

## HTML 演習

### HTML 演習 No.11 (CSS3)

課題 1 以下の HTML を入力する。(transition を使ったアニメーション)

ファイル名: ensyu11\_1.html

```
<!-- 作成日: 20XX/XX/XX -->
<!-- 作成者: 情報 太郎 -->

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>transition を使ったアニメーション</title>
  <style type="text/css">
    /* 共通設定 */
    div {
      width: 100px;
      height: 100px;
      padding: 1em;
      margin: 2em 2em 4em;
      background: #ad4444;
      color: #fff;
      font-size: 14px;
    }

    /* 右に伸びる */
    .width { transition: width 0.5s ease-out; }
    .width:hover { width: 400px; }

    /* 四角から丸に */
    .circle { transition: border-radius 0.2s; }
    .circle:hover { border-radius: 50%; }

    /* ホバーして1秒後にズームアップ */
    .scale { transition: transform 0.5s ease-in-out 1s; }
    .scale:hover { transform: scale(1.5); }
  </style>
</head>
<body>
  <h1>ホバーアニメーション</h1>
  <p>下の 3 つの要素をホバーしてみよう！</p>
```

0.5 秒かけて width を変化させる

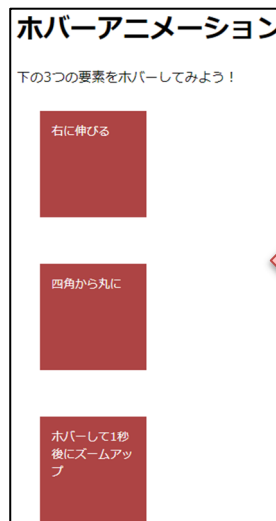
0.2 秒かけて角丸部分を変化させる

1 秒おいて 0.5 秒かけて変形 (1.5 倍拡大) させる

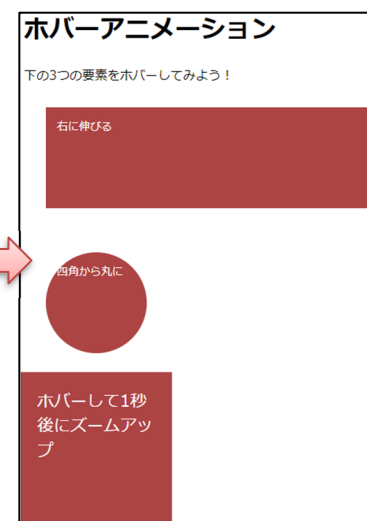
```
<div class="width">右に伸びる</div>
<div class="circle">四角から丸に</div>
<div class="scale">ホバーして1秒後にズームアップ</div>
</body>
</html>
```

[実行結果]

通常時



ホバー時



POINT

```
.scale { transition: transform 0.5s ease-in-out 1s; }
```

① ② ③ ④

① 変化させるスタイル ②変化にかかる時間 ③イージング ④遅延時間

③イージングとは、アニメーションの速度に変化をつけること。

初期値は ease(開始・終了をなめらかに), ease-in(ゆっくり開始),

ease-out(ゆっくり終了), ease-in-out(開始・終了をゆっくり),

linear(等速)がある。

## HTML 演習

### HTML 演習 No.11 (CSS3)

課題 2 以下の HTML を入力する。(animation を使ったアニメーション)

ファイル名: ensyu11\_2.html

```
<!-- 作成日: 20XX/XX/XX -->
<!-- 作成者: 情報 太郎 -->

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>animation を使ったアニメーション</title>
    <style type="text/css">
      /*共通設定*/
      div {
        width: 100px;
        height: 100px;
        padding: 1em;
        margin: 2em 2em 4em;
        background: #ad4444;
        color: #fff;
        font-size: 14px;
      }

      /*ジグザグに動くアニメーション*/
      .sample1 { animation: zigzag 2s ease 0s 3 normal; }
      /*zigzag のキーフレームを定義*/
      @keyframes zigzag {
        0% {transform: translate(0, 0);}
        50% {transform: translate(50px, 50px);}
        100% {transform: translate(100px, 0px);}
      }

      /*大きさと色が変わるアニメーション*/
      .sample2 { animation: scale-and-color 4s ease 0s infinite
alternate; }
      /*scale-and-color のキーフレームを定義*/
      @keyframes scale-and-color {
        0% {background: #ad4444; transform: scale(1);}
        50% {background: #3946ad; transform: scale(1.5);}
        100% {background: #4dad48; transform: scale(1);}
      }
    </style>
  </head>
  <body>
    <h1>キーフレームアニメーション</h1>
    <p>サンプル 1</p>
    <div class="sample1">ジグザグに動く</div>

    <p>サンプル 2</p>
    <div class="sample2">イージングも逆再生される</div>

    <p>サンプル 3、4</p>
    
    
  </body>
</html>
```

2 秒かけて再生、3 回繰り返す

座標を変えて div の  
位置を動かしている

4 秒再生、4 秒逆再生の繰り返し

```
/*ローディングアニメーション*/
img {
  width: 100px;
  margin-right: 2em;
}
.sample3 { animation: loading 1s linear 0s infinite; }
/*キーフレーム*/
@keyframes loading {
  0% {transform: rotate(0deg);}
  30% {transform: rotate(180deg);}
  100% {transform: rotate(360deg);}
}
/*鼓動しているようなアニメーション*/
.sample4 { animation: heart 0.8s ease 0s infinite normal; }
/*キーフレーム*/
@keyframes heart {
  0% {transform: scale(1);}
  8% {transform: scale(1);}
  15% {transform: scale(1.1);}
  100% {transform: scale(1);}
}
</style>
</head>
<body>
  <h1>キーフレームアニメーション</h1>
  <p>サンプル 1</p>
  <div class="sample1">ジグザグに動く</div>

  <p>サンプル 2</p>
  <div class="sample2">イージングも逆再生される</div>

  <p>サンプル 3、4</p>
  
  
</body>
</html>
```

1 秒で 1 回転を繰り返す

前半 30% で 180° 回転  
後半 70% で 180° 回転  
(前半速く、後半遅い)

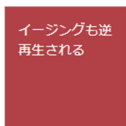
急に大きくなり、  
ゆっくり小さくなる

## キーフレームアニメーション

サンプル1



サンプル2



サンプル3、4



サンプル 1 :  
開始 1 秒で右下へ、次の 1 秒で右上へ。  
を 3 回繰り返す。

サンプル 2 :  
開始 2 秒で拡大しながら赤→青へ、  
次の 2 秒で縮小しながら青→緑へ。  
その後は逆再生。

サンプル 3、4 :  
ローディングアニメーション

プロパティ	説明
transition	<ul style="list-style-type: none"> <li>・ hover や checked などのトリガー（きっかけ）で発動</li> <li>・ 1 回のトリガーでアニメーションは 1 回のみ</li> <li>・ 単純なアニメーション向き</li> </ul>
animation	<ul style="list-style-type: none"> <li>・ トリガーがなくても発動</li> <li>・ 無限に繰り返すこともできる</li> <li>・ svg 画像や Javascript と組み合わせ、多彩な表現が可能</li> </ul>

※アニメーションを取り入れすぎると閲覧するユーザにストレスを与えることもあるので、程々に…

また、アニメーションは速度も重要で、特にホバー時のアニメーションは 1 秒以内に収めるとよい。



```
.sample1 { animation: zigzag 2s ease 0s 3 normal; }
```

① ② ③ ④ ⑤ ⑥

①任意のアニメーション名    ②アニメーション 1 回分の時間    ③イージング    ④遅延時間    ⑤繰り返しの回数    ⑥繰り返しの際の再生方法

⑤繰り返しの回数は、「infinite」で無限に繰り返す。

⑥繰り返しの際の再生方法は、「normal」で「0%→100%」のアニメーションを繰り返す。

「alternate」で偶数回目の再生だけ「100%→0%」と逆再生になり、⑤繰り返し回数によって「0%→100%→0%→100%→0%・・・」という繰り返しになる。

```
@keyframes zigzag {
```

アニメーションの開始を 0%、終了を 100%として、任意のポイントに細かく変化の過程（キーフレーム）を記述する

```
}
```