

# CSSについて

高井ゼミ 1 班

木村勇貴 末松侃大 岡寄允義 奥田廉 吉田陽彩

# 目次

- ▶ ブラウザレンダリング
- ▶ z-indexの例
- ▶ compositeの例
- ▶ CSSのアニメーション例

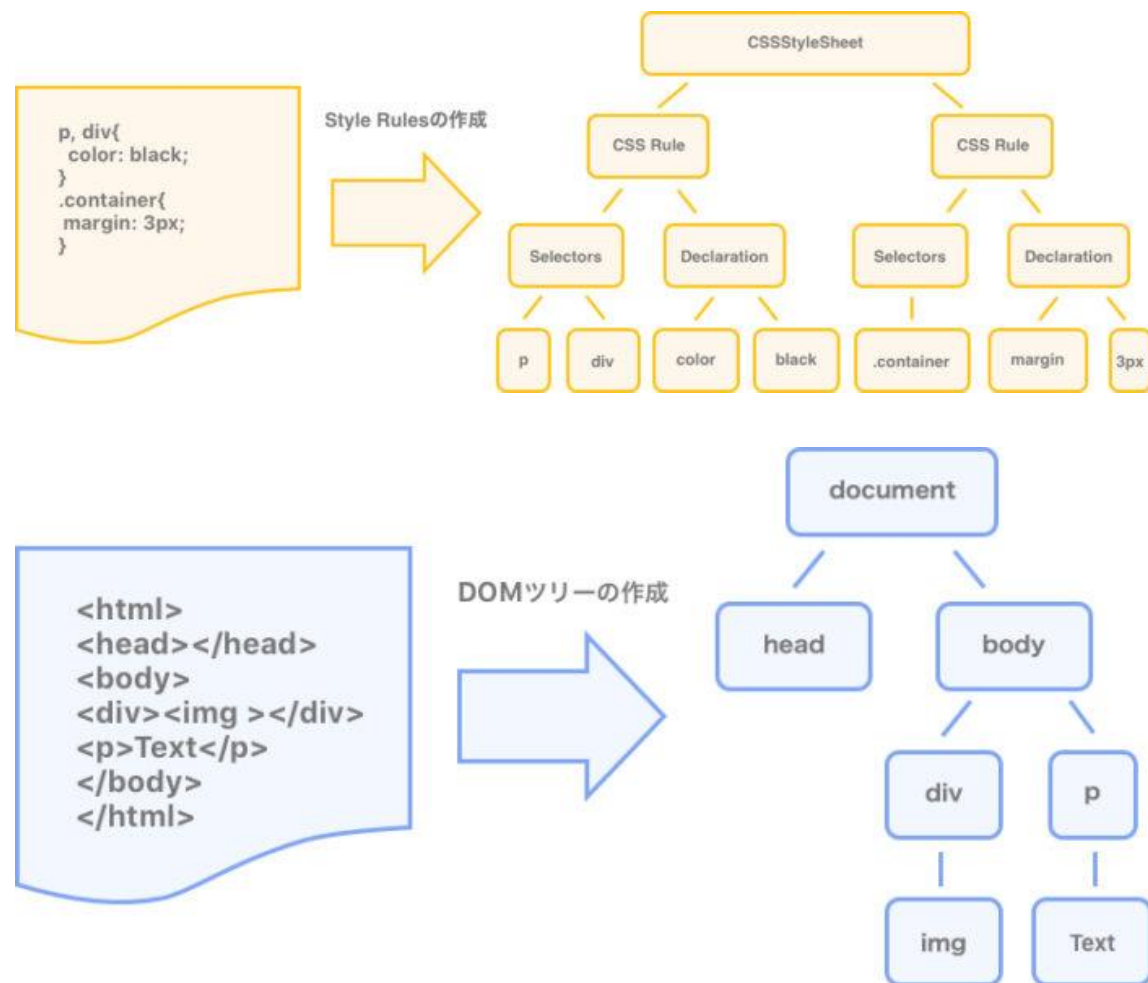
# ブラウザレンダリングとは

- ▶ ブラウザレンダリングは、ウェブブラウザがHTML、CSS、JavaScriptなどのウェブページのリソースを処理して、視覚的なページとしてユーザーに表示するプロセスを指します。
- ▶ 以下の順に処理される



# parse

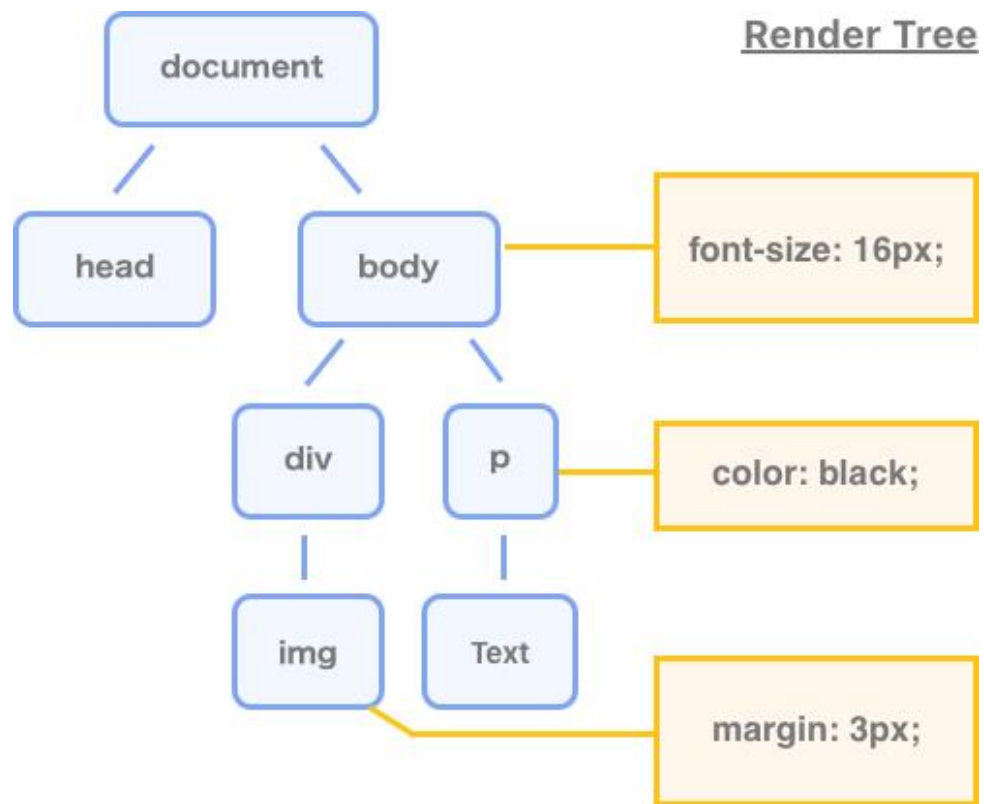
- ▶ CSSとhtmlの解析をし、DOMツリーとスタイルツリーを作成する。
- ▶ なぜツリー型になのか  
というとは色々理由はあるが、機械にとって都合の良い構造であるから。



# style

DOMツリーとスタイルツリーを紐付けする工程

紐づけたものをrender treeという



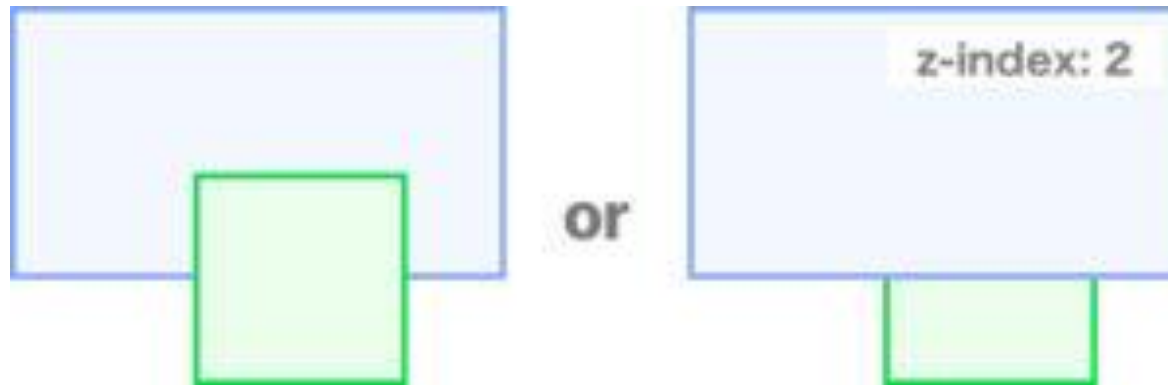
# layout



- ▶ この工程では、それぞれの要素と大きさの計算を行う

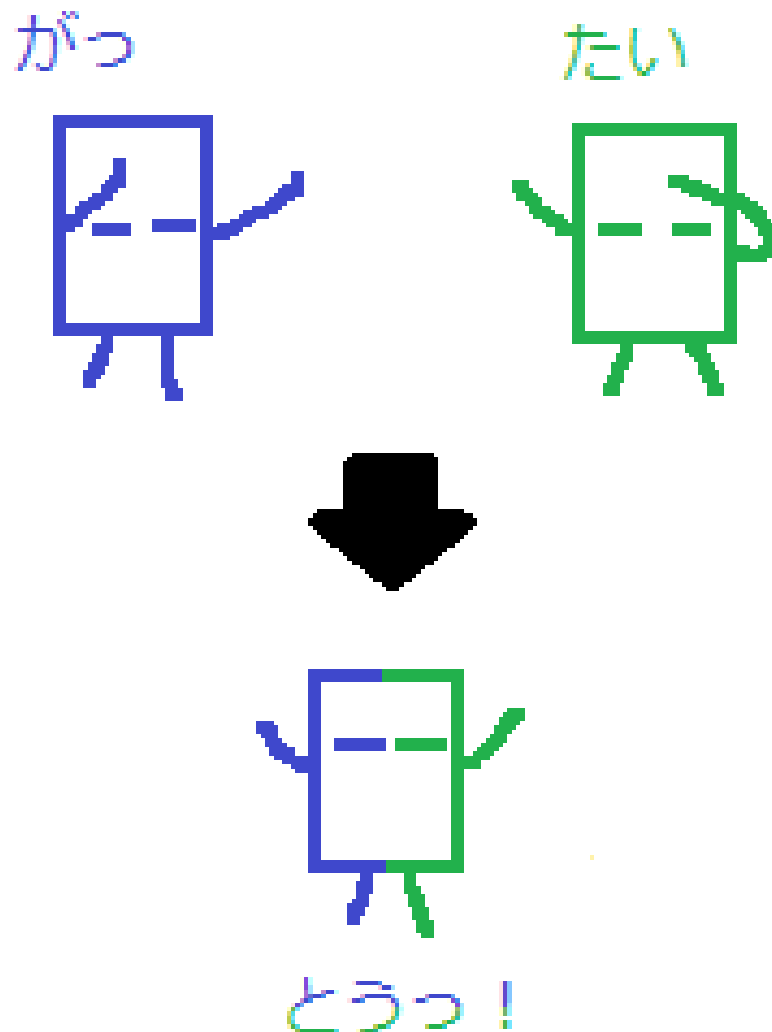
# paint

ペイントの工程では、構成要素をどの優先順位で表示するか決定します。



# composite

- ▶ コンポジットとは、複合、合成という意味。
- ▶ ペイントは「素材を作る」、コンポジットは「素材を組み合わせる」と考えると分かりやすいです。
- ▶ 最終的なページを作り上げる工程





# 目次

- ▶ ブラウザレンダリング
- ▶ z-indexの例
- ▶ compositeの例
- ▶ CSSのアニメーション例

# paint工程で紹介されたz-indexを使ったCSSの例

タッチパッドを動かしてみると、、、



## 補足 要素の配置方法→position+何か

- ▶ positionで要素の配置方法を指定する (static ,relativeなど)



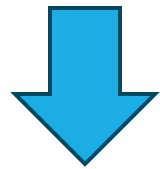
- ▶ top, bottom, left, rightで要素の位置を指定 (auto, pxなど)



# 先程の例ではposition+z-indexを使った

- ▶ position:要素の配置方法を指定する

relative→相対位置で配置、fixed→ウィンドウ内の絶対位置で配置



- ▶ z-index:要素の重なり順を指定する、整数値が大きい方が上に来る

コード例)要素A→z-index: 0; 要素B→z-index: 1; ⇨この場合Aが下、Bが上



# z-indexを使ったCSSの例

上側の水色の部分: position→fixed, z-index→0

下側の白色の部分: position→relative,  
z-index→9999



```
1 <!DOCTYPE html>
2 <head>
3   z-index具体例
4   <style>
5     .z-index_test1 {
6       font-size: 300%;
7       text-align: center; /* 文字中央揃え */
8       padding: 120px 0; /* 余白指定 */
9       background-color: #00BCD4; /* 背景色指定 */
10      color: #fff; /* 文字色指定 */
11      position: fixed; /* 位置指定 */
12      width: 100%; /* 幅指定 */
13      box-sizing: border-box; /* 崩れ防止 */
14      z-index: 0; /* 重なり順指定 */
15      top: 0; /* 位置指定 */
16    }
17    .z-index_test {
18      font-size: 150%;
19      height: 500px; /* 高さ指定 */
20      box-shadow: 0px 2px 3px 0px #000; /* 影指定 */
21      width: 90%; /* 幅指定 */
22      margin: 0 auto; /* 要素中央揃え */
23      background-color: #fff; /* 背景色指定 */
24      position: relative; /* 位置指定 */
25      z-index: 9999; /* 重なり順指定 */
26      margin-top: 265px; /* 位置指定 */
27      padding: 20px; /* 余白指定 */
28      box-sizing: border-box; /* 崩れ防止 */
29    }
30    main {
31      height: 600px; /* 高さ指定 */
32    }
33  </style>
34  </head>
35  <main>
36    <div class="z-index_test1">
37      z-index
38    </div>
39    <div class="z-index_test">
40      z-indexとは要素の重なり順を指定するCSSのプロパティのことです。
41    </div>
42  </main>
43
44  </html>
```

# 目次

- ▶ ブラウザレンダリング

- ▶ z-indexの例

- ▶ compositeの例

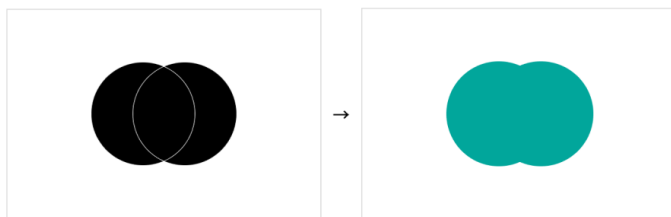
- ▶ アニメーション例

# Composite

## Add

それぞれの領域を足し合わせる

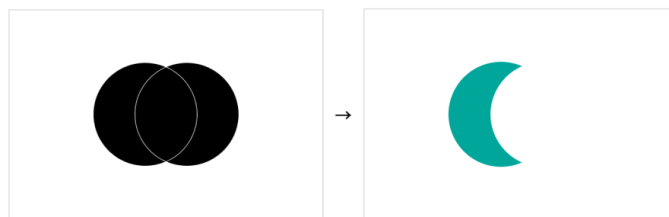
add



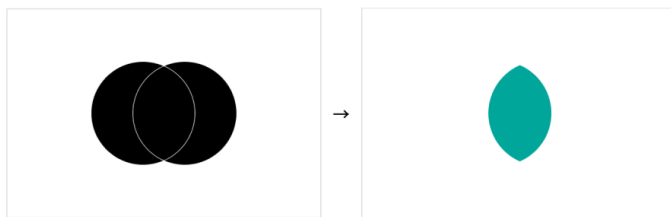
## Subtract

1つ目のマスク画像から2つ目のマスク画像を減算する

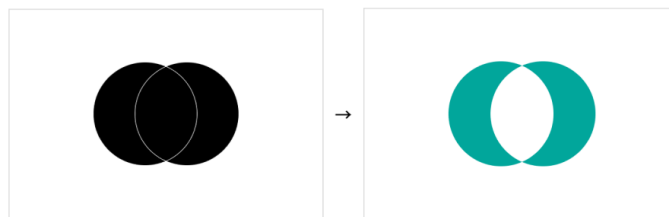
subtract



intersect



exclude



## Intersect

マスク画像の塗りが重なる部分をマスク領域とする

## Exclude

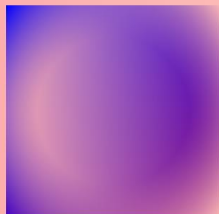
マスク画像の塗りが重なる部分以外をマスク領域とする

# CSSの例 : composite

赤枠→背景、linear-gradient→左から右にグラ  
デーション

緑枠→前面、四角形、左から右にグラデーション

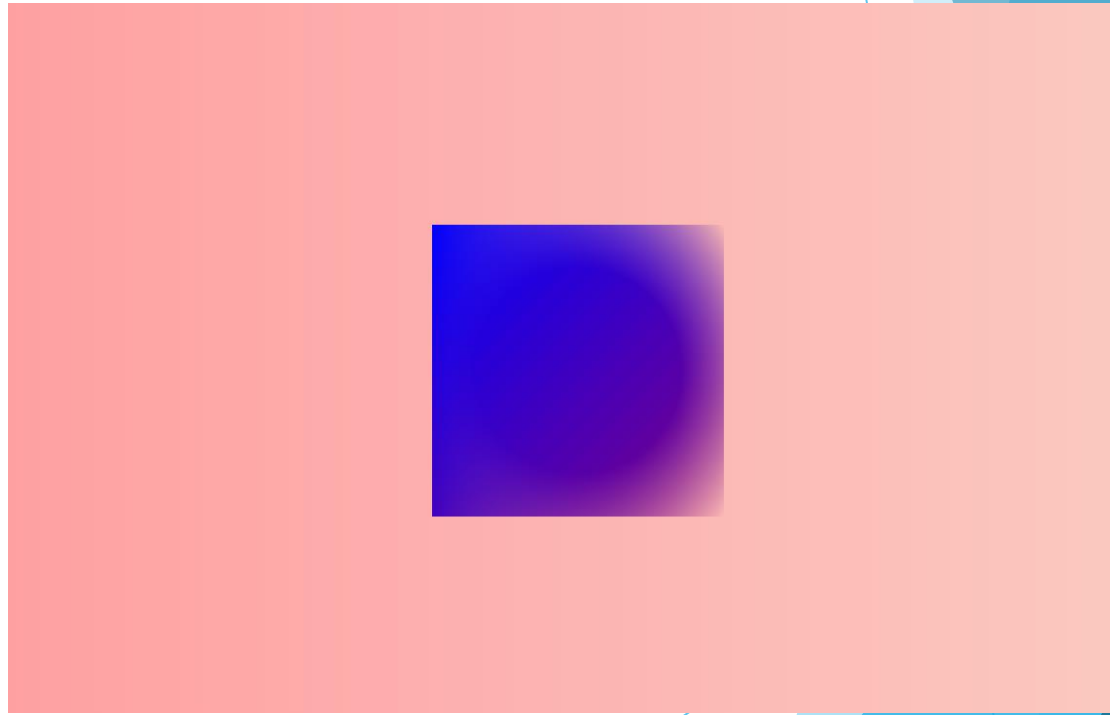
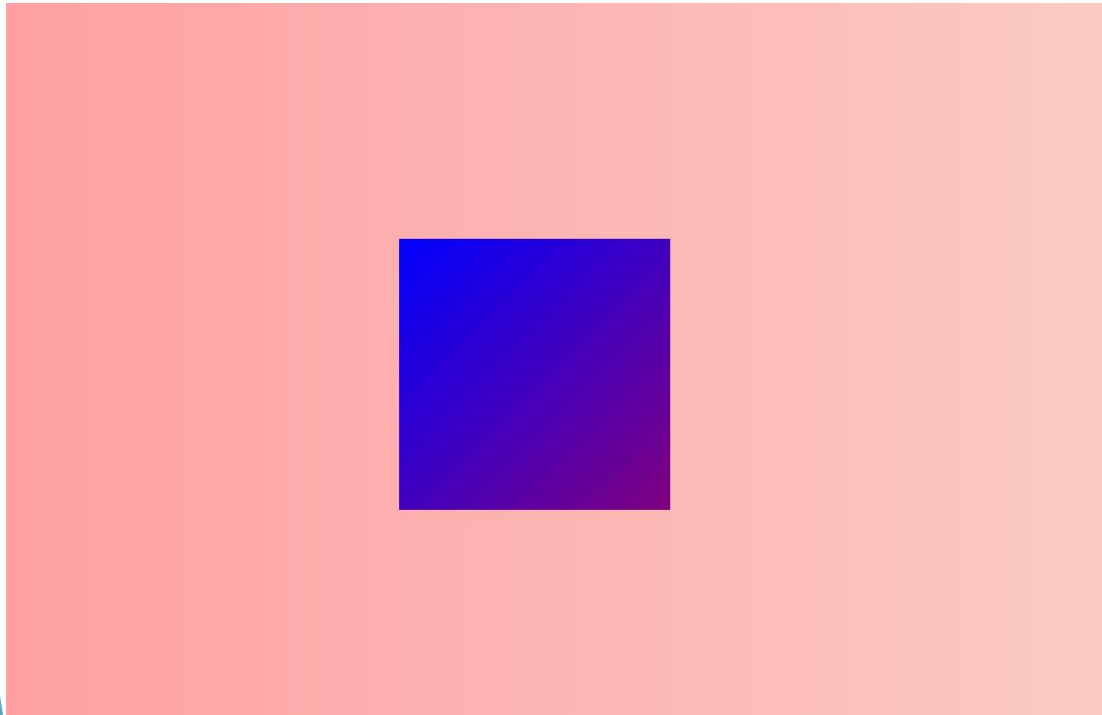
青枠→前面と背景と円の複合、全面の円形  
Mask-composite:exclude  
↳重なった部分を除外しその部分を透明に



```
composit2.html > html > head > style > .masked-box
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>CSS Mask Composite 例</title>
7   <style>
8     body {
9       margin: 0;
10      display: flex;
11      justify-content: center;
12      align-items: center;
13      height: 100vh;
14      background: linear-gradient(90deg, #ff9a9e, #fad0c4);
15    }
16
17    .masked-box {
18      width: 300px;
19      height: 300px;
20      background: linear-gradient(135deg, blue, purple);
21
22      /* マスクの設定 */
23      mask-image: radial-gradient(circle, black 50%, transparent 100%),
24                 linear-gradient(to right, black, transparent);
25      mask-composite: exclude;
26      -webkit-mask-image: radial-gradient(circle, black 50%, transparent 100%),
27                         linear-gradient(to right, black, transparent);
28      -webkit-mask-composite: xor; /* Webkitブラウザ対応 */
29    }
30  </style>
31 </head>
32 <body>
```



# Composite



# 目次

- ▶ ブラウザレンダリング
- ▶ z-indexの例
- ▶ composite具体例
- ▶ CSSのアニメーション例

# CSSのアニメーション例

▶ CSSは主にデザインの役割

→ それでは動きはJavaScriptのみであるか？

**CSS3がリリースされてから様々な機能が追加された**

ex. 動的アクション、グラデーション、透明度など

# CSSのアニメーション例

## ▶ CSSを用いてアニメーションを表示させる意義

→ 前回の発表（下の表は再掲）

CSS	javascript
<ul style="list-style-type: none"><li>・コンポジットスレッドを使っているため負担が少ない</li><li>・ javascriptよりコードが簡単</li><li>・ メンテナンスが簡単</li><li>・ javascriptよりできることが少ない</li></ul>	<ul style="list-style-type: none"><li>・ メインスレッドを使用しているため、負担が重い</li><li>・ コードがCSSより複雑</li><li>・ メンテナンスがCSSより難しい</li><li>・ CSSよりできることが多い</li></ul>

# CSSのアニメーション例

- ▶ CSSでは主にプロパティ（ex. Background-color , margin）によってフォントや行間などのデザインを行う

→CSSでアニメーションを用いるにあたり、使用するプロパティは「**animation**」

# CSSのアニメーション例

## ▶ キーフレーム

→アニメーションを指定する対象の、ある時点の状態を記述

## ▶ 例えば対象がページの右端から左端に移動する場合...

アニメーション開始前の状態	←	<code>@keyframe slide {</code>
		<code>0% { margin-left 0 }</code>
アニメーション終了後の状態	←	<code>100% { margin-right 0 }</code>

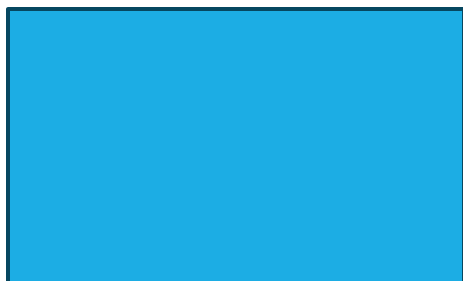
# CSSのアニメーション例

## ▶ animationのプロパティとして

animation-name	: アニメーションの名前
animation-duration	: アニメーションの開始～完了時間
animation-timing-function	: アニメーションの変化速度
animation-delay	: アニメーションの開始タイミング
animation-iteration-count	: アニメーションの繰り返し設定
animation-direction	: アニメーションの向き

# CSSのアニメーション例（具体例①）

## ▶ フェードイン

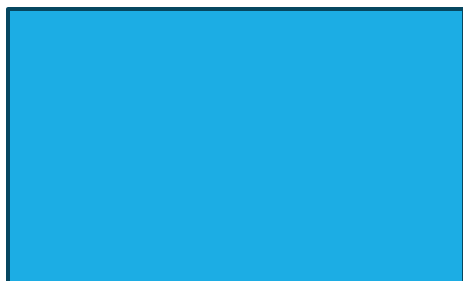


```
.anim-box.fadein.is-animated {  
  animation: fadeIn 0.7s cubic-bezier(0.33, 1, 0.68, 1) forwards;  
}  
  
@keyframes fadeIn {  
  0% {  
    opacity: 0;  
  }  
  100% {  
    opacity: 1;  
  }  
}
```



# CSSのアニメーション例（具体例②）

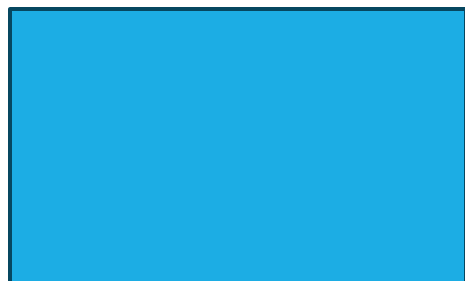
## ▶ スライドイン



```
.anim-box.slidein.is-animated {  
  animation: slideIn 1s cubic-bezier(0.25, 1, 0.5, 1) forwards;  
}  
  
@keyframes slideIn {  
  0% {  
    transform: translateX(180px);  
    opacity: 0;  
  }  
  100% {  
    transform: translateX(0);  
  }  
  40%, 100% {  
    opacity: 1;  
  }  
}
```

# CSSのアニメーション例（具体例③）

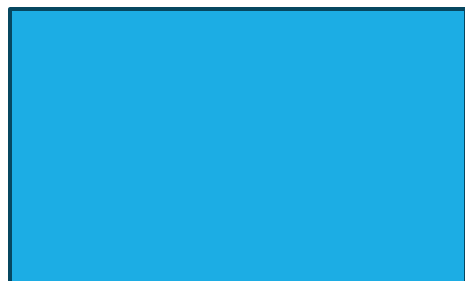
## ▶ フェードアップ



```
.anim-box.fadeup.is-animated {  
  animation: fadeup 1s cubic-bezier(0.33, 1, 0.68, 1) forwards;  
}  
  
@keyframes fadeup {  
  0% {  
    transform: translateY(30px);  
    opacity: 0;  
  }  
  80% {  
    opacity: 1;  
  }  
  100% {  
    opacity: 1;  
    transform: translateY(0);  
  }  
}
```

# CSSのアニメーション例（具体例④）

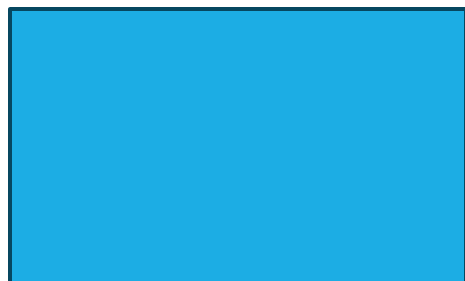
## ▶ ズームイン



```
.anim-box.zoomin.is-animated {  
  animation: zoomIn 0.8s cubic-bezier(0.25, 1, 0.5, 1) forwards;  
}  
  
@keyframes zoomIn {  
  0% {  
    transform: scale(0.8);  
    opacity: 0;  
  }  
  100% {  
    opacity: 1;  
    transform: scale(1);  
  }  
}
```

# CSSのアニメーション例（具体例⑤）

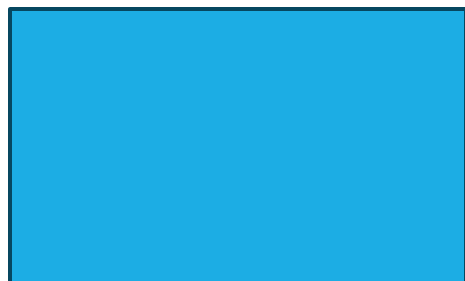
## ▶ ポップアップ



```
.anim-box.popup.is-animated {  
  animation: popup 0.6s cubic-bezier(0.22, 1, 0.36, 1) forwards;  
}  
  
@keyframes popup {  
  0% {  
    transform: translateY(40px) scale(0.8);  
    opacity: 0;  
  }  
  100% {  
    transform: translateY(0) scale(1.0);  
  }  
  80%, 100% {  
    opacity: 1;  
  }  
}
```

# CSSのアニメーション例（具体例⑦）

## ▶ 回転表示



```
.kurukuru.is-animated {  
  animation: kurukuru 1.4s ease-out;  
  opacity: 1;  
}  
@keyframes kurukuru {  
  0% {  
    transform: rotateY(0) translateY(40px);  
    opacity: 0;  
  }  
  100% {  
    transform: rotateY(360deg) translateY(0);  
    opacity: 1;  
  }  
}
```

# CSSのアニメーション例

- ▶ 以上のアニメーションはあくまでも一例

**これらのアニメーションを用いた動きでどこまでできるのか？**

# CSSのアニメーション例

- ▶ <https://codepen.io/iamlark/pen/jYzYJg>

# 参考文献

- ▶ [https://www.sejuku.net/blog/55061#index\\_id3](https://www.sejuku.net/blog/55061#index_id3)
- ▶ <https://codepen.io/iamlark/pen/jYzYJg>
- ▶ <https://web-design-textbook.com/css-middle/#レイアウトの仕組み>
- ▶ <https://qiita.com/na-suzuki/items/037592b14ee0b2d44278>
- ▶ [https://saruwakakun.com/html-css/basic/relative-absolute-fixed#google\\_vignette](https://saruwakakun.com/html-css/basic/relative-absolute-fixed#google_vignette)
- ▶ <https://b-risk.jp/blog/2021/01/anim-reference/>
- ▶ [フロントエンジニアなら知っておきたいブラウザレンダリングの仕組みをわかりやすく解説！ | Tech Blog](#)
- ▶ [mask-compositeでつくるCSSの新しい表現方法 - ICS MEDIA](#)