

100連ガチャをプログラムで作ろう！

-アルゴリズムの基本とプログラミング-

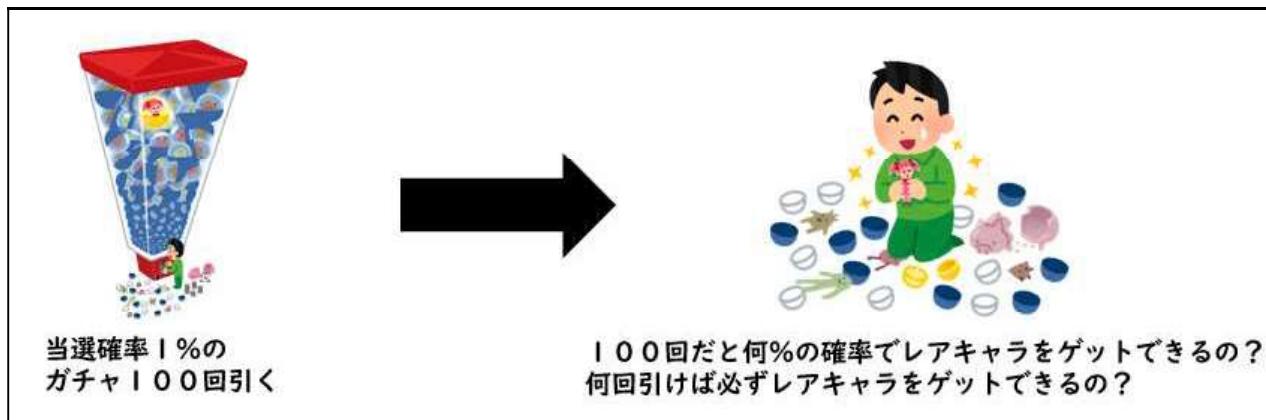
_____組 _____番 氏名 _____

1. ねらい

プログラミング言語の、Pythonによるプログラミングを体験し、身近な事情を（ ）し、プログラムが3つの基本構造で作られていることを理解しよう。

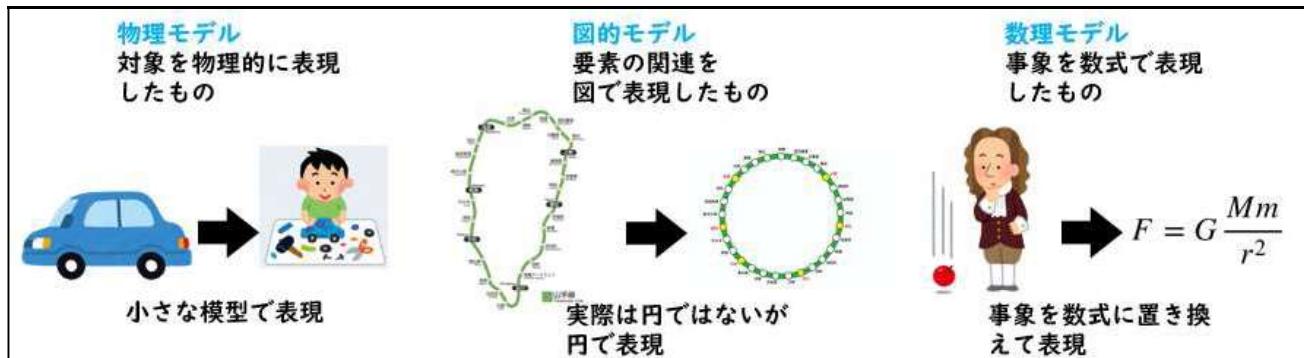
2. シミュレーションとは

シミュレーションとは（ ）を使った（ ）のことである。例えば、当たりの当選確率が1%のガチャを100回引いて、当たりのレアキャラをゲットできるか考えた時に、当選確率をモデル化して（ ）により実際にガチャを引かないで、どのくらいの確率で入手できるか確認できる。今回は当選確率1%のガチャ100回引くシミュレーションを行うプログラミングを行う。



3. モデル化とは

モデルとは物事を（ ）して表現したもの。大きく分けて（ ）、（ ）、そして今回使用する（ ）の3つに分けられる。



問. 今回のシミュレーションはどのモデルに当たるでしょうか？モデルを1つ選び、その理由を書きなさい。

4. 3つの基本構造を組み合わせて、ガチャのシミュレータを作ろう

The diagram illustrates the progression of a Python program. It starts with a simple script:

```
import random  
a=random.randint(1,100)  
print(a)
```

This is followed by a step where the code is modified to include an if-else condition:

```
import random  
a=random.randint(1,100)  
if(a==1):  
    print('atari')  
else:  
    print('hazure')
```

Finally, the code is shown in its completed form, including a loop and a counter variable:

```
[17] import random  
for i in range(100):  
    a=random.randint(1,100)  
    if(a==1):  
        print('atari')  
    else:  
        print('hazure')
```

The output of the final code is displayed below:

```
hazure  
hazure  
hazure
```

5. このガチャのシミュレータの内容を踏まえて、身近な事象をプログラミングで表現してみよう

対象とした事象

(例) あたりが1%のガチャの当たりがでるまでガチャを回す

プログラミングで書き換える内容

(例) 決められた回数繰り返す `range` 文ではなく、条件を満たすまで繰り返す `while` 文を使い、ガチャの当たりがでるまで回す

実行してみてわかったこと

(例) 実行結果を100回分見て数えなくても、何回目で当たったのか簡単に確認できた

6. プログラミングで身近な事象をシミュレーションし、問題解決するとはどのようなことだと感じましたか？振り返りを書こう。

100連ガチャをプログラムで作ろう！

-アルゴリズムの基本とプログラミング-

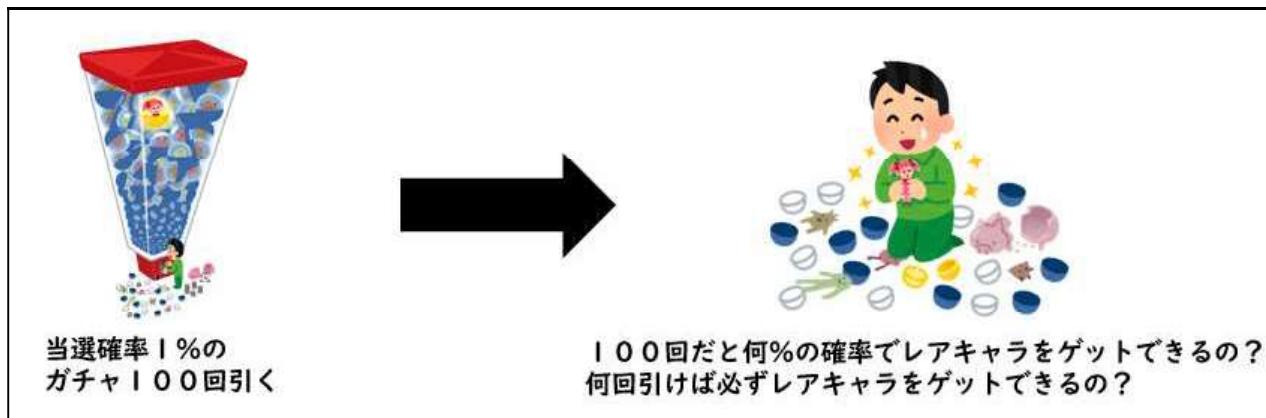
_____組 _____番 氏名 _____

1. ねらい

プログラミング言語の、Pythonによるプログラミングを体験し、身近な事情を（**モデル化**）し、プログラムが3つの基本構造で作られていることを理解しよう。

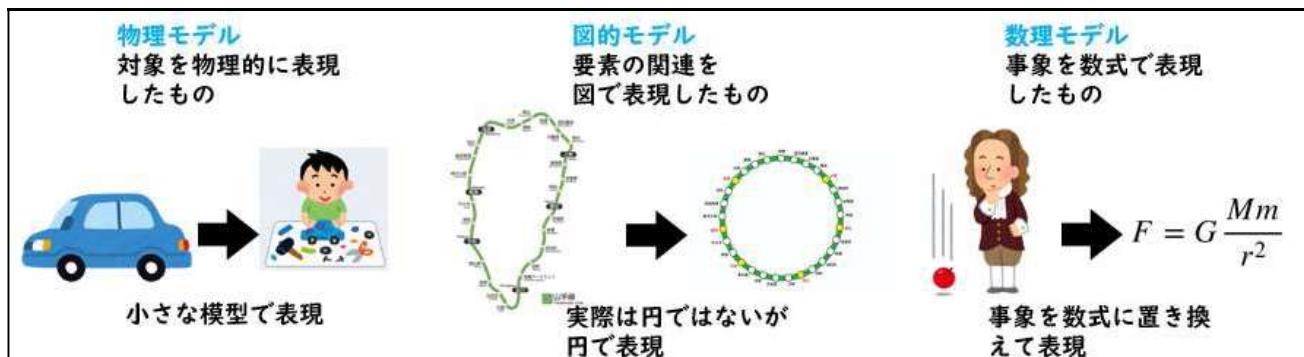
2. シミュレーションとは

シミュレーションとは（**モデル**）を使った（**実験**）のことである。例えば、当たりの当選確率が1%のガチャを100回引いて、当たりのレアキャラをゲットできるか考えた時に、当選確率をモデル化して（**プログラミング**）により実際にガチャを引かないで、どのくらいの確率で入手できるか確認できる。今回は当選確率1%のガチャ100回引くシミュレーションを行うプログラミングを行う。



3. モデル化とは

モデルとは物事を（**単純化**）して表現したもの。大きく分けて（**物理モデル**）、（**図的モデル**）、そして今回使用する（**数理モデル**）の3つに分けられる。



問. 今回のシミュレーションはどのモデルに当たるでしょうか？モデルを一つ選び、その理由を書きなさい。

4. 3つの基本構造を組み合わせて、ガチャのシミュレータを作ろう

The diagram illustrates the progression of a Python program. It starts with a simple script:

```
import random  
a=random.randint(1,100)  
print(a)
```

This is followed by a step where the code is modified to include an if-else condition:

```
import random  
a=random.randint(1,100)  
if(a==1):  
    print('atari')  
else:  
    print('hazure')
```

Finally, the code is shown in its completed form, including a loop and a counter variable:

```
[17] import random  
for i in range(100):  
    a=random.randint(1,100)  
    if(a==1):  
        print('atari')  
    else:  
        print('hazure')
```

The output of the final code is displayed below:

```
hazure  
hazure  
hazure
```

5. このガチャのシミュレータの内容を踏まえて、身近な事象をプログラミングで表現してみよう

対象とした事象

(例) あたりが1%のガチャの当たりがでるまでガチャを回す

プログラミングで書き換える内容

(例) 決められた回数繰り返す `range` 文ではなく、条件を満たすまで繰り返す `while` 文を使い、ガチャの当たりがでるまで回す

実行してみてわかったこと

(例) 実行結果を100回分見て数えなくても、何回目で当たったのか簡単に確認できた

6. プログラミングで身近な事象をシミュレーションし、問題解決するとはどのようなことだと感じましたか？振り返りを書こう。