

# 情報デザイン応用演習I 4.Canvas応用

# 目次

- 初めに
  - CanvasFirstStep
- バグ対策
- Canvas応用
  - 今日の狙い
  - 物理的な動き
  - オブジェクト指向
    - クラス・インスタンス・プロパティ・メソッド
    - 今日やることは
    - パーティクルプラグイン
- 演習
  - では今日も演習に取り組みましょう。

## 初めに

「ECMAScript入門」「Canvas入門」やっていない人は、今日の内容全くわかりません。  
必ずJavascriptFirstStep,CanvasFirstStepやってからこちらに取り掛かりましょう。

## CanvasFirstStep

軽くおさらいしてみましょう。

一定時間ごとに適当に (ランダム) に描画させることにより、アニメーションを作成するという方法を取りました。

# バグ対策

## バグとは

プログラムの不具合部分のことを指します。

プログラミングの半分以上の作業は、**バグ取り**と言って良いかもしれません。

改めて

プログラムとはある作業を実行させるために、コンピュータに正しく意図を伝えること

であり、正しく意図を伝えられない場合にうまく動かないことを再確認しておきましょう。

## スペルミス・半角全角スペース

スペルミスがあると、動きません。エディターが「これ、なんか変じゃない？」と言ってることがありますので、注意しましょう。

文字としての「全角スペース」以外はプログラミングで全角スペースを利用することはありません。「Zenkaku」拡張機能を入れて、間違いに気づきやすくしましょう。

## 大文字小文字

多くのプログラムでは大文字と小文字は異なるものと解釈されます。  
そのため、間違えずに入力しましょう。



## プログラマー文の終わり

JavaScriptでは命令の後に「;」を入力することが求められます。  
これがないと、どこで区切られているかわかりません。

逆にいうと半角スペースの複数連続・改行はプログラムを解釈するときにはスルーされるということになります。

Python等、プログラミング言語によっては、この辺の作法が変わります。

## 実行時エラー

実行時にエラーを感知すると、「この行のこの辺が怪しいんじゃないかな？」とデベロッパーツールのコンソールに表示されます。

このヒントは非常に有効ですので、頼りにしましょう。

## 意図的に内部の状況を知る

```
console.log(文字列)
```

とすることで、その行を追加したときに文字列をコンソールに表示させることができます。

このことによって、確実にプログラムの実行がそこを通過していることを確認できます。

```
console.log(変数)
```

とすることで、その時点での変数を表示させることもできます。これによって意図しないあたりに変数の値がなっていないか確認できます。

## エラーがない時

やりたいことがあるのに、エラーがなく、期待通りに動かないときが一番面倒くさいです。

コンピュータが指示通り動いているのに、指示の方法(アルゴリズム)が間違えていることを意図します。

変数の値などを確認しながら、期待通りに動くように頑張りましょう。

# Canvas応用

## 今日の狙い

前に描画した情報を元に、次の情報を更新しながら書き換え、「動き」のあるアニメーションを作成します。

## 物理的な動き

`draw()`が呼び出されたタイミングで描画されるのは前と変わりませんが、ひとつ前で、どこに描画したか、を変数で覚えておいて、次の位置を計算しながら描画するとアニメーションとなります。

## オブジェクト指向

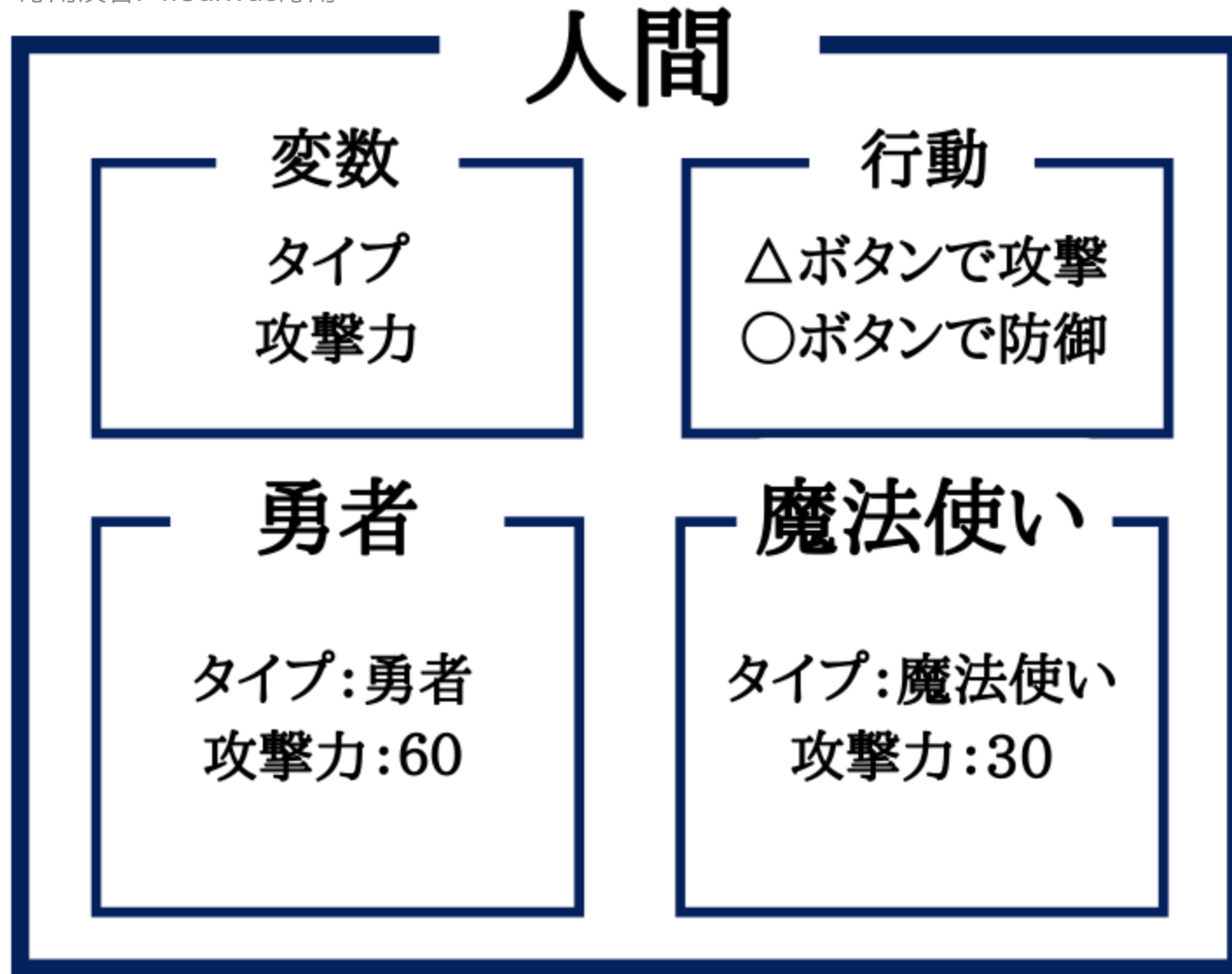
円を一つだけ描画するなら、前回と同じような書き方で良いのですが、ややこしくなってくるので、オブジェクト指向という考え方を導入します。

これは、現在多くのプログラミング言語で採用されている考え方です。

この2,30年**オブジェクト指向**の考え方が良いとされてきましたが、最近では**関数型**の方が良いという動きもあります。

どちらを選ぶ？ 関数型プログラミングとオブジェクト指向プログラミング(OOP)





## クラス・インスタンス・プロパティ・メソッド

- クラス：同じ特徴や性質を持つモノをまとめたオブジェクト。抽象化された概念
- インスタンス：クラスによって、作られたオブジェクト。実体化されたもの
- プロパティ：オブジェクトが持っている特徴
- メソッド：オブジェクトが起こすアクションのこと

## 今日やることは

- クラス：パーティクル(粒子)
- インスタンス：実体化されたパーティクル(粒子)
- プロパティ：色・速度・位置
- メソッド：draw(描く), update(位置情報を更新)

粒子でできちゃえば、線でも三角形でも同じようにプログラムすることができます。

## パーティクルプラグイン

AfterEffectsやBlender等でパーティクルと呼ばれるプラグインがあります。ちょっとみてみましょう。

これを、プログラミングでどのように記述するか、が今日の内容となります。

- **【プラグイン無し】 簡単！パーティクルで紙吹雪を作ろう！【AfterEffects チュートリアル】**

# 演習

**では今日も演習に取り組みましょう。**

今日の教材は

- Canvas SecondStep

です。

(時間なかったら、ちょっと古い書き方のJavaScriptのままかも...)

ひとつずつ、理解しながら進めましょう。