

CSS3 - 追加されたプロパティ

CSS3 で新たに追加された他のプロパティを学習します

要素のアニメーション

フレキシブルボックスの他に、CSS3 となり新しく追加された便利なプロパティがあります。制作の中でうまく使えば、作業時間が短縮されたりと便利になった部分が沢山ありますので、今回は新たに追加されたプロパティを取り上げます。

animation Flash と同じく、キーフレームでアニメーションを指定する。ループ再生可能。

transtion 特定のアクション (hover) をした場合にのみ動作し、その後逆の動作を行う。ループ再生不可。

transition:

CSS

transition-property

再生時間

transition-duration

トランジション

transition-timing-function

開始時間

transition-delay

要素の状態間のトランジションを作成します。

値	意味
transition-property	トランジション効果を適用する CSS プロパティを指定する (初期値 all)
transition-duration	アニメーションが終了するまでにかかる時間を秒で指定 (初期値 0s)
transition-timing-function	アニメーションの中間値の算出方法 (加速度 初期値 ease)
transition-delay	アニメーションの要求が出てから実行されるまでの時間 (初期値 0s)

transition-timing-function の値

ease (初期値)	開始と終了が滑らかに動く
ease-in	トランジションの最初、ゆっくりと動き始まる
ease-out	トランジションの最後、ゆっくりと動き終わる
ease-in-out	ゆっくりと始まり加速、その後ゆっくりと減速する
linear	一定の加速度でアニメーションを行う
step-start	トランジション開始時に終了値までジャンプする
step-end	トランジション経過後に終了値までジャンプする
steps(step 数 ,start/end)	トランジション間を指定したステップ数で動かす
cubic-bezier(x1,y1,x2,y2)	加速度をカスタムで作成する (数値: 0~1)

animation:

キーフレーム

animation-name

再生時間

animation-duration

トランジション

animation-timing-function

開始時間

animation-delay

再生回数

animation-iteration-count

再生方向

animation-direction

再生後の CSS

animation-fill-mode

判定

animation-play-state

要素に対してアニメーションを実装する

状態をトリガーとして動く transition と違い、animation は指定した要素をキーフレームを使用してアニメーションを指定することが可能です。animation-duration、animation-timing-function、animation-delay に関しては transition と同様の設定をすることが出来ます。

値	意味
animation-name	再生するアニメーションのキーフレームを指定（初期値 none）
animation-iteration-count	アニメーションの再生回数を指定（初期値 1）
animation-direction	逆方向のアニメーションの指定（初期値 normal）
animation-fill-mode	アニメーション終了時の要素のスタイルの指定（初期値 none）
animation-play-state	アニメーションが実行中かどうかの状態を判定（初期値 running）

animation-iteration-count	
数値 または infinite	再生回数を数値で指定する。常に再生させたい場合は infinite を使用

animation-direction	
normal（初期値）	順方向のアニメーションを毎回実行する
alternate	逆方向のアニメーションも毎回実行する
reverse	アニメーションの再生を毎回逆方向に実行する
alternate-reverse	初回アニメーションを逆方向に実行し、次は順方向に実行、繰り返す

animation-fill-mode	
none（初期値）	アニメーションで実行されたスタイルは維持しない
forwards	アニメーションの最後に適用されたスタイルを維持する
backwards	再生後、最初のキーフレームのスタイルを維持する
both	再生後、最後のキーフレームのスタイルを維持し、delay 時は最初のキーフレームのスタイルが適用される

animation-play-state	
running または paused	アニメーションを一時停止する / アニメーションを実行する

キーフレームの設定

transition は状態間アニメーションですが、animation プロパティはキーフレームを指定し、自分でアニメーションを作成する必要があります。キーフレームの作成は @keyframes を使用し作成します。最初キーフレームは 0%(from) から 100%(to) の中で指定します。詳しくは下記コードを確認してください。

exsample - HTML

```
<div class="box">
  box area
</div>
```

exsample - CSS

```
.box{
    position: relative;
    width: 250px;
    height: 250px;
    animation: 3s ease-in-out 0s infinite alternate moveBox;
}

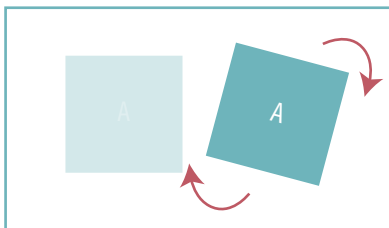
@keyframes moveBox{
    from{
        right: 0;
    }
    50%{
        right: 200px;
    }
    to{
        right: 0;
    }
}
```

キーフレームアニメーションを設定する際に、各キーフレーム内でスタイルを統一しておくことを忘れないようにしましょう。例えば 50% の位置から急に top の指定を書いた場合、意図しない動作になる可能性があります。

transform: 変形値

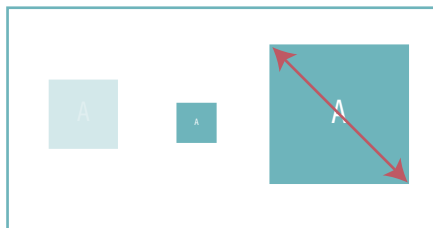
要素を変形する

値	意味
rotate(angle)	要素を原点 (transform-origin) を中心に、時計回りに指定した分回転する
scale(sx[,sy])	幅, 高さの大きさの比率を変更する (1 桁の場合は両方)
scaleX(sx)	幅の大きさの比率を変更する
scaleY(sy)	高さの大きさの比率を変更する
skew(ax[, ay])	X または Y 軸に対して、指定した角度だけ要素が傾斜されます
skewX(angle)	X 軸に対して、指定した角度だけ要素が傾斜されます
skewY(angle)	Y 軸に対して、指定した角度だけ要素が傾斜されます
translate(tx[, ty])	X または Y 軸に沿って平面の移動を行う
translateX(tx)	X 軸に沿って平面の移動を行う
translateY(ty)	Y 軸に沿って平面の移動を行う



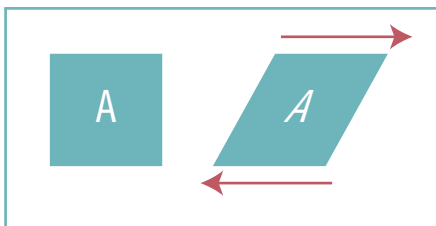
rotate(angle)

角度を指定して回転させる



scale(x [,y])

倍率 (1 = 100%) を指定して、拡大縮小させる



skew(ax [, ay])

X(Y) 軸に対して角度分傾斜させる



translate(tx [, ty])

X(Y) 軸に沿って移動させる

奥行きの設定（立体的な表示）

平面（2D）の操作が出来れば、立体的な表現をすることが可能です。要素を立体的に表示することが可能です。奥行きを持たせる場合、`transform-style` を使い、立体表現に設定し、奥行き（`perspective`）の設定や基準点（`transform-origin`）の設定などを行います。

transform-style: 子要素の空間の設定

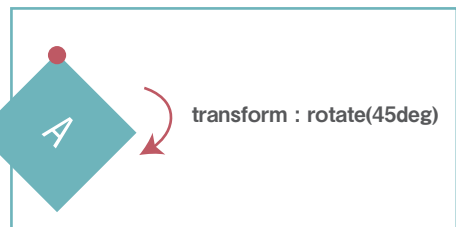
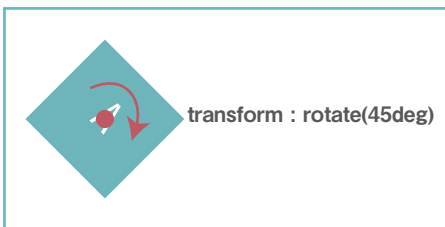
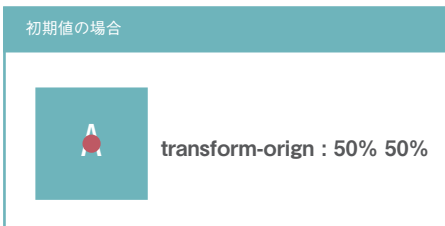
平面か立体か要素の配置方法の設定

値	意味
flat （初期値）	要素の子要素を要素自身の面上（2D）に置くことを示すキーワードです
preserve-3d	要素の子要素を 3D 空間に配置することを示すキーワードです

transform-origin: 横位置 縦位置

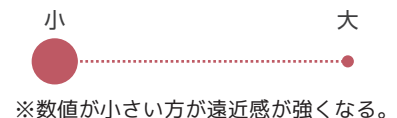
要素の原点の指定

値	意味
50% 50% （初期値）	要素の中心を原点とします left center right top bottom も可能



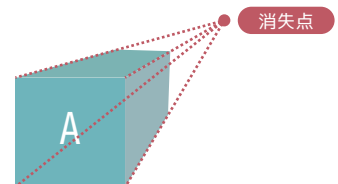
perspective: 数値（初期値：none）

3D 配置された要素の遠近感の設定



perspective-origin: 横位置 [縦位置]

3D 配置された要素の遠近感の消失点の設定



transform3D

要素を立体的に変形する

値	意味
rotate3d(x,y,z,angle)	要素を原点 (transform-origin) を中心に、各軸に合わせて回転する
translate3d(x,y,z)	要素を X,Y,Z 軸に沿って移動させる
scale3d(x,y,z)	要素を X,Y,Z 軸に沿って大きさの倍率を変更する

各 3d プロパティは一括指定の為、任意の軸のみ変更する場合、例えば Z 軸に沿って移動したい場合は translateZ() など、基準軸のプロパティがそれぞれ用意されています。

example - HTML

```

<div class="sampleWrap">
  <div class="boxWrap">
    <div class="front">front</div>
    <div class="side">side</div>
  </div>
</div>

```

example - CSS

```

.sampleWrap{
  width: 100%;
  height: 500px;
  padding: 10px;
  border: solid 1px #333;
}
.boxWrap{
  width: 400px;
  height: 400px;
  padding: 20px;
  transition: 1s ease-in-out;
  transform-origin: center center;
  transform-style: preserve-3d;
}
.boxWrap:hover{
  transform: rotateY(-90deg);
}
.boxWrap div{
  text-align: center;
  font-weight: bold;
  color: #FFF;
  line-height: 50px;
  width: 200px;
  height: 50px;
  position: absolute;
  top:0;
  left:0;
  right:0;
  bottom:0;
  margin: auto;
}

```

CSS 続き

```

.front{
  background:#1abc9c;
  transform: translateX(0) translateY(0)
translateZ(50px);
  transform-origin: center center;
}
.side{
  background: #e74c3c;
  transform-origin: center center;
  transform: translateX(100px) translateZ
(-50px) rotateY(90deg);
}

```