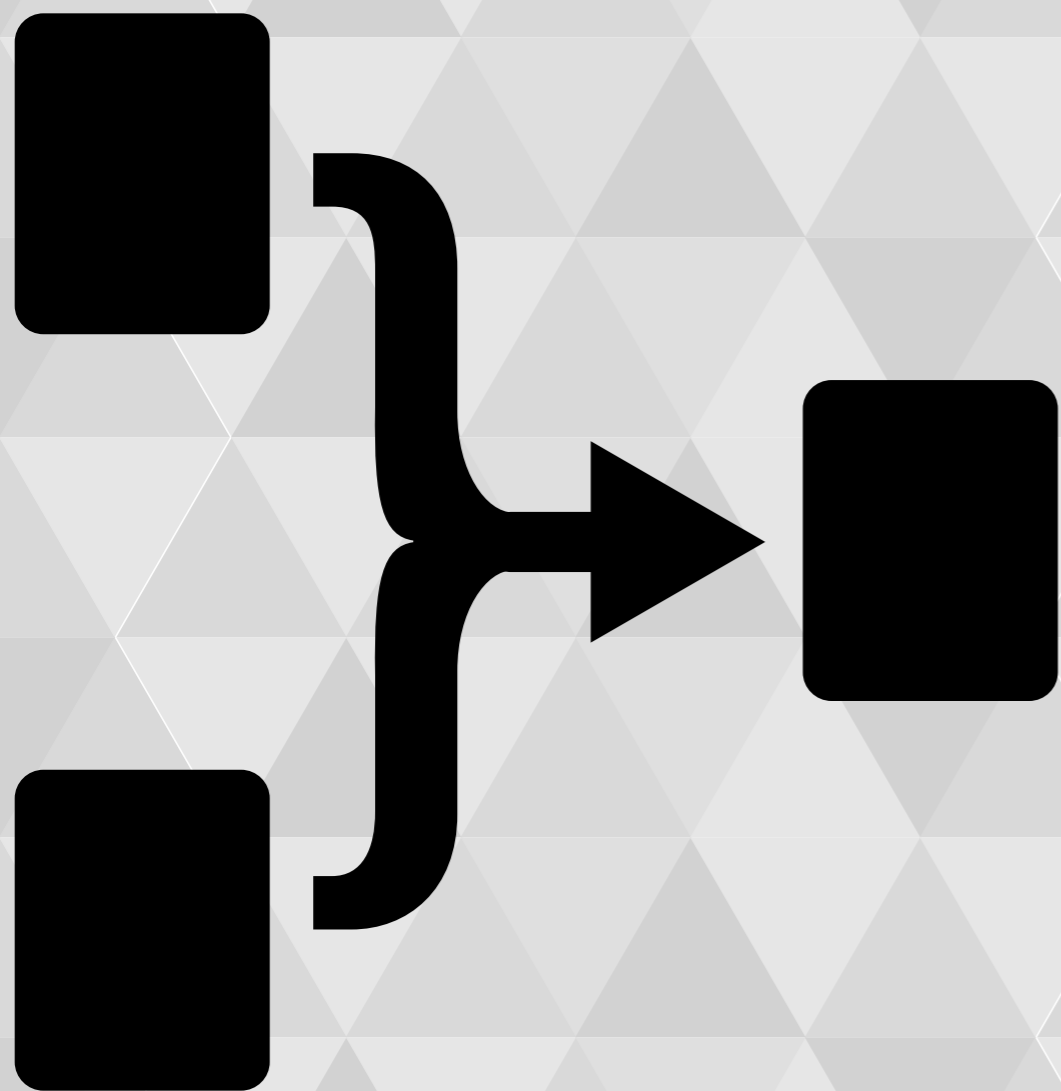


```
@import url("reset.css");  
@import url("layout.css");  
@import url("basic.css");
```

import.css

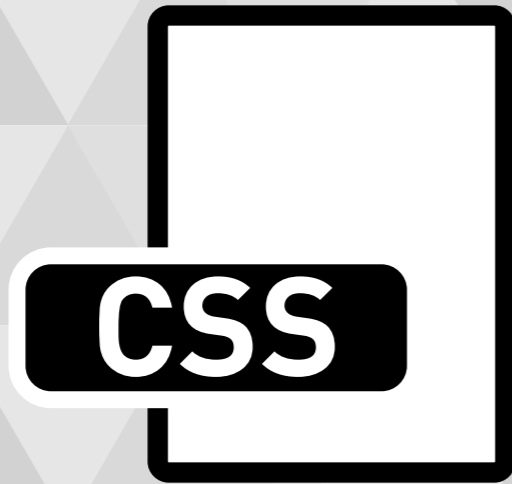


```
/* reset.cssの内容 */  
html { ...  
/* layout.cssの内容 */  
#main { ...  
/* basic.cssの内容 */  
body { ...
```

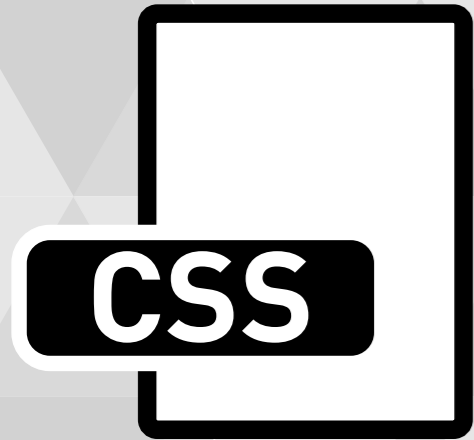


CSSを
まとめる

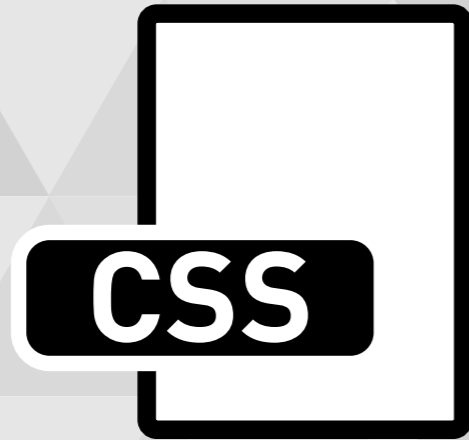




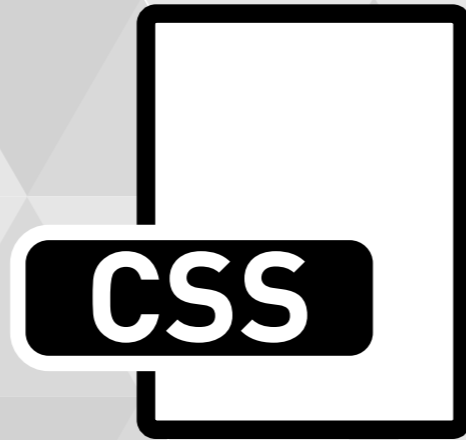
import.css



reset.css



layout.css



basic.css



`import.scss`



`_reset.scss`



`_layout.scss`



`_basic.scss`



import.scss



_reset.scss



_layout.scss



_basic.scss

ファイル名先頭に
_ (アンダースコア)

単一ファイルでコンパイルしなくなる



`import.scss`

```
@import url("reset.css");  
@import url("layout.css");  
@import url("basic.css");
```



Sa

im

アンダースコアと
拡張子.scssを取る

```
@import "reset";  
@import "layout";  
@import "basic";
```



import.scss

```
@import "reset", "layout", "basic";
```



import.scss

「,」でまとめる

```
@import "reset", "layout", "basic";
```



`import.scss`



`_reset.scss`



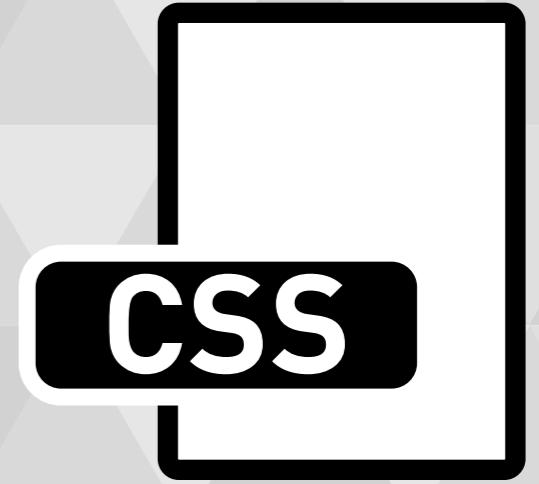
`_layout.scss`



`_basic.scss`



`import.scss`



`import.css`



`_reset.scss`



`_layout.scss`



`_basic.scss`

```
/* reset.cssの内容 */
```

```
html { ...
```

```
/* layout.cssの内容 */
```

```
#main { ...
```

```
/* basic.cssの内容 */
```

```
body { ...
```

_reset.scss

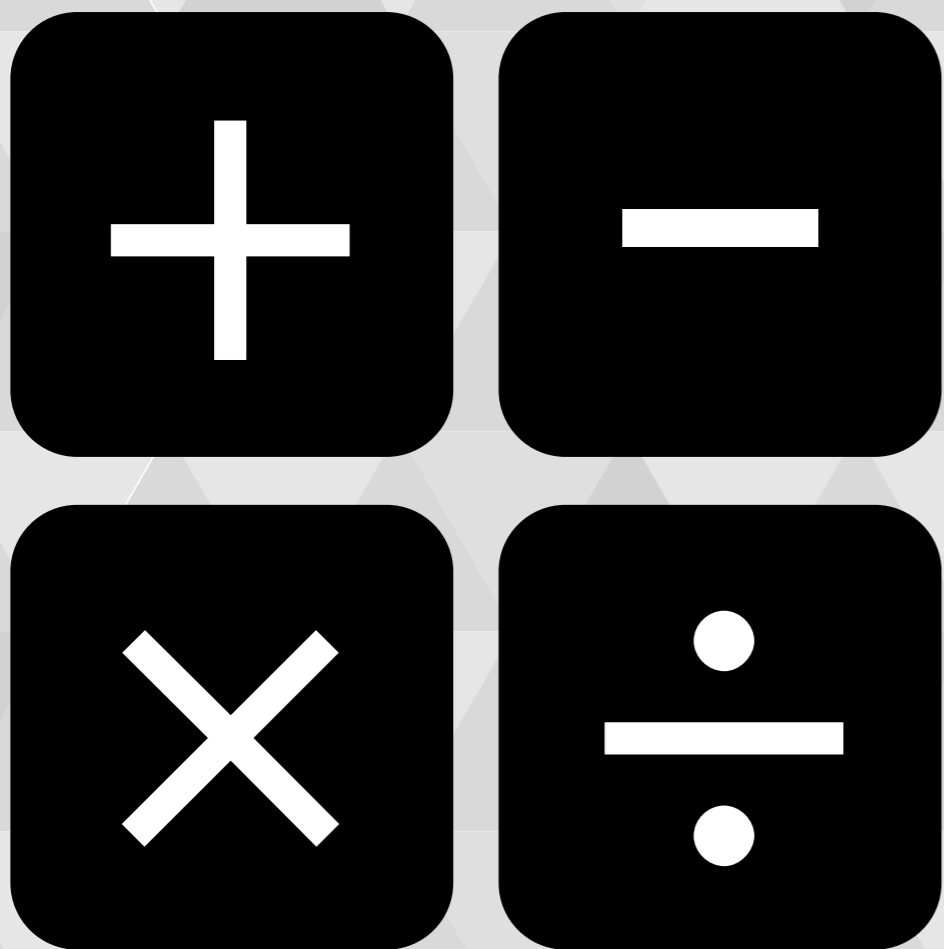
_layout.scss

_basic.scss

A white document icon with a black border and a black tab at the top containing the text "CSS" in white.

CSS

import.css



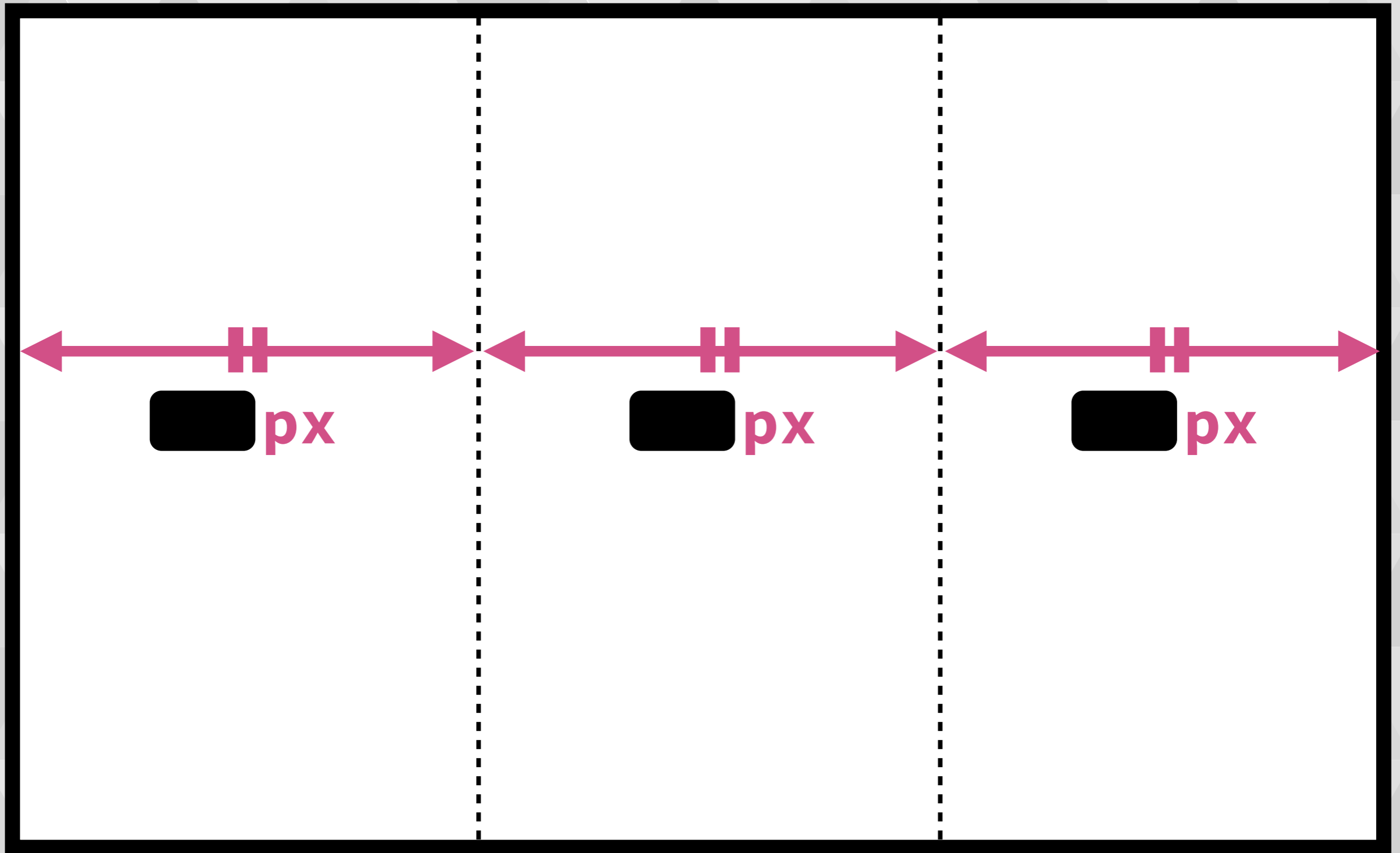
計算できる



ここが大変CSS

余白計算は電卓...

960px

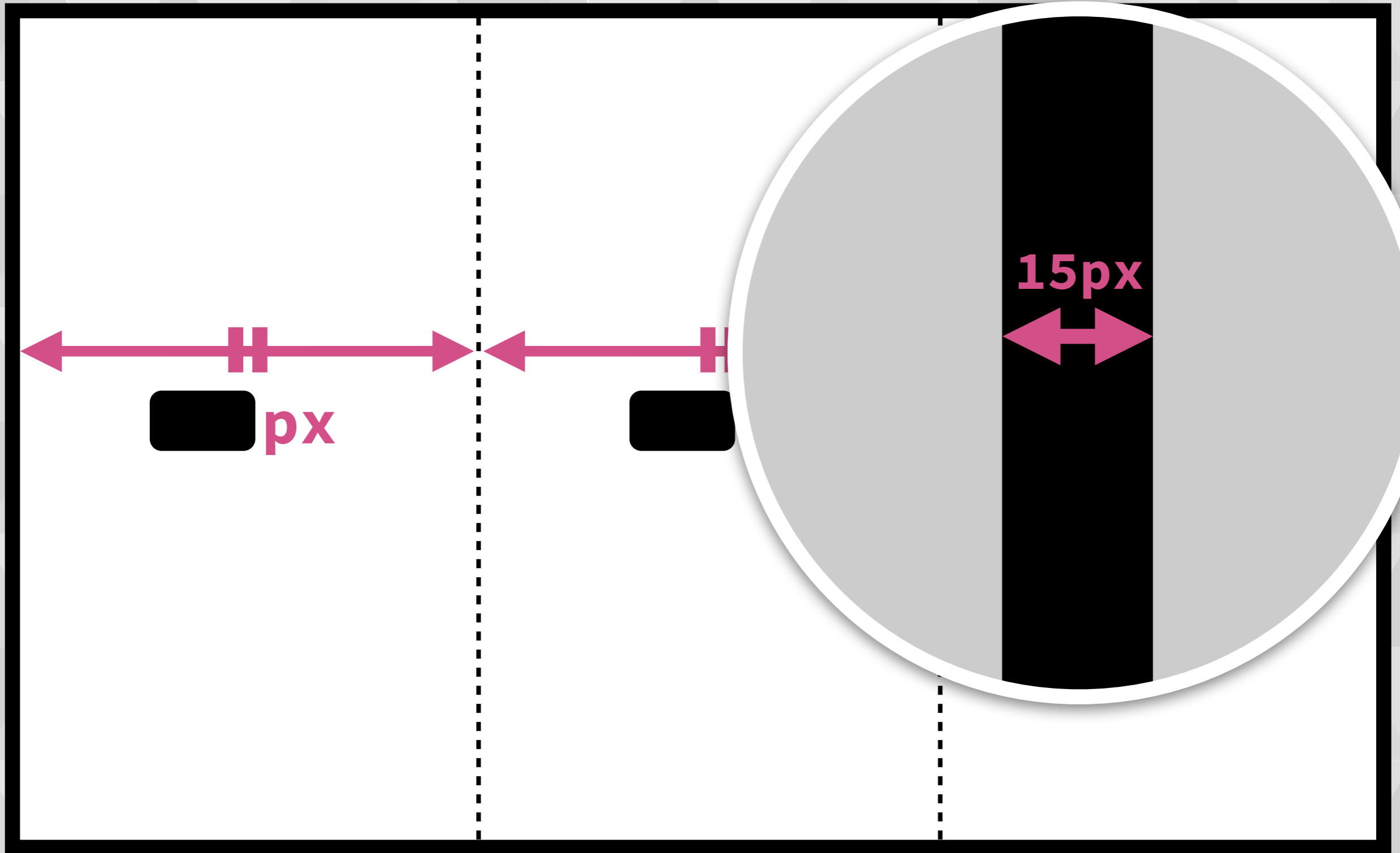


px

px

px

960px



px

15px

$$(960 - 15 * 2) / 3$$

$$\frac{(960 - 15 * 2)}{3}$$

全体幅 左右のボーダー幅 カラム数

[ウェブ](#)[画像](#)[地図](#)[ショッピング](#)[もっと見る](#)[検索ツール](#)

約 13,150,000,000 件 (0.16 秒)

(960 - (15 * 2)) / 3 =

310

Rad		x!	()	%	AC
Inv	sin	ln	7	8	9	÷
π	cos	log	4	5	6	×
e	tan	√	1	2	3	-
Ans	EXP	x ^y	0	.	=	+

[詳細](#)

Sassで書くと

```
.wrap {  
  width: 960px;  
  border: 15px solid #000;  
  > div {  
    float: left;  
    width: (960px - 15 * 2) / 3;  
  }  
}
```

Sass Meisterで見る

Sassで書くと

```
.wrap {  
  width: 960px;  
  border: 15px solid #000;  
  > div {  
    float: left;  
    width: (960px - 15 * 2) / 3;  
  }  
}
```

Sass Meisterで見る

変換するところなる

```
.wrap {  
  width: 960px;  
  border: 15px solid black;  
}  
  
.wrap > div {  
  float: left;  
  width: 310px;  
}
```

+ 足し算

- 引き算

***** かけ算

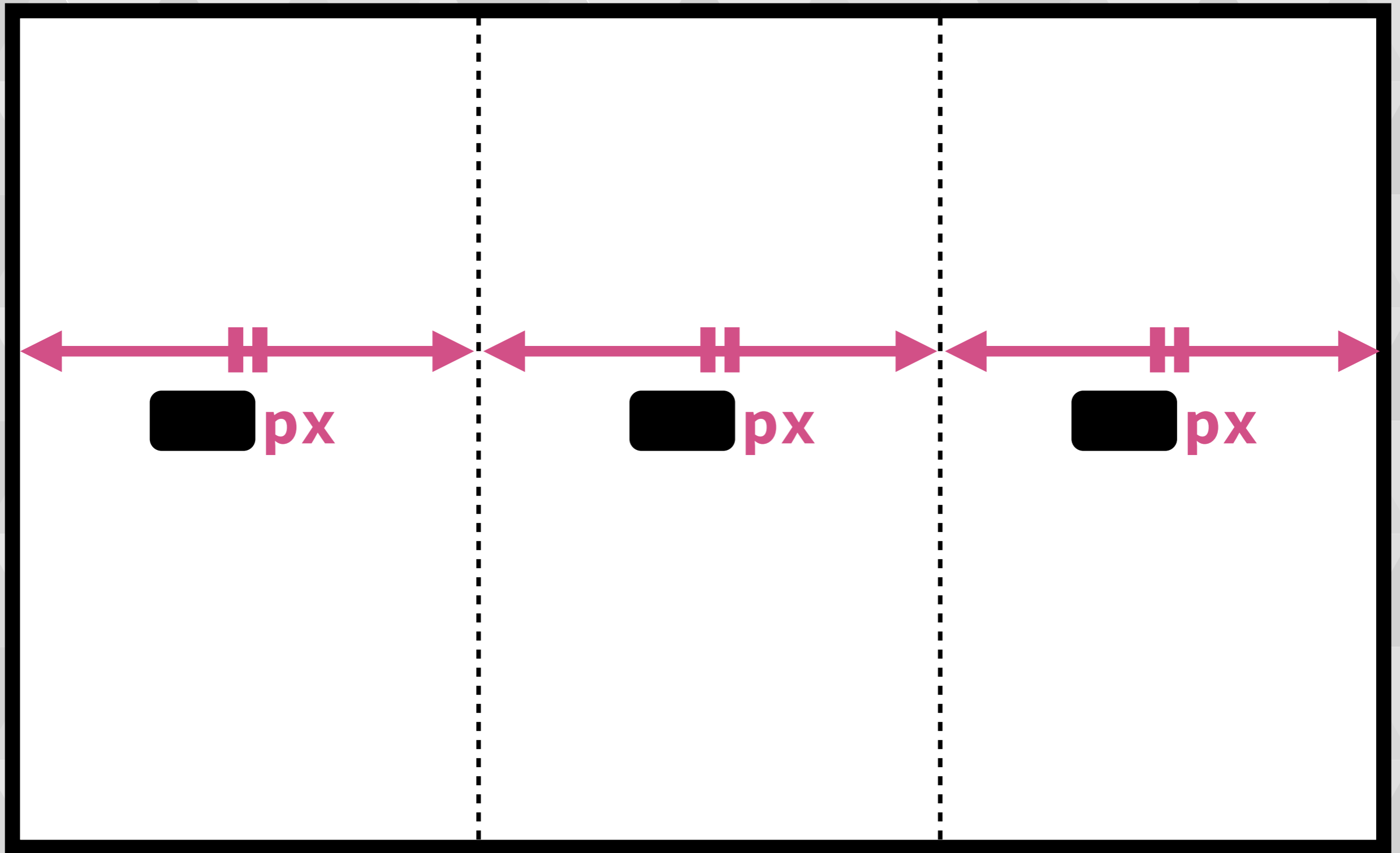
/ 割り算

Sassで書くと

```
.wrap {  
  width: 960px;  
  border: 15px solid #000;  
  > div {  
    float: left;  
    width: (960px - 15 * 2) / 3;  
  }  
}
```

半角スペース入れる

960px

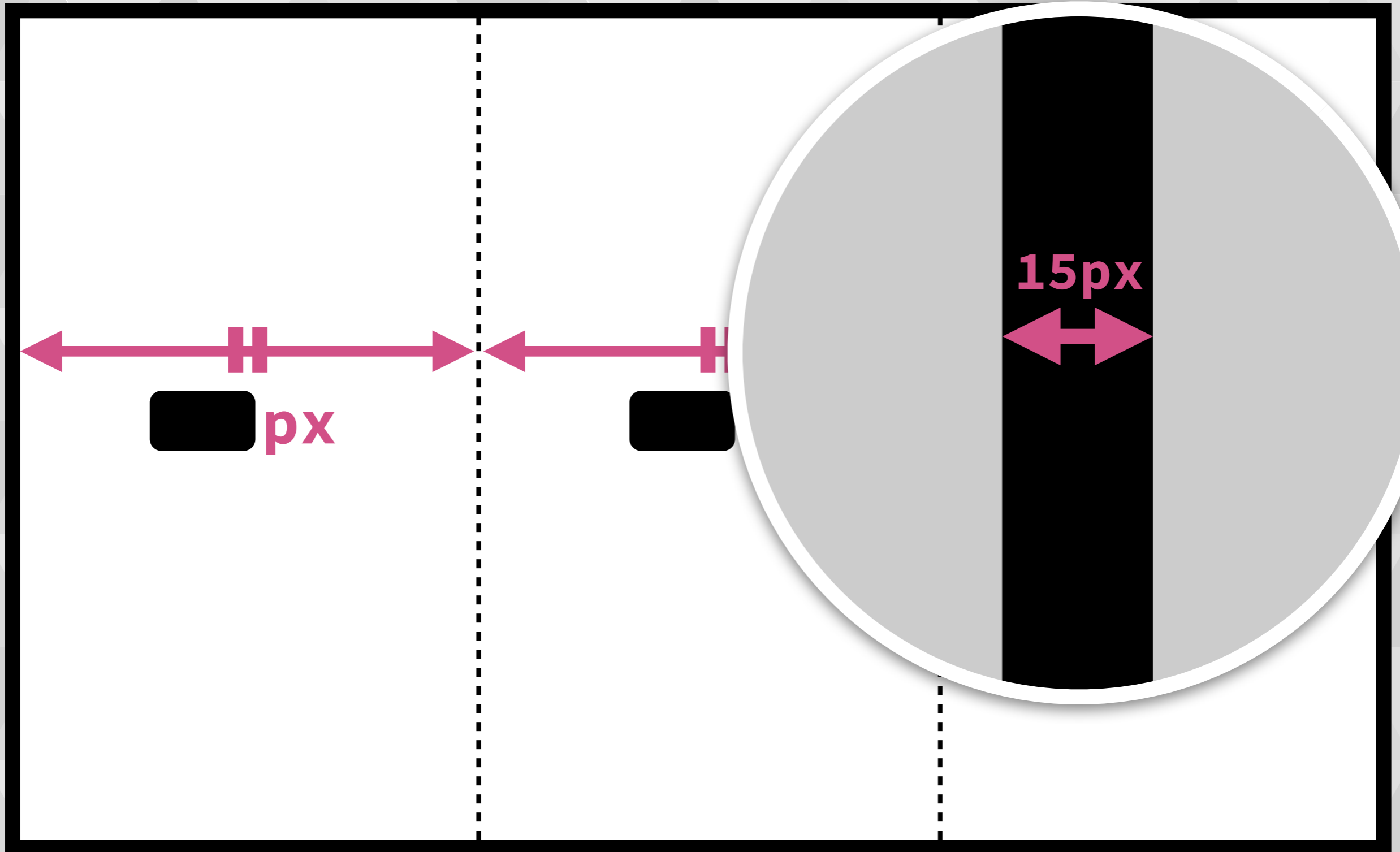


px

px

px

960px



1px

15px

Sassで書くと

```
.wrap {  
  width: 960px;  
  border: 15px solid #000;  
  > div {  
    float: left;  
    width: (960px - 15 * 2) / 3;  
  }  
}
```

Sassで書くと

```
.wrap {  
  width: 960px;  
  border: 15px solid #000;  
  > div {  
    float: left;  
    width: (960px - 15 * 2) / 3;  
  }  
}
```

Sassで書くと

```
.wrap {  
  width: 960px;  
  border: 15px solid #000;  
  > div {  
    float: left;  
    width: (960px - 15 * 2) / 3;  
  }  
}
```

Sassで書くと

```
.wrap {  
  width: 960px;  
  border: 15px solid #ccc;  
  > div {  
    float: left;  
    width: (960px - 15 * 2) / 3;  
  }  
}
```

根拠が同じ値

変数を使うと

```
.wrap {  
  $bd_width : 15px;  
  width: 960px;  
  border: $bd_width solid #000;  
  > div {  
    float: left;  
    width: (960px - $bd_width * 2) / 3;  
  }  
}
```

Sass Meisterで見る

CSSに変換すると

```
.wrap {  
  width: 960px;  
  border: 15px solid black;  
}  
  
.wrap > div {  
  float: left;  
  width: 310px;  
}
```

CSSに変換すると

```
.wrap {  
  width: 960px;  
  border: 15px solid black;  
}  
  
.wrap > div {  
  float: left;  
  width: 310px;  
}
```

CSSに変換すると

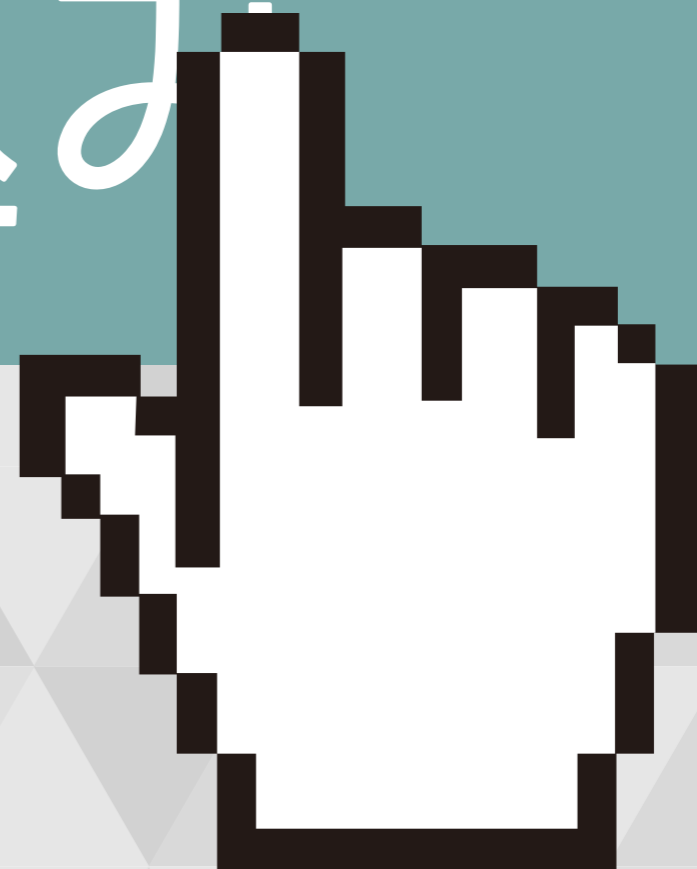
```
.wrap {  
  width: 960px;  
  border: 15px solid black;  
}  
  
.wrap > div {  
  float: left;  
  width: 310px;  
}
```

ここが大変CSS

色の微調整ができない

お申し込み

お申し込み



:hover 忘れた...

お申し込み



Sassで書くとこうなる

```
button {  
  background: #669999;  
  &:hover {  
    background: darken(#669999, 20%);  
  }  
}
```

Sass Meisterで見る

Sassで書くとこうなる

```
button {  
  background: #  
  &:hover {  
    background: darken(#669999 , 20%);  
  }  
}
```

暗くする

Sass Meisterで見る

Sassで書くとこうなる

```
button {  
  background: #  
  &:hover {  
    background: darken(#669999, 20%);  
  }  
}
```

暗くする

どれぐらい？

Sass Meisterで見る

Sassで書くところなる

```
button {  
  background: #  
  &:hover {  
    background: darken(#669999, 20%);  
  }  
}
```

暗くする

どれぐらい？

親セレクタ

button

Sass Meisterで見る

Sassで書くところなる

```
button {  
  background: #669999;  
  &:hover {  
    background: darken(#669999, 20%);  
  }  
}
```

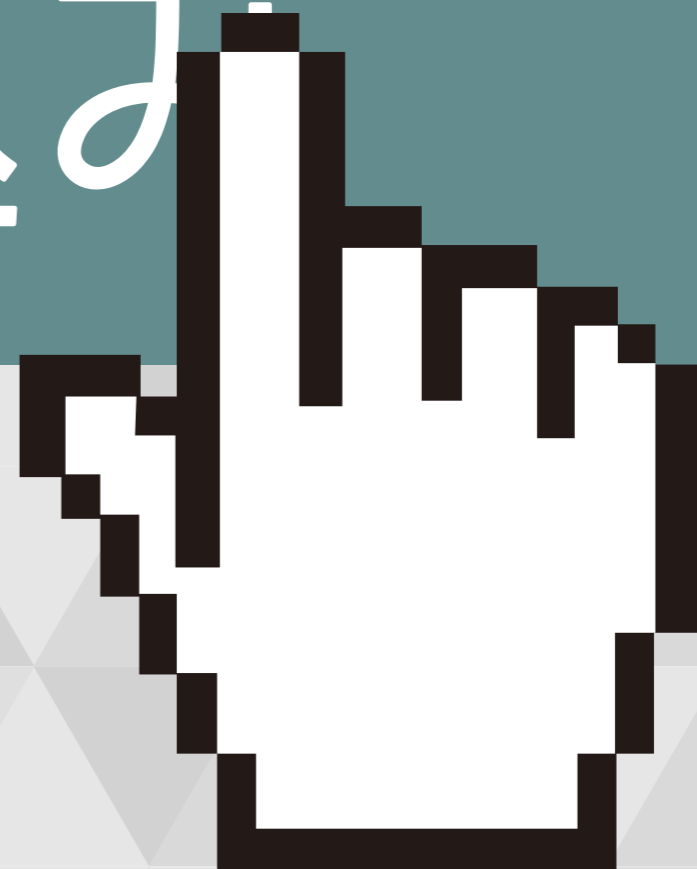


```
button {  
  background: #669999;  
}  
button:hover {  
  background: #527a7a;  
}
```

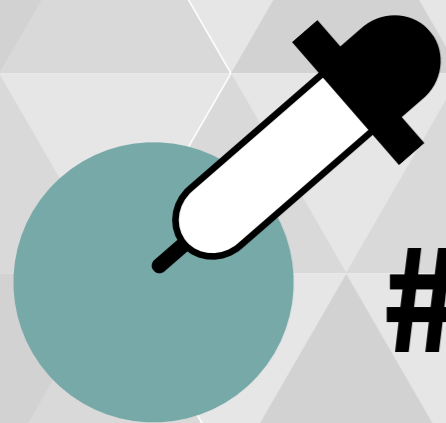
Sass Meisterで見る

お申し込み

お申し込み

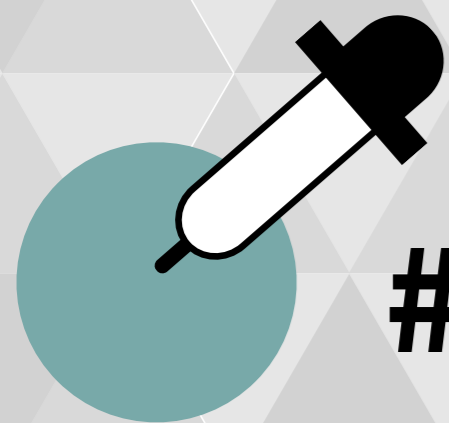


お申し込み

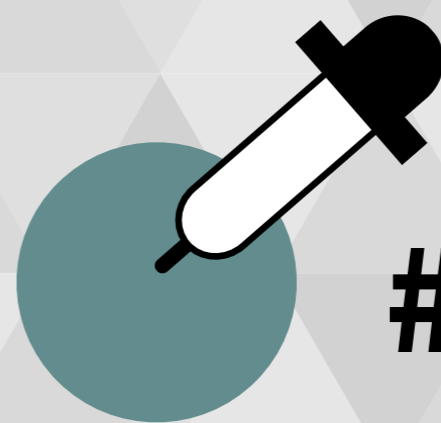


#669999

お申し込み

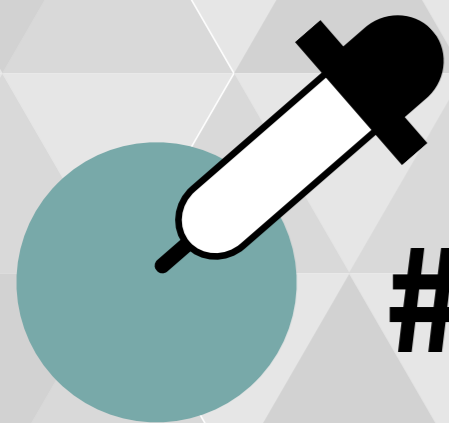


#669999

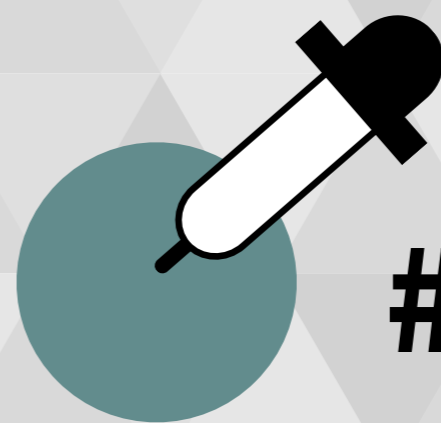


#527A7A

お申し込み



#669999



#527A7A

daken	色を暗くする	<code>darken(色,度数%)</code>
lighten	色を明るくする	<code>lighten(色,度数%)</code>
saturate	彩度を上げる	<code>saturate(色,度数%)</code>
desaturate	彩度を下げる	<code>desaturate(色,度数%)</code>
adjust-hue	色相をずらす	<code>adjust-hue(色,度数deg)</code>

変数を使うともっと便利！

```
$key_color: #669999;

h1 {
  background-color: darken($key_color, 10%);
}
h2 {
  border-left: 5px solid darken($key_color, 5%);
}
h3 {
  border-bottom: 1px solid
    saturate(darken($key_color, 20%), 10%);
}
```

Sass Meisterで見る

CSSに変換すると


```
h1 {  
  background-color: #527a7a;  
}  
h2 {  
  border-left: 5px solid #5c8a8a;  
}  
h3 {  
  border-bottom: 1px solid #366363;  
}
```

h1



テキスト

h2



テキスト

h3

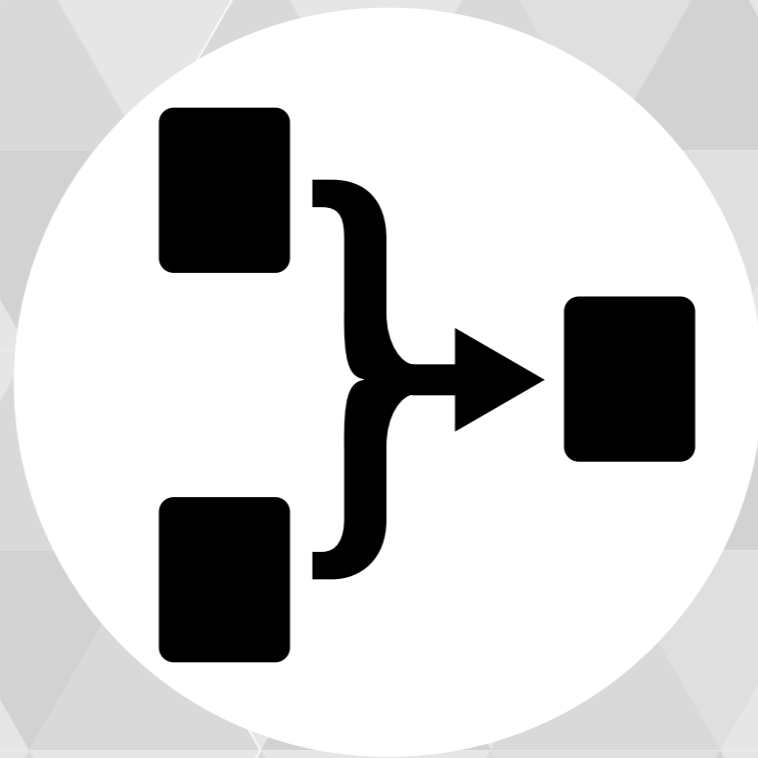


テキスト

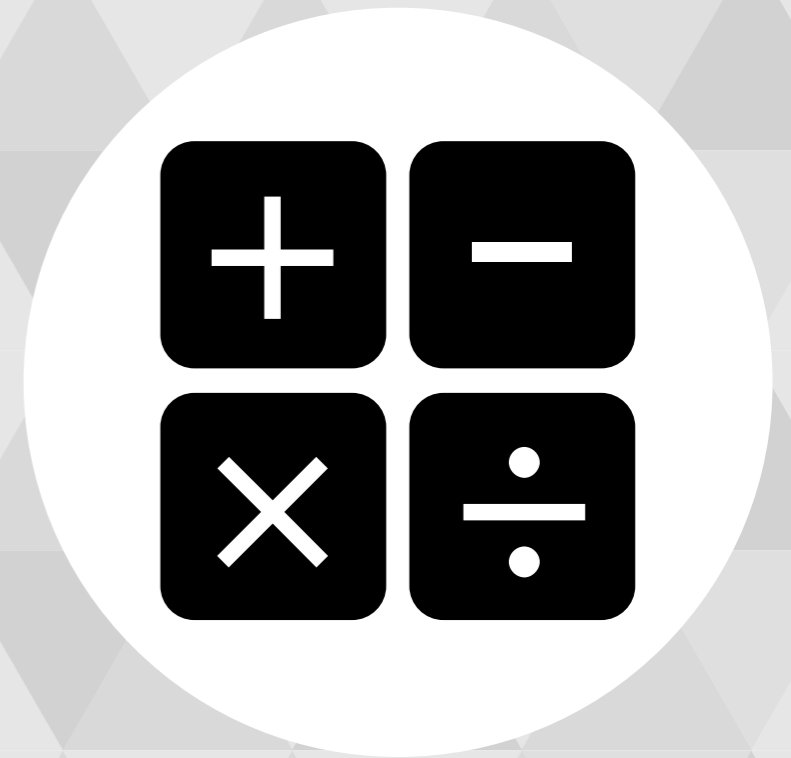
今日のお話



入力の
手間省く



CSSを
まとめる



計算
できる

他にも

- 一行コメント
 - //～
- 条件分岐
 - @if～、 @else～
- ループ
 - @for、 @each～

他にも

ここから

未公開スライド！



最初は難しいかもしれませんが。

でも、覚えるとかなりの手数を減らせます！

基本的なコトを覚えるまでは

「こんなこともできるのか」程度に留めて

基本に集中してくださいね。

● @ror、 @each~

ここが大変CSS

PHPやJSみたいな
一行コメントを書きたい

Sassでは一行コメント書けます

発表時カット
スライド

```
// グローバルナビゲーション
```

```
#gNav {
```

```
  li {
```

```
    $item_number : 4; // 個数を定義
```

```
    width: 960px / $item_number;
```

```
  }
```

```
}
```

Sass Meisterで見る

CSSに変換すると

発表時カット
スライド

```
// グローバルナビゲーション
#gNav {
  li {
    $item_number : 4; // 個数を定義
    width: 960px / $item_number;
  }
}
```

CSSに変換すると

発表時カット
スライド

```
// グローバルナビゲーション
#gNav {
  li {
    $item_number : 4; // 個数を定義
    width: 960px / $item_number;
  }
}
```

//より後ろは、変換後消える

検証に一時的にスタイルをOFFにしたいというとき便利です。

SublimeText+Emmetだと、WindowsならCtrl+/,
MacならCmd+/, でカーソルが行の中にあっても先頭にコメントを
付けたり、消したりしてくれます。個人的にかなり便利。

CSSではこうなる

発表時カット
スライド

```
#gNav li{  
  width: 240px;  
}
```

ここが大変CSS

ちよつと融通効いた
コード書けないの？

// 背景色からコントラストの高い文字色のどちらかを付ける

```
@mixin textContrast($color) {  
  background-color: $color;  
  @if( lightness($color) > 50% ) {  
    color: #000;  
  } @else {  
    color: #FFF;  
  }  
}
```

```
p {  
  @include textContrast(#333);  
}  
ul {  
  @include textContrast(#EFEFEF);  
}
```

発表時カット
スライド

Sass Meisterで見る

```
@mixin textContrast($color) {  
  background-color: $color;  
  @if( lightness($color) > 50% ) {  
    color: #000;  
  } @else {  
    color: #FFF;  
  }  
}
```

lightness(カラーコード)

色の明るさを数値化する関数

0%

暗

100%

明



明るさが50%未満？

明るさが50%未満？

明るさが50%未満？

YES

明るさが50%未満？

NO

YES

明るさが50%未満？

NO

TRUE

明るさが50%未満？

FALSE

TRUE

`$color: #FFF`

`$color: #000`

40%

発表時カット
スライド

明るさが50%未満？

FALSE

TRUE

`$color: #FFF`

`$color: #000`

明るさが50%未満？

FALSE

TRUE

40%

`$color: #FFF`

`$color: #000`

80%

明るさが50%未満？

FALSE

TRUE

40%

`$color: #FFF`

`$color: #000`

明るさが50%未満？

FALSE

TRUE

40%

`$color: #FFF`

80%

`$color: #000`

明るさ50%未満？

```
@mixin light($color) {  
  background-color: $color;  
  @if( lightness($color) > 50% ) {  
    color: #000;  
  } @else {  
    color: #FFF;  
  }  
}
```

明るさ50%未満？

```
@mixin light($color) {  
  background-color: $color;  
  @if( lightness($color) > 50% ) {  
    color: #000;  
  } @else {  
    color: #FFF;  
  }  
}
```

TRUEの場合

明るさ50%未満？

```
@mixin dark($color) {  
  background-color: $color;  
  @if( lightness($color) > 50% ) {  
    color: #000;  
  } @else {  
    color: #FFF;  
  }  
}
```

TRUEの場合

FALSEの場合

// 背景色からコントラストの高い文字色のどちらかを付ける

```
@mixin textContrast($color) {  
  background-color: $color;  
  @if( lightness($color) > 50% ) {  
    color: #000;  
  } @else {  
    color: #FFF;  
  }  
}  
  
p {  
  @include textContrast(#333);  
}  
  
ul {  
  @include textContrast(#EFEFEF);  
}
```

発表時カット
スライド

```
p {  
  background-color: #333333;  
  color: #FFF;  
}  
ul {  
  background-color: #efefef;  
  color: #000;  
}
```

ここが大変CSS

同じようなコードを
書くのが面倒くさい

こんなコード書いたことありませんか？

発表時カット
スライド



クラス名でmargin
制御しよう

こんなコード書いたことありませんか？

発表時カット
スライド

```
.mb10 {  
  margin-bottom: 10px;  
}  
.mb20 {  
  margin-bottom: 20px;  
}  
.mb30 {  
  margin-bottom: 30px;  
}  
.mb40 {  
  margin-bottom: 40px;  
}  
.mb50 {  
  margin-bottom: 50px;  
}
```

セレクトタの中に数字
数字は値にも使われてる

何回も同じこと書いているような気がしてイヤだ！

Sassならこう書ける

発表時カット
スライド

$\$i$ が50未満の間 繰り返す

```
 $\$i$  : 10;  
@while  $\$i$  <= 50 {  
  .mb#{$i} {  
    margin-bottom:  $\$i$  + px;  
  }  
   $\$i$  :  $\$i$  + 10;  
}
```

Sass Meisterで見る

`$i`に10を代入

`$i`は50以上？

NO

YES

```
.mb#{$i} {  
  margin-bottom: $i + px;  
}
```

`$i`に10プラス

戻る

$\$i$ に10を代入

10

$\$i$ は50以上？

NO

YES

```
.mb#{$i} {  
  margin-bottom: $i + px;  
}
```

$\$i$ に10プラス

戻る

$\$i$ に10を代入

$\$i$ は50以上?

NO

YES

10

```
.mb#{$i} {  
  margin-bottom: $i + px;  
}
```

$\$i$ に10プラス

戻る

\$iに10を代入

#{変数}

セレクターやプロパティに直接変数を書こうとすると、意図通りの結果にならないことがあります。そのため、変数を#{ }で囲みます。

```
.mb#{ $i } {  
  margin-bottom: $i + px;  
}
```

\$iに10を代入する

戻る

$\$i$ に10を代入

$\$i$ は50以上？

NO

YES

10

```
.mb#{$i} {  
  margin-bottom: $i + px;  
}
```

$\$i$ に10プラス

戻る

$\$i$ に10を代入

$\$i$ が10のとき

```
.mb10 {  
  margin-bottom: 10px;  
}
```

$\$i$ は50以上?

NO

YES

```
.mb#{ $\$i$ } {  
  margin-bottom:  $\$i$  + px;  
}
```

10

$\$i$ に10プラス

戻る

$\$i$ に10を代入

$\$i$ が10のとき

```
.mb10 {  
  margin-bottom: 10px;  
}
```

$\$i$ は50以上?

NO

YES

```
.mb#{ $\$i$ } {  
  margin-bottom:  $\$i$  + px;  
}
```

20

$\$i$ に10プラス

戻る

$\$i$ に10を代入

$\$i$ は50以上?

NO

YES

```
.mb#{$i} {  
  margin-bottom: $i + px;  
}
```

$\$i$ に10プラス

戻る

$\$i$ が10のとき

```
.mb10 {  
  margin-bottom: 10px;  
}
```

$\$i$ が20のとき

```
.mb20 {  
  margin-bottom: 20px;  
}
```

$\$i$ が30のとき

```
.mb30 {  
  margin-bottom: 30px;  
}  
:  
:
```

$\$i$ が50のとき

```
.mb50 {  
  margin-bottom: 50px;  
}
```

$\$i$ に10を代入

$\$i$ は50以上?

NO

YES

```
.mb#{$i} {  
  margin-bottom: $i + px;  
}
```

$\$i$ に10プラス

戻る

$\$i$ が10のとき

```
.mb10 {  
  margin-bottom: 10px;  
}
```

$\$i$ が20のとき

```
.mb20 {  
  margin-bottom: 20px;  
}
```

$\$i$ が30のとき

```
.mb30 {  
  margin-bottom: 30px;  
}  
:  
:
```

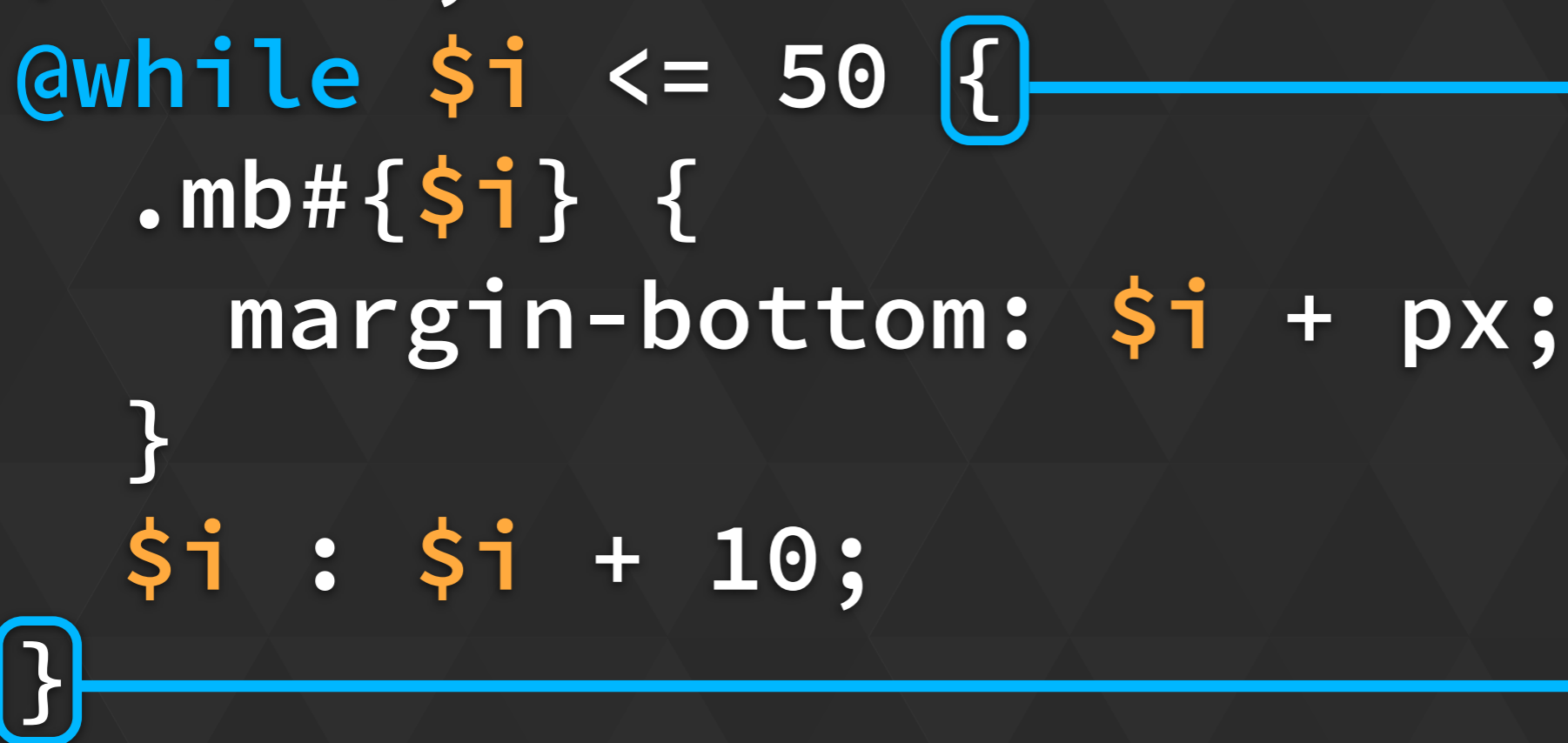
$\$i$ が50のとき

```
.mb50 {  
  margin-bottom: 50px;  
}
```

Sassならこう書ける

発表時カット
スライド

```
$i : 10;  
@while $i <= 50 {  
  .mb#{ $i } {  
    margin-bottom: $i + px;  
  }  
  $i : $i + 10;  
}
```



CSSに変換するところなる

発表時カット
スライド

```
.mb10 {  
  margin-bottom: 10px;  
}  
.mb20 {  
  margin-bottom: 20px;  
}  
.mb30 {  
  margin-bottom: 30px;  
}  
.mb40 {  
  margin-bottom: 40px;  
}  
.mb50 {  
  margin-bottom: 50px;  
}
```

CSSに変換するところなる

発表時カット
スライド

```
.mb10 {  
  margin-bottom: 10px;  
}
```

```
.mb20 {  
  margin-bottom: 20px;  
}
```

```
.mb30 {  
  margin-bottom: 30px;  
}
```


```
.mb40 {  
  margin-bottom: 40px;  
}
```

```
.mb50 {  
  margin-bottom: 50px;  
}
```

未公開スライド
ここまで！

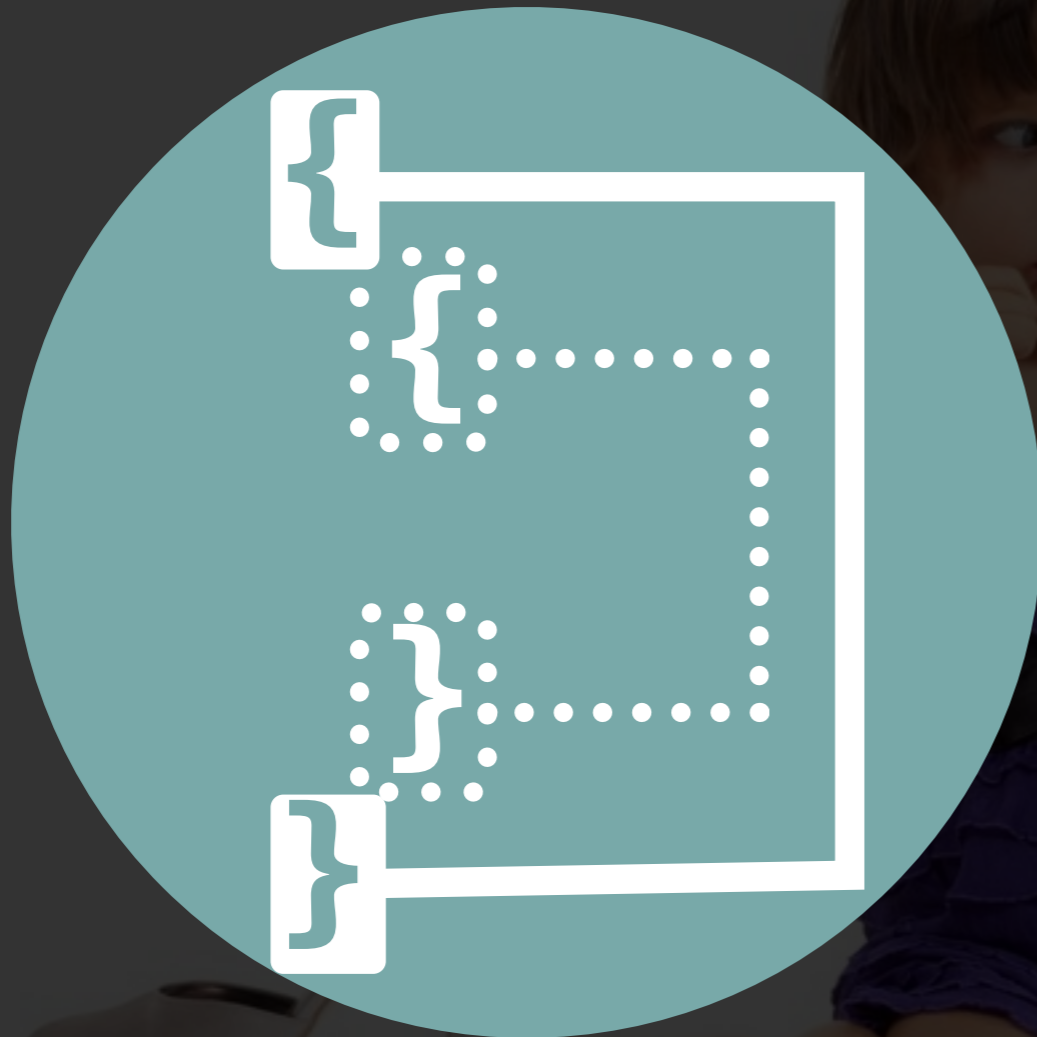
ありがとうございました！



A young child with short brown hair, wearing a purple ruffled dress and a pink necklace, is sitting on a light-colored surface. The child is holding a white high-heeled shoe in their right hand and looking at it with a curious expression. On the floor around the child are various pieces of adult jewelry: a black makeup palette, a long black pearl necklace, a yellow beaded bracelet, a red bangle, a pink beaded bracelet, and a yellow bangle. The background is a plain, light-colored wall.

いきなり全部やると
つまずく

ネスト



変数



できるところから

SCSS ▾

```
1 ▾ #main {  
2 ▾   h1 {  
3   |   margin: 10px;  
4   }  
5 }
```

Extensions ▾

CSS (expanded ▾)

```
1 ▾ #main h1 {  
2   margin: 10px;  
3 }  
4 |
```

Sassmeister <http://sassmeister.com/>

Sassをオンライン上で試せるWebサービス





THANK YOU

小さなことからコツコツと



しばっち

 <http://blog.1dz.jp/>

 @sivacchi

 sivacchi

 sivacchi@u-copi.com