

おみくじコース

Copyright © Asial Corporation. All Rights Reserved.

1

- ※ スライドにはnoteが付属します。
- ※ noteにはトークスクリプトと先生向けのメモが記載されています
- ※ 『※』からはじまる文章は先生向けのメモです
- ※ 以下トークスクリプトとなります

この單元では「おみくじアプリ」の制作を通じて、
プログラミングの基礎となる「変数」や「分岐」、「関数」や「反復」を学習します。

■ おみくじのコース展開について

0. Monacaとコースの利用方法
1. 【変数と乱数】 おみくじを動かそう
2. 【条件分岐】 おみくじの結果を追加してみよう
3. 【関数と反復】 おみくじを100回引こう
4. 【配列】 配列でおみくじの結果を表示しよう
5. 【応用①】 画像を変更しよう
6. 【応用②】 確率を変更しよう

※ 実際の授業で利用する時間を伝えてください。

おみくじアプリの教材は「4時間」分の内容と応用が「2時間」分、用意されています。
授業では、1～〇コマ目の教材までを扱います。

なお、0コマ目はMonacaとコースの利用方法です。
この教材で学習する間は、授業が始まったら毎回、
0コマ目の手順でMonacaにログインしてコースを開始できる状態にしてください。

※ コマ毎の概要や進め方は、「教材概要(タイムライン付き)」を参照してください
※ 「管理者」でMonacaにログインすることで「コース詳細」より入手できます。」

0コマ目：Monacaとコースの利用方法

■ コースを利用しよう

1. Monaca Educationにアクセス
2. ログイン
3. コース選択
4. コマを「開始」



<https://edu.monaca.io/>

Copyright © Asial Corporation. All Rights Reserved.

4

まず、パソコンのブラウザを起動してください。
コースを利用するために必要な事を教えます。

Monaca Educationにアクセスしてください。
EducationではないMonacaは企業向けなのでEducationを使います。
以降、Monacaと言った場合はEducation版のことを指します。

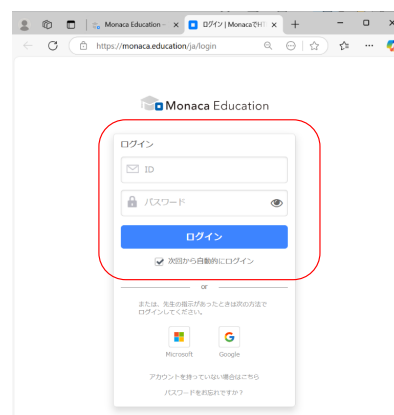
※このタイミングでアカウントを配布する場合
みなさんが作業している間にアカウントを配布します。

このあと、ログインをしてコースを選択していきます。
アカウントは配布した物を使うので『アカウント作成』はしないで下さい。

※ アカウント作成するとフリー版となり、また、コース機能も使えません。
※ 配布したアカウントを『後から』Googleアカウントなどと紐付けることは可能です
※ 生徒が間違っって『アカウント作成』をしてしまった場合、IDを添えて、アシアルまでご
相談下さい。

■ コースを利用しよう

1. Monaca Educationにアクセス
2. ログイン
3. コース選択
4. コマを「開始」



サイト右上の「ログイン」ボタンを押して、ログインページに進んで下さい。

ログインページにて、配られたアカウントのIDと仮パスワードを入力して下さい。

※ 配布したアカウントを『後から』Googleアカウントなどと紐付け場合、次回からはMonacaの仮パスワードを使わずにログインすることも可能となります。学校の実情に合わせて、ご検討ください。

■ コースを利用しよう

1. Monaca Educationにアクセス
2. ログイン
3. コース選択
4. コマを「開始」



Copyright © Asial Corporation. All Rights Reserved.

6

この画面はダッシュボードと呼びます。
右側のパネルのコースから「おみくじコース」を選択して下さい。

もしコースが表示されない場合、先生に教えてください。

※生徒が自身でアカウント作成を行った場合にはコースは表示されません。ログアウトさせて、配布のアカウントでのログインを促します。

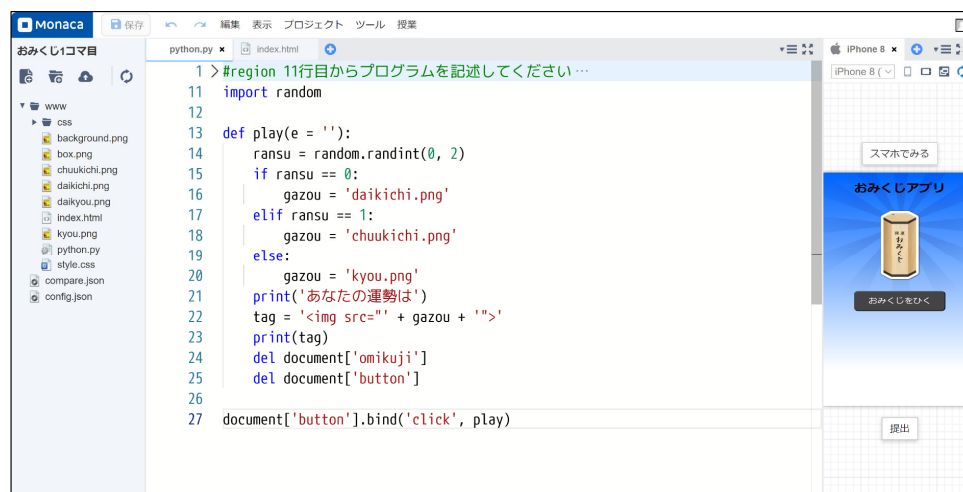
■ コースを利用しよう

1. Monaca Educationにアクセス
2. ログイン
3. コース選択
4. コマを「開始」



今回利用するコマを『開始』して下さい。

■ クラウドIDEが表示されます



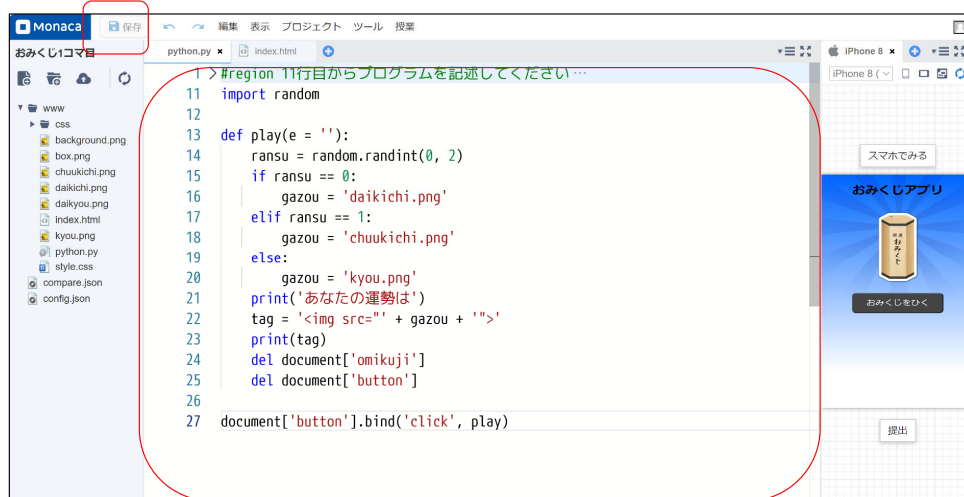
Copyright © Asial Corporation. All Rights Reserved.

8

これで準備完了です。

※ 時間が許せば、以降、クラウドIDEの各機能を、次ページ以降のスライドを元に説明してください ※

【IDE機能】 エディタパネルと保存



Copyright © Asial Corporation. All Rights Reserved.

9

授業では主にエディタパネルの機能を使ってプログラムを修正します。

修正したら左上の『保存』ボタンで保存してください。

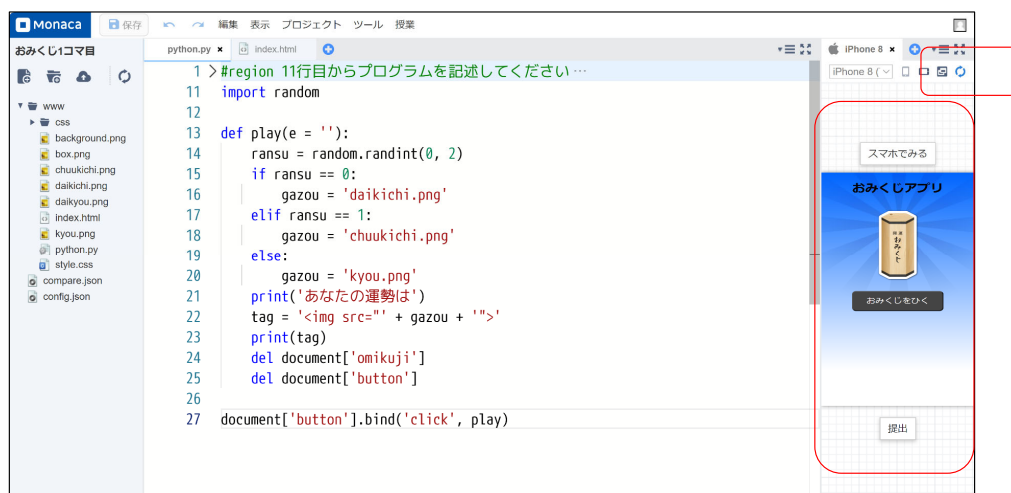
※エディタパネルのタブでは編集するファイルを選択できます。

※『python.py』が最初から開いているので、基本的にはそのまま問題ありません。

※見た目を変えたい人はindex.htmlやstyle.cssを変更すると変えられます。

※意図しない修正でアプリが動かなくなるとフォローが大変です、教えるかどうかはお任せします。

【IDE機能】プレビューパネル



Copyright © Asial Corporation. All Rights Reserved.

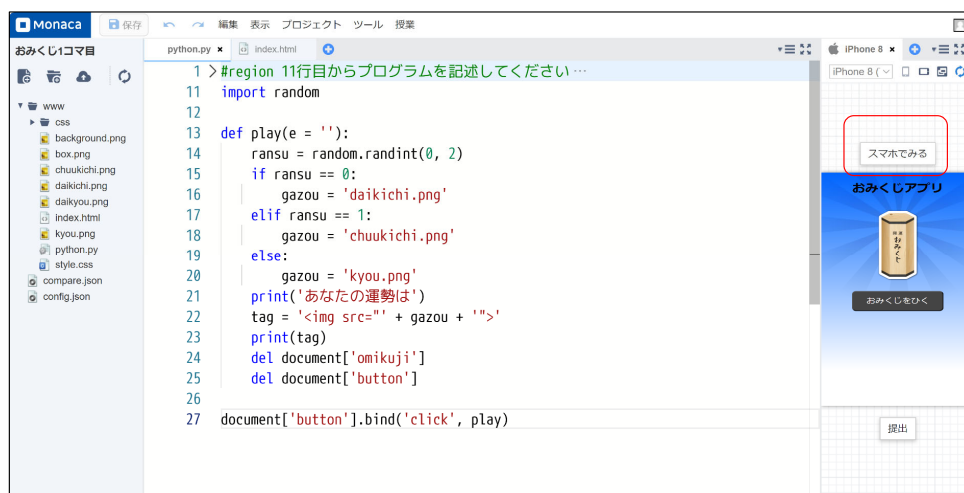
10

「プレビューパネル」でアプリが動きます。

エディタで修正するとアプリは更新されます。

手動で更新したい場合は右上の青いボタンで更新します。

【IDE機能】 スマホでみる

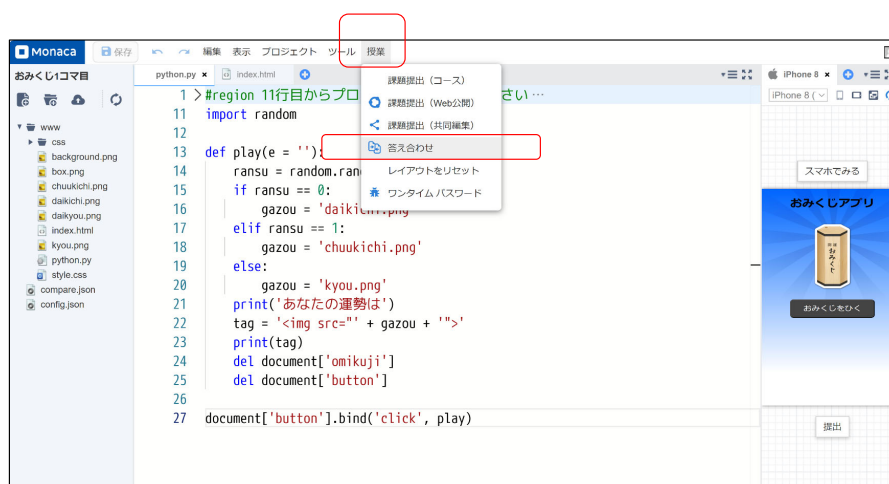


Copyright © Asial Corporation. All Rights Reserved.

||

「スマホでみる」ボタンを押すとQRコードが表示されます。
スマホやタブレットでQRコードを読み取ればアプリを手元で動かせます。

【IDE機能】 答え合わせ機能（許可されている場合）



Copyright © Asial Corporation. All Rights Reserved.

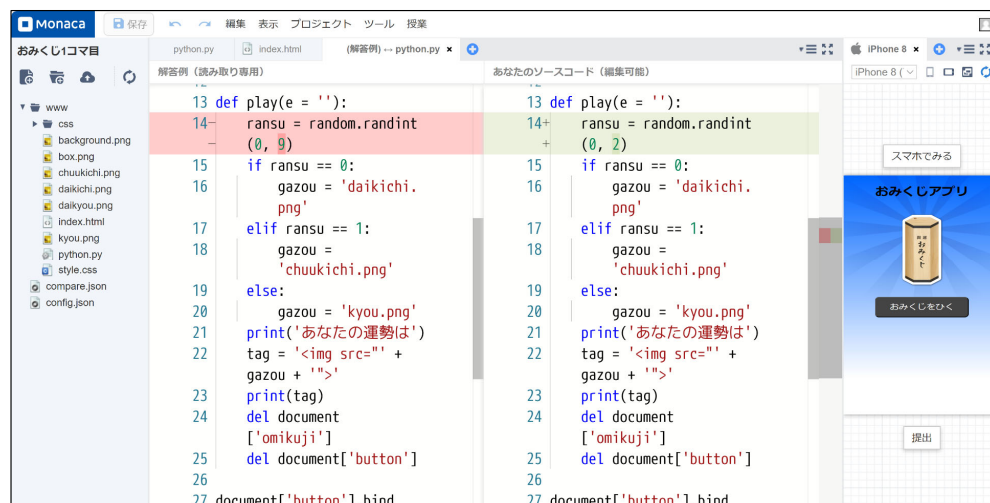
12

上のメニューの「授業」には、授業で良く使う機能がまとめられています。

特に「答え合わせ」機能は、模範解答があればそれを表示してくれる便利機能です。

※ 答え合わせ機能をoffで運用する場合は許可されていない旨をお伝えください ※

【IDE機能】 答え合わせ機能（模範解答例）



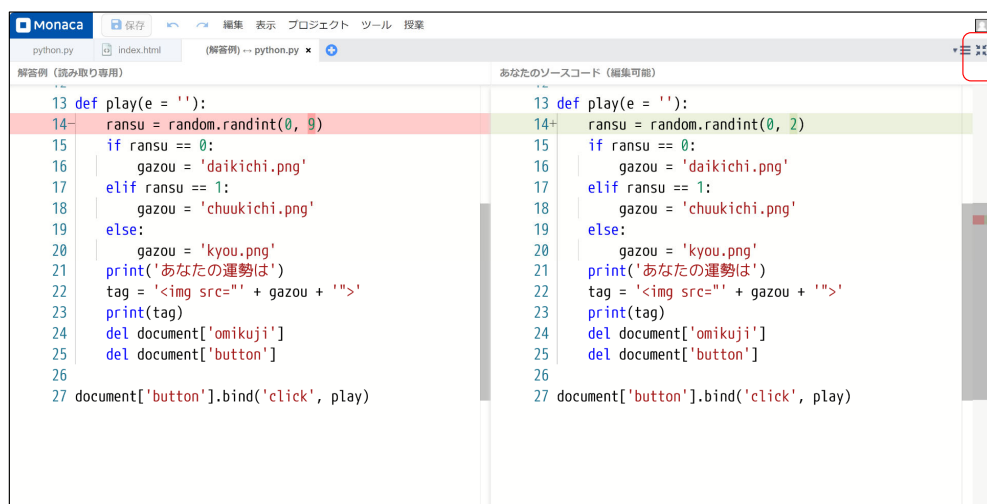
Copyright © Asial Corporation. All Rights Reserved.

13

模範解答が表示できる場合、エディタの左側に模範解答が表示されます。
エディタの右側が自分のコードになります。

模範解答と差分がある場合に背景色が変わります。
模範解答と一致すると背景色が消えます。

【IDE機能】エディタパネルの最大化



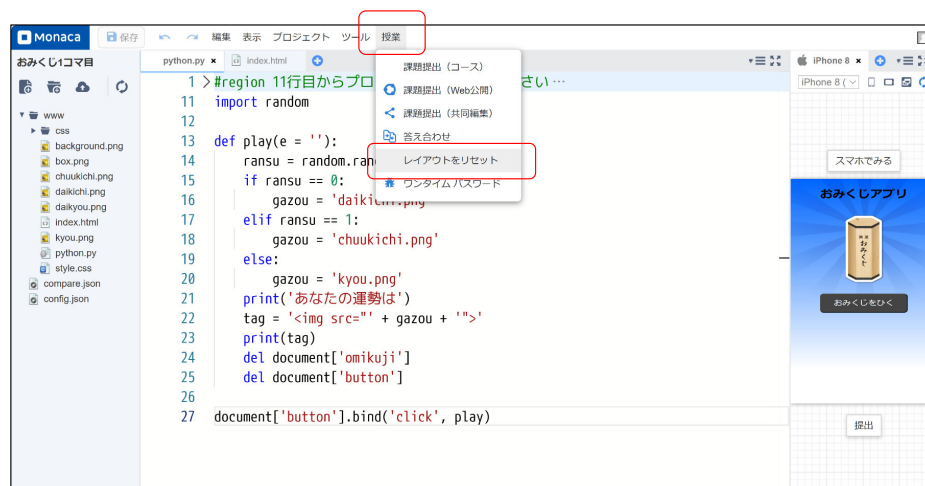
Copyright © Asial Corporation. All Rights Reserved.

14

エディタパネルを広く使いたい場合はエディタパネル右上のアイコンで最大化できます。

答え合わせ機能の模範解答と見比べるときなどに便利です。

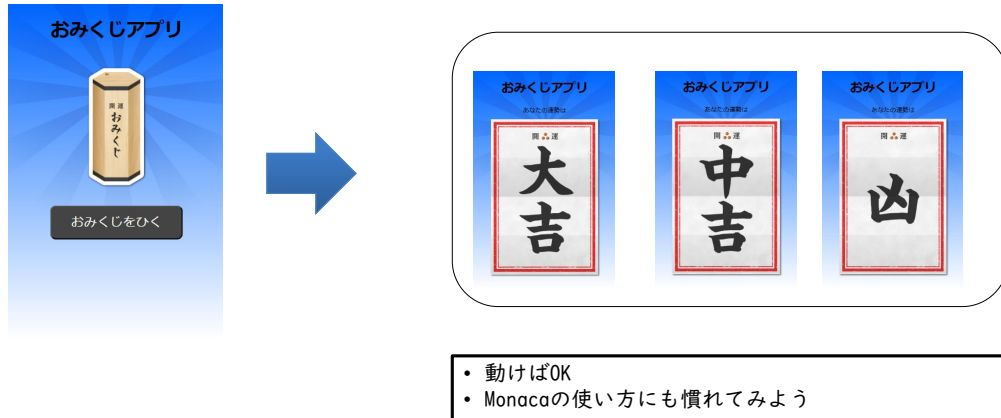
【IDE機能】 レイアウトリセット



クラウドIDEにはエディタの最大化以外にも様々な表示機能があります。
意図せずIDEの表示が崩れてしまったときはレイアウトリセットを行ってください。

Iコマ目：【乱数と変数】おみくじを動かそう

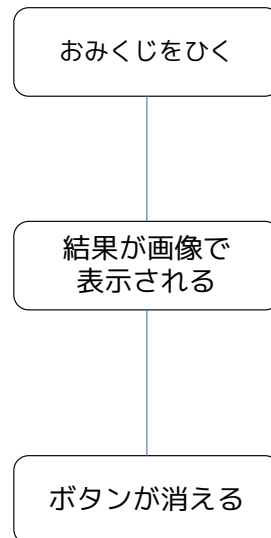
■ 今回のテーマ：おみくじ(とMonaca)を動かしてみよう



今回のテーマはおみくじアプリを動かすことです。
また、Monacaにも慣れていきましょう。

最後に課題提出がありますが、アプリを改造して動かなくなっても、今回はアプリの評価は行わないので大丈夫です。

【解説①】 おみくじアプリ全体のフローチャート

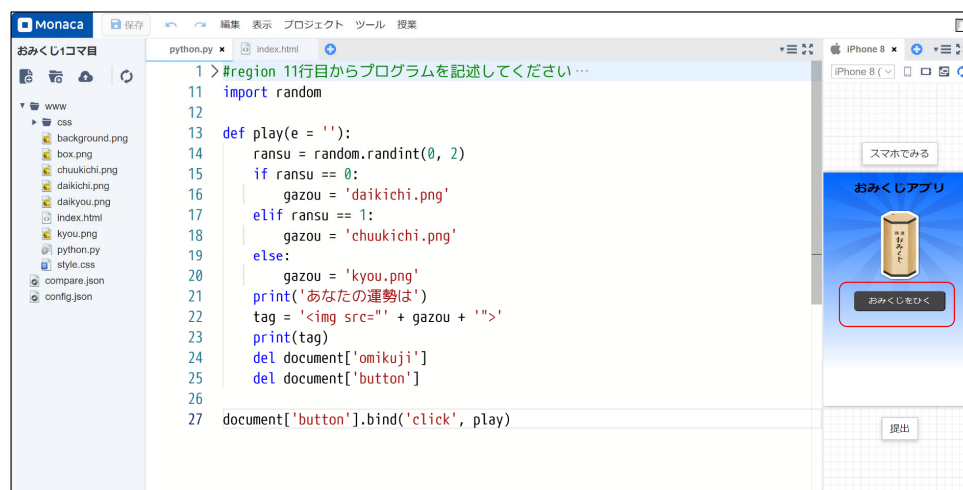


これはおみくじアプリの全体的な流れを表したフローチャートです。
「おみくじをひく」ボタンが押されると、プログラムが動き出します。
押されるまでは、何も起こりません。

ボタンを押すと、結果として何らかの画像が表示されます。
細かい処理は次に解説しますので、まずは実習でおみくじをひいてみましょう。

そして、ボタンが消えます。

【実習①】 おみくじアプリを動かしてみよう



Copyright © Asial Corporation. All Rights Reserved.

19

おみくじアプリを動かしてみましょう

コースを『開始』して出してもらった画面です。
 Monaca EducationのクラウドIDEが表示されます。
 IDEは日本語にすると統合開発環境という意味です。
 要するに、プログラミングに必要なツールが統合されたソフトということです。

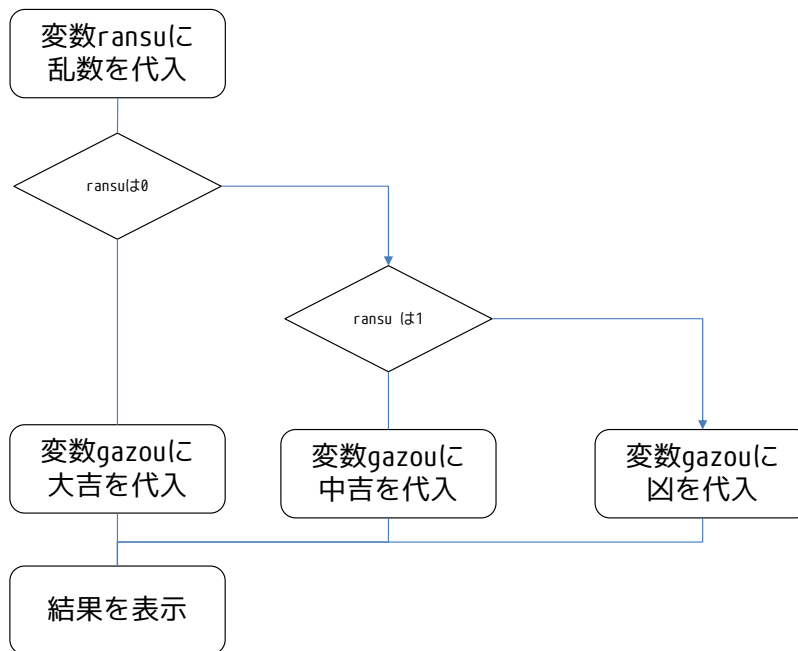
※統合開発環境は長いので教材上にはIDEと表記します。
 ※授業でどのように呼ぶかはお任せいたします。

右側の画像が出ているエリアは「プレビューパネル」と呼びます。
 ここにおみくじアプリが表示されます。

おみくじアプリの「おみくじをひく」をクリックして下さい。
 ランダムに結果が表示されます。

今日からこのアプリのプログラム(ソースコード)やフローチャートを読んで動作を確認したり、修正して動作を変更したりします。

【解説②】 play関数のフローチャート



Copyright © Asial Corporation. All Rights Reserved.

20

結果を表示するまでの細かい処理の流れを確認しましょう。

まず変数ransuに乱数が代入されます。

次に、ransuの値に従って条件分岐が行われています。
ransuの値が0ならば大吉です。

0以外の値の場合は次の条件分岐に処理が分かります。

1なら中吉になります。

また、このフローチャートでは0～1に収まらなかった乱数は全部、凶になります。

コード確認（概要）

```
import random
```

① 乱数の機能を利用するための記述

```
def play(e = ''):
    ransu = random.randint(0, 2)
    if ransu == 0:
        gazou = 'daikichi.png'
    elif ransu == 1:
        gazou = 'chuukichi.png'
    else:
        gazou = 'kyou.png'
    print('あなたの運勢は')
    tag = ''
    print(tag)
    del document['omikuji']
    del document['button']
```

① 乱数を生成して変数ransuに代入

② 条件分岐をして変数gazouにファイル名を代入

③ 表示処理

play関数の中を見ていきましょう。
関数の中もおおまかに3ブロックに分けてみていくことができます。

- ① 乱数を生成して変数ransuに代入
- ② 条件分岐をして変数gazouにファイル名を代入
- ③ 表示処理

今回は主に①のところを解説します。

【解説③】 変数とは？

- └ プログラムの計算結果などを一時的に保持する仕組みです
- └ 「=」で値を変数に代入できます
- └ 値として「文字列」や「数値」などを格納できます

■ おみくじアプリにおける変数ransuの活用

- └ `random.randint()`は呼ぶ度に結果が変わります
- └ 変数ransuに代入することで同じ値を参照できるようにしています

変数について解説します。

変数はプログラムの計算結果などを一時的に保持する仕組みです。
多くのプログラミング言語では「=」で値を変数に代入できます。

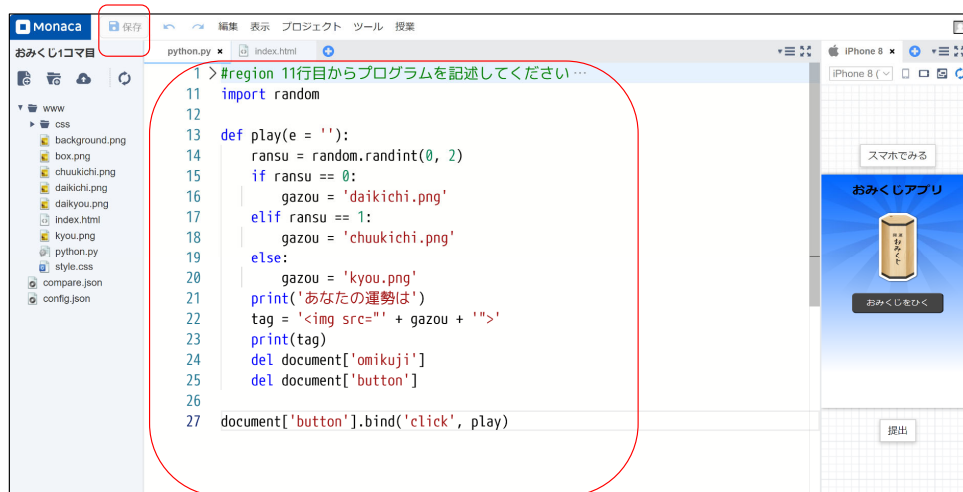
値として「文字列」や「数値」などを格納できます。

おみくじアプリでは変数ransuに、`random.randint()`の結果を代入しています。

`random.randint()`は呼ぶたびに結果が変わります。
そのため、変数ransuに代入することで、呼び出したときと同じ値を参照できるようにしています。

【解説④】 ソースコードを変更する方法

- └ IDEのエディタパネルでソースコードを変更できます
- └ 変更は「保存」することで反映されます



Copyright © Asial Corporation. All Rights Reserved.

24

次の実習に備えて、ソースコードを変更する方法を先に解説します。

IDEのエディタパネルでソースコードを変更できます。

※エディタパネルのタブでは編集するファイルを選択できます。

※『python.py』が最初から開いているので、基本的にはそのまま問題ありません。

※見た目を変えたい人はindex.htmlやstyle.cssを変更すると変えられます。

※意図しない修正でアプリが動かなくなるとフォローが大変です、教えるかどうかはお任せします。

変更は「保存」することで反映されます

【実習②】 凶ばかりでるおみくじを作ってみよう

```
def play(e = ''):
    ransu = random.randint(0, 2)
    if ransu == 0:
        gazou = 'daikichi.png'
    elif ransu == 1:
        gazou = 'chuukichi.png'
    else:
        gazou = 'kyou.png'
    print('あなたの運勢は')
    tag = ''
    print(tag)
    del document['omikuji']
    del document['button']
```

2の部分大きな数値にすると凶ばかりでる

おみくじの確率を変化させる実習を行います。

プログラムの一部を変更することで、乱数の確率を変化させられます。
それによって、おみくじの出やすさも変えることができます。

まずは凶ばかりでるおみくじを作ってみましょう。
2の部分大きな数値にすると凶ばかりでます。

「python.py」のプログラムを修正してみましょう。
エディタパネルで修正を行い、保存を実行して下さい。

保存はメニュー左にある「保存」ボタンで行えます。
古の記録媒体「フロッピー」のアイコンです。

※ショートカットキーの「Ctrl + s」でも保存できます。

改造したアプリは後で提出して貰います。
今回は保存して改造するのがゴールなので、間違った修正をして動かなくなっても構いません。
提出のやり方は後ほど解説します。
確認テストもあるので、このあとのまとめは良く聞いて下さい。

■ まとめ

└ プログラミング

- └ フローチャートを使うとプログラムの処理を図で表現できます
- └ 「random.randint()」 命令で乱数を生成できます
 - └ random.randint(0, 2) ならば 0～2 の整数が生成されます
 - └ random.randint(0, 9) ならば 0～9 の整数が生成されます
- └ 変数には文字列や数値などを代入できます

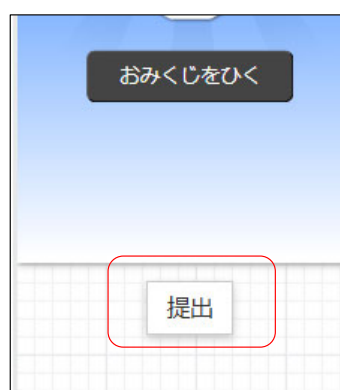
└ Monaca の使い方

- └ 「保存」を忘れずに行いましょう
- └ 「課題提出」も忘れずに行いましょう

※※※進捗状況に合わせてまとめのお話を伝えてください※※※

■ 課題提出

- └ 実習が終わったら課題提出を行って下さい。
- └ 課題には「確認テスト」もあります。



プレビューの下部にある「提出」を押してアプリを提出して下さい。
確認テストも表示されるので回答してください。

■ 次回予告

- └ おみくじの結果を追加します
- └ 今回学習した条件分岐を追加します

※※※進捗状況に合わせて次回のお話を伝えてください※※※

2コマ目：【条件分岐】おみくじの結果を追加してみよう

■ 今回のテーマ：おみくじの結果を追加してみよう



今回のテーマはおみくじアプリの結果を追加することです。

最後に課題提出があります。

【解説①】 分岐とは？

- └ 条件を満たしたときにプログラムを実行します(if)
- └ 条件を満たさなかったときに、別の条件を試みて別のプログラムを実行させることもできます (elif)
 - └ elifは複数、記述できます
- └ いずれの条件も満たさなかったときの処理も指定できます(else)

■ 補足

- └ ifが存在しないとelifやelseは書けません

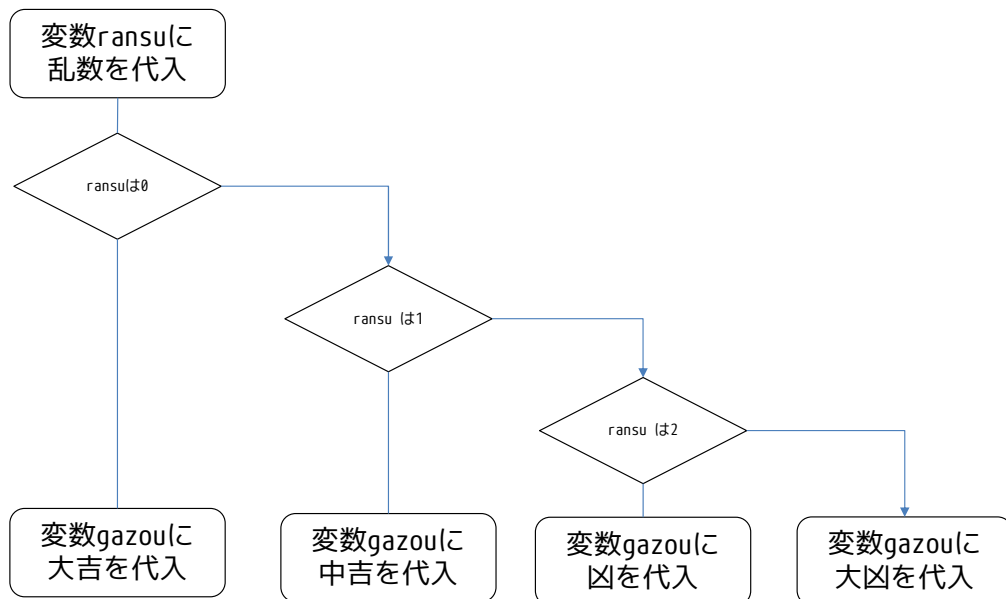
分岐では条件を満たしたときにプログラムを実行します。

まず、条件を満たしたときにプログラムを実行します。
これがif文です。

if文が条件を満たさなかったときに、別の条件を試みて別のプログラムを実行させることもできます。
これがelif文です。
elifは複数、記述できます。

そして、いずれの条件も満たさなかったときの処理がelse文になります。

【解説②】 4パターンに分岐するフローチャート



Copyright © Asial Corporation. All Rights Reserved.

32

結果を表示するまでの細かい処理の流れを確認しましょう。

まず変数ransuに乱数が代入されます。

次に、ransuの値に従って条件分岐が行われています。

ransuの値が0ならば大吉です。

ソースコードではif文を使います。

0以外の値の場合は次の条件分岐に処理が分かります。

1なら中吉になります。

2なら凶です。

この二つの分岐はelif文を使います。

また、このフローチャートでは0～2に収まらなかった乱数は全部、大凶になります。

これはelse文ですね。

【実習①】 elifの値を変更する

```
def play(e = ''):
    ransu = random.randint(0, 9)
    if ransu == 0:
        gazou = 'daikichi.png'
    elif ransu == 1:
        gazou = 'chuukichi.png'
    elif ransu == 1:
        gazou = 'kyou.png'
    else:
        gazou = 'daikichi.png'
    print('あなたの運勢は')
    tag = ''
    print(tag)
    del document['omikuji']
    del document['button']
```

① 文字列 'daikyou.png' が代入されるようにする

今はelseの時に大吉の画像が出るようになっています。
文字列を変更して大凶がでるように変更してください。

【解説④】 演算子とは？

- └ 「=」 は代入演算子と呼びます
- └ 「==」 は比較演算子と呼びます
- └ 「+」 は加算演算子と呼びます
 - └ プログラミング言語によっては文字列同士を連結に使われる場合もあります。

ソースコード上に「=」や「==」、また「+」が出てきましたが、それらは演算子と呼びます。

プログラミングでは「=」は代入になります。
変数に数値や文字列の値を代入するときに使います。

値を比較するときには「==」を使います。
if文で変数ransuの値が1かどうかを確認するときには「=」ではなく「==」を使う必要があります。

また「+」は足し算の意味ですが、文字列同士で使うと文字列連結になります。

■ ま と め

- └ 条件分岐のパターンを増やすためには「elif」を使用します
- └ 文字列は「'」で括ります
 - └ 数値は括りません

※※※進捗状況に合わせてまとめのお話を伝えてください※※※

■ 課題提出

- └ 大凶が表示できる状態で提出してください。
- └ 課題には「確認テスト」もあります。



プレビューの下部にある「提出」を押してアプリを提出して下さい。
確認テストも表示されるので回答してください。

■ 次回予告

- └ おみくじを沢山ひきます（いわゆるガチャ）
- └ 反復に挑戦します

※※※進捗状況に合わせて次回のお話を伝えてください※※※

3コマ目：【関数と反復】おみくじを100回引こう

■ 今回のテーマ：おみくじを100回引こう



おみくじアプリ									
大吉	大吉	大吉	中吉	大吉	中吉	大吉	凶		
中吉	大吉	大吉	大吉	大吉	凶	大吉	大吉		
大吉	大吉	大吉	大吉	凶	中吉	中吉	大吉		
中吉	大吉	大吉	中吉	大吉	大吉	大吉	大吉		
大吉	大吉	大吉	中吉	中吉	凶	大吉	大吉		
凶	大吉	大吉	大吉	大吉	大吉	大吉	大吉		
大吉	大吉	大吉	凶	大吉	大吉	大吉	大吉		
大吉	大吉	大吉	凶	凶	大吉	大吉	凶		
大吉	大吉	大吉	大吉	大吉	大吉	大吉	大吉		
大吉	大吉	大吉	大吉	大吉	大吉	大吉	凶		

- ボタンを100回押す？
- それは大変なので関数playをプログラムで100回呼び出したい
- ボタンで呼ぶ関数をgachaに変更
- そして、gachaからplay関数を呼ぶことにする

今回のテーマはおみくじアプリを100回引きます。

手動でボタンを100回押すのは大変なので、プログラミングで100回引きます。
今まではボタンが押されたらplay関数が1回呼ばれていましたが、今回は100回呼びます。

まず、代わりにplay関数を100回呼んでくれる関数としてgacha関数を作ります。
gacha関数の中で反復処理を書いてplay関数を100回呼ぶことにします。

そして、ボタンが押されたら実行される関数を現在のplay関数から、gacha関数に差し替えます。

今回も最後に課題提出があります。

【解説①】関数とは？

- └ 何度も実行したい命令は『関数』にしておくことで再利用できます
- └ 関数は自分で宣言して作ることができます
 - └ 今回は『def』で宣言
- └ 便利な命令は標準の『関数』として言語にも組み込まれています
 - └ 例えばprint()命令など

■ 補足

- └ プログラミング言語によって標準の関数は異なります
- └ また、便利な関数群を後から追加する仕組みもあります

何度も実行したい命令は『関数』にしておくことで再利用できます。

関数は自分で宣言して作ることができます
今回は『def』で宣言しています。

便利な命令は標準の『関数』として言語にも組み込まれています
例えば、文字を表示するprint命令や乱数を生成する命令などは用意されていることが多いです。

補足として、プログラミング言語によって標準の関数は異なります。
また、便利な関数群を後から追加する仕組みもあります。

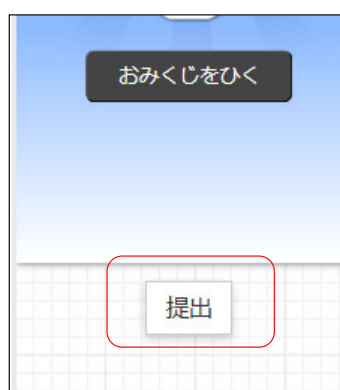
【実習①】 ボタンで呼ぶ関数をgachaに変更する

```
def play(e = ''):
    ransu = random.randint(0, 3)
    if ransu == 0:
        gazou = 'daikichi.png'
    elif ransu == 1:
        gazou = 'chuukichi.png'
    elif ransu == 2:
        gazou = 'kyou.png'
    else:
        gazou = 'daikyou.png'
    tag = ')を抑制できます
- └ 「del」でHTML要素を削除できます

※※※進捗状況に合わせてまとめのお話を伝えてください※※※

## ■ 課題提出

- └ おみくじを100回引ける状態で提出してください。
- └ 課題には「確認テスト」もあります。



プレビューの下部にある「提出」を押してアプリを提出して下さい。  
確認テストも表示されるので回答してください。



## ■ 次回予告

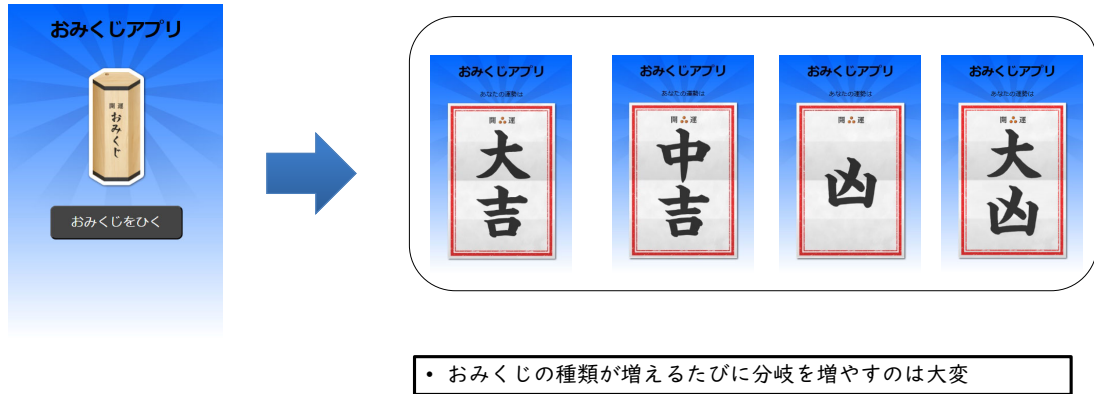
- └ 条件分岐を使わないでおみくじアプリを作ります
- └ 代わりに配列を利用します

※※※進捗状況に合わせて次回のお話を伝えてください※※※

4コマ目：【配列】配列でおみくじの結果を表示しよう

---

## ■ 今回のテーマ：おみくじを配列で表現



今回のテーマはおみくじアプリを「条件分岐」ではなく「配列」で表現します。

実は、条件分岐で結果の種類を増やすのは大変です。  
おみくじなら種類は数種類ですが、仮に元素記号でも出そうとしたら100個以上の分岐が必要です。  
もし本格的にゲーム開発などを考えはじめたら、もっと大変です。

実は、複数の値を管理できる仕組みがあれば条件分岐無しでも結果の出し分けは可能です。  
今回は一つの変数に複数の値を格納できる『配列』という仕組みを学習します。  
配列を活用しておみくじアプリを動かすことに挑戦します。

## 【解説①】 配列とは？

└ 複数の文字列や数値をまとめて扱うための仕組みです

|       |              |               |          |             |
|-------|--------------|---------------|----------|-------------|
| 添字    | 0            | 1             | 2        | 3           |
| gazou | daikichi.png | chuukichi.png | kyou.png | daikyou.png |

配列は、複数の文字列や数値をまとめて扱うための仕組みです。  
添字を持たせることで、任意の値を参照できます。

## 【解説②】 配列の使い方

- └ 配列は変数に代入して利用することが一般的です

```
gazou = ['daikichi.png', 'chuukichi.png', 'kyou.png', 'daikyou.png']
```

- └ 参照時は添字をそえます

```
print(gazou[1])
```

### ■ 添字について

- └ 0はじまりが一般的です（しかし1からはじまる場合もあります）
  - └ 0はじまりなら0はdaikichi.pngを、1はchuukichi.pngを参照します
  - └ 1はじまりなら1がdaikichi.pngを参照します

配列は変数に代入して利用することが一般的です。

変数gazouに配列としてdaikichi.pngやchuukichi.pngの文字列を格納する場合は、[] (かっこ) とカンマを使って、例のように記述します。

また、変数gazouから任意の値を参照するときは添字を指定します。

プログラミング言語に寄りますが一般的に、0はdaikichi.pngを、1はchuukichi.pngを参照します

まれに1はじまりのケースもあります。  
その場合は1を指定するとdaikichi.pngが参照できます。

過去に大学入学共通テストで1はじまりの配列がでたことがあります。  
試験を受ける方は説明文を良く読んで、注意して下さい。

## 【確認①】 配列の参照

└ 次のプログラムが実行された時に表示される文字列は？

└ ※添字は0はじまりとします

```
tenki = ['晴れ', '曇り', '雨', '雪']
print(tenki[1])
```

おみくじ以外の例で、配列から値を参照する例を確認しましょう。

配列tenkiがあり、添字で1を指定した場合に表示される文字列を考えてみてください。

※ 生徒に少し考えて貰います ※

はい、正解は「曇り」です。

0が晴れで1が曇り、2が雨で3が雪です。

もし添字が1はじまりだった場合には、晴れになるので注意して下さい。

## 【実習①】 配列の値を一つ変更する

```
import random #
```

```
gazou = ['daikichi.png', 'daikichi.png']
```

① まずはdaikyou.pngも出るようにしてみましょう

```
def play(e = ''):
 ransu = random.randint(0, 1)
 tag = ''
 print(tag, end='')
```

```
def gacha(e = ''):
 for i in range(100):
 play()
 del document['omikuji']
 del document['button']
```

```
document['button'].bind('click', gacha)
```

配列の値を変更する実習を行います。

値として大吉の文字列が2つ、入っています。  
今おみくじを引いても全部大吉が出ます。

今回は、2番目の大吉をdaikyou.pngに変更してみましょう。

間違ってシングルクォート・カンマ・カクカッコを消すと、上手く動かなくなるので注意して下さい。

### 【解説③】 配列の記述方法

- └ カンマ区切りで列挙します
- └ 最後の要素にはカンマは不要です

```
gazou = ['daikichi.png', 'chuukichi.png', 'kyou.png', 'daikyou.png']
```

### 【復習】 乱数は「random.randint()」で生成

- └ random.randint(0, 1)の場合、0～1の値を生成します
- └ random.randint(0, 9)の場合 0～9の値を生成します

次の実習は、配列の値を追加する実習を行います。

角括弧の中に、カンマで区切って追加します。  
最後の要素にはカンマは不要です。

文字列を追加するときには、シングルクォーテーションが必要です。

補足すると、数値を入れる場合はシングルクォーテーションは不要です。

また、現在のアプリでは乱数として0と1の二つしか生成しないようになっています。  
このままだと、daikichiとchuukichiの二つしか出ませんので乱数も変更します。

0、1、2、3の乱数を生成するにはどうすれば良いか、改めて確認してください。



## 【実習②】 配列の値を追加し乱数も変更する

```
import random #
```

```
gazou = ['daikichi.png', 'daikichi.png']
```

 ① 模範コードを元に他の値も出るようにしてみましょう

```
def play(e = ''):
```

```
 ransu = random.randint(0, 1)
```

 ② 配列の要素数を変更したら乱数の範囲も変更しましょう

```
 tag = ''
```

```
 print(tag, end='')
```

【参考コード】 おみくじの配列

```
gazou = ['daikichi.png', 'chuukichi.png', 'kyou.png', 'daikyou.png']
```

配列の値を追加する実習です。

配列と乱数を書きかえて、  
kyouなども表示できるおみくじアプリを完成させてください。

## 【確認②】 配列と乱数

- └ 次のプログラムが実行された時「daikichi.png」が表示される確率は？
  - └ ※添字は0はじまりとします
  - └ ※存在しない添字を参照しても無視して実行されるものとします

```
ransu = random.randint(0, 9)
gazou = ['daikichi.png', 'chuukichi.png', 'kyou.png', 'daikyou.png']
print(gazou[ransu])
```

配列と乱数を組み合わせた問題を確認しましょう。

配列gazouがあり、添字に乱数を指定します。  
daikichi.pngが表示される確率は何%でしょうか？

※ 生徒に少し考えて貰います ※

はい、正解は10%です。

乱数として0～9の10通りの値を生成しているので、  
daikichi.pngに対応した値である0が出る確率は10分の1で10%になります。

引っかけ問題なので気をつけましょう。

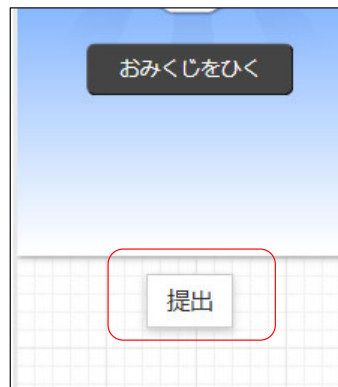
なお、配列で値が存在しない4～9の6通りはエラーになります。  
このプログラムは60%の確率でエラーになります。

## ■ まとめ

- └ 配列を使えば複数の文字列や数値を代入できます
- └ 条件分岐を配列で代替できる場合があります

## ■ 課題提出

- └ 配列版のおみくじを提出してください。
  - └ 大吉・中吉・凶・大凶が出る状態にしてください。
- └ 課題には「確認テスト」もあります。

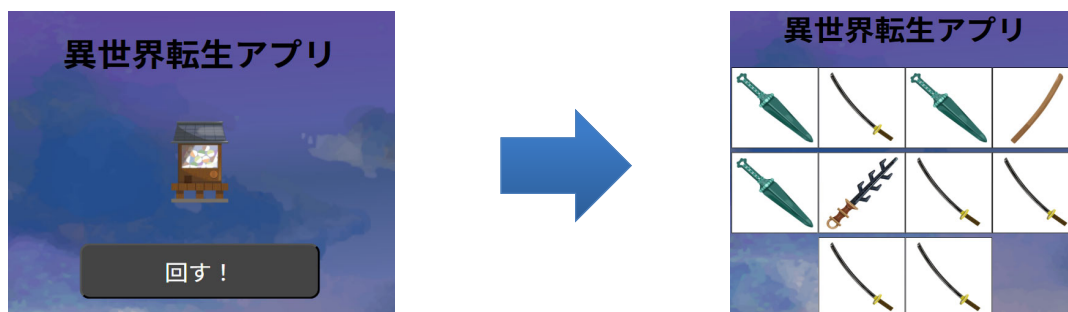


プレビューの下部にある「提出」を押してアプリを提出して下さい。  
確認テストも表示されるので回答してください。

## 5コマ目：【応用①】画像を変更しよう

---

## ■ 今回のテーマ：画像を変更しよう



主に画像を変更して異なる世界観のアプリにカスタマイズします

- ※ 実際には、進捗具合や使える時間にあわせて、指導方針を提示してください。
- ※ 以降のnoteは教材に従って2コマで規定のカスタムを行う指導となります。
- ※ 動画の利用がお勧めです

今回のテーマは応用です。おみくじアプリを参考にしてオリジナルのアプリを作成してください。

いきなりオリジナルのアプリと言われても大変だと思いますが、教材の中では『異世界転生アプリ』を作成する方法が解説されています。教材は、架空の学校で架空のチームが作品作りに挑戦するシナリオになっています。

5コマ目と6コマ目は、教材に従って『異世界転生アプリ』を作成して提出してください。

## 【シナリオ・前置き】異世界転生アプリを作ろう

- └ ある架空の学校で、Monaca Educationを利用したプログラミング実習の仕上げとして、おみくじアプリを改造して提出することになりました。
- └ 彼らの活動を参考に、おみくじアプリを改造して提出して下さい。
- └ なお彼らの活動は架空です、実際に改造すべき点や評価のポイントは、先生から指示があった場合、それに従って下さい。

架空のシナリオはスライドの通りです。

※シナリオの説明は教材の動画をお使い頂くのがお勧めです。

## 【シナリオ・架空の先生からのお題】

- └ プログラミング実習の仕上げとして、おみくじアプリを改造して作品を提出して下さい。
- └ 主な評価のポイントは以下の通りです。
- └ 『乱数により結果が毎回変化することを理解した上で、確率の変更方法や画像の差し替えなどを検討し、実践する』
- └ オリジナルの作品として工夫した場合には追加で加点します。

※スライドを参照して下さい。



## 【シナリオ・架空のチームの会話①】

生徒A：オリジナルの作品か、僕は折角なので「ゲーム風」にしたい。

生徒B：私、ゲームってあんまり興味ないから、「ゲーム風」だけだと、ちょっとイメージが沸かないわ。

生徒C：俺が思うに、ガチャを回すとランダムに強い武器やキャラクターなんかが手に入るっていうのがゲーム風じゃないか？

生徒B：ふうん、そういう感じなのね、ところで、手に入る武器は完全にランダムなの？

生徒C：強い武器は「レア」や「スーパーレア」といって、確率が低くなる。例えば1%だったり。そういえば、一定回数のガチャを回すと確定で「スーパーレア」がでるようなシステムもあるらしい

生徒A：それ、天井と言うらしいよ。

生徒B：その仕組みを調べてレポートしたら、先生から良い評価が貰えるかもね。

※スライドを参照して下さい。

## 【シナリオ・架空のチームの会話②】

生徒A：おみくじだって、結果のおみくじを戻さずに全部引いたら、大吉も大凶もぜんぶでるはずだものね。

生徒B：くじ引きや福引きは、結果を戻さないわよね？でないと、用意した景品が足りなくなってしまうもの。

生徒A：架空の商店街の福引きを作るのも面白そうだけど、僕は、やっぱりゲームを作りたい。しかし、どこから手を付ければ良いんだろう？

生徒C：まずは、画像を差し替えて「ゲーム風」に変えてみてはどうだろう？『異世界に転生したボーナスでガチャを回す』という設定で。

生徒A：なるほど、でも画像を入手したり登録する方法が分からないんだけど。

生徒B：Monacaにはイラスト素材集があったはず、でも、使い方が分からないわ。

生徒C：とりあえず、画像をアップロードしておくから、プログラムを書きかえて、おみくじの画像を武器の画像に変えてみてくれないか？

生徒A：分かった、やってみるよ

※スライドを参照して下さい。

## 【実習①】

└ 配列の値を、武器の画像のファイル名に変更します

```
gazou = ['sitisitou.png', 'katana.png', 'douken.png', 'bokutou.png']

def play(e = ''):
 ransu = random.randint(0, 3)
```

配列の値を変更する実習です。

4コマ目の復習を兼ねています。

## 【解説①】 武器の画像は何処にある？

- 左のパネル（プロジェクトパネル）で管理されています
- ファイル名をクリックするとプレビューできます

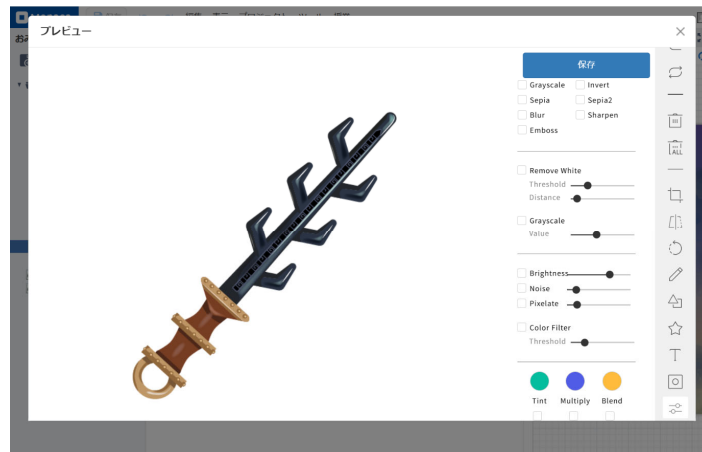


今回の課題では、すでに武器の画像が用意されています。

左側のプロジェクトパネルでファイルをクリックすると画像をプレビューできます。

## 【解説①】 武器の画像は何処にある？

- プレビュー機能では簡易的な編集も行えます
  - ・ ※プレビュー機能は将来、変更される可能性があります



プレビュー機能では画像などを表示できます。

スライドの画像プレビュー機能は簡易的な編集機能も備えています。

なお、将来、プレビュー機能は変更される可能性があるそうです。

**【シナリオ・架空のチームの会話③】**

生徒A：このおみくじアプリ、背景画像が変更されている？

生徒C：お気づきになりましたか

生徒B：いや、気づくでしょう

生徒C：実は、Webアプリはスタイルシートを変更することで文字のサイズや色だけでなく、背景画像なども変更できる。だけど、見た目を変更しはじめると面白くなって、時間が幾らあっても足りなくなる

生徒B：それは困るわね、課題提出が間に合わない

生徒A：一箇所、おみくじの箱はゲーム風にしたいのだけど、スタイルシート？

生徒C：いや、そちらはHTML言語。画像を表示するためのIMGタグが書かれていて、『box.png』を参照している。どちらにせよ、『box.png』の名前で新しい画像をアップロードすれば変更できる

生徒B：素材集にあるかしら、素材集の使い方を教えて貰える？

※スライドを参照して下さい。

**【シナリオ・架空のチームの会話④】**

先生：素材集を使いたいのか？

生徒A：あ、先生、居たんですか！

先生：『コース詳細』に戻って、教材の『素材集サイト』を押せば素材集を開ける。だが、実習中のプロジェクトに途中から戻るにはコツがいる。

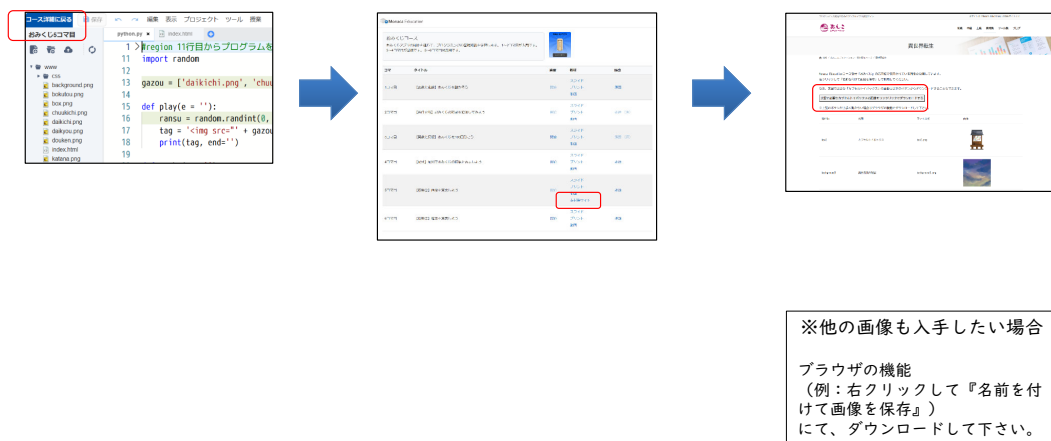
生徒B：素材集の実習を①にして、素材を使う実習を②にすれば良かったのでは？

先生：オリジナルの作品作りをする場合プロジェクトの中断や再開は避けて通れないからね。けど、6コマ目までしか時間が取れないので、5コマ目の今、挑戦して欲しい

※スライドを参照して下さい。

## 【実習②】 ファイルのダウンロードとアップロード

- IDE の『コース詳細に戻る』からコース詳細に戻ります
- 5コマ目の教材『素材集サイト』にて素材集に移動
- 素材としてカプセルトイボックスの画像を入手

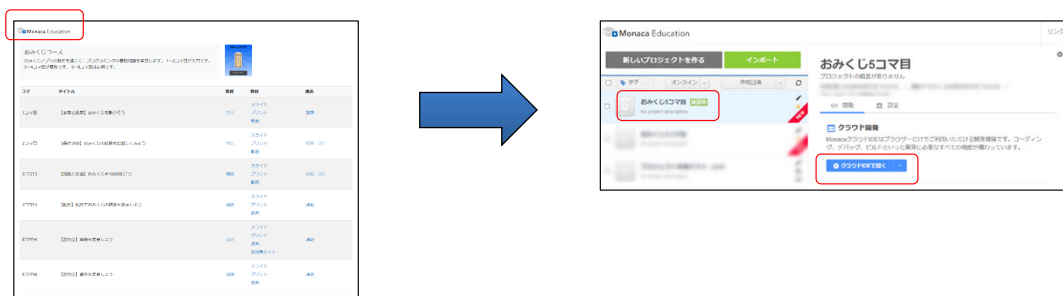


スライドの内容を参考にして、カプセルトイボックスの画像を入手して下さい。  
今、実習しているパソコンにダウンロードします。  
次のスライドでアップロードの解説を行います。



## 【実習②の続き】 ファイルのダウンロードとアップロード

- └ コース詳細の左上のアイコンからダッシュボードに戻ります
- └ ダッシュボード左のプロジェクト一覧から、『おみくじ5コマ目』を選択します
- └ 『クラウドIDEで開く』を選択するとIDEの画面に戻れます



### アップロードの解説をします

先ほどのプロジェクトにまずは戻って下さい。  
コース詳細からは戻れません。

ダッシュボードに移動して、ダッシュボードの左側にある『おみくじ5コマ目』を選択して下さい。

## 【実習②の続き】 ファイルのダウンロードとアップロード

- プロジェクトパネルのファイルアップロードアイコンを押下します
- ファイル名に『box.png』と記述します
- 『ファイル選択』でダウンロードしたファイルを選択してアップロードしてください
- プレビューパネルの更新ボタンを押下します
- 箱の画像が変更されたら成功です。プロジェクトを提出してください。



プロジェクトパネルにアップロードのアイコンがあります。

押下するとファイルアップロードのための画面が表示されます。  
ファイル名に「box.png」と記述して下さい。

次に「ファイル選択」を押下して、パソコンの中にある、先ほどのダウンロードしたファイルをアップロードします。

※先ほどダウンロードしたファイルは「box3.png」ですが、ファイル名を指定することで「box.png」としてアップロードされます。

※アップロード機能は、ファイルを強制的に上書き仕様のため、既存のおみくじの画像は上書きされます。

## 【付録】ダッシュボードから素材サイトを開く方法

- ダッシュボードのリンク集からアクセスできます



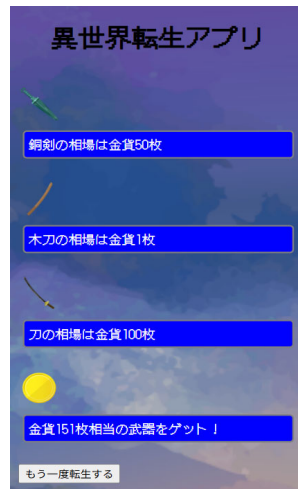
付録の情報です。

素材集サイトにはダッシュボードから移動することもできます。

## 6コマ目：【応用②】確率を変更しよう

---

## ■ 今回のテーマ：確率を変更しよう



- 武器毎に出現する確率を変更して下さい。
- 武器を全部売却したときの合計金額を表示して下さい。

※ 実際には、進捗具合や使える時間にあわせて、指導方針を提示してください。  
※ 以降のnoteは教材に従って2コマで規定のカスタムを行う指導となります。

6コマ目も引き続き、教材に従って『異世界転生アプリ』を作成して提出してください。

## 【シナリオ・架空のチームの会話①】

生徒A：あとはレアリティに応じて、武器を出し分ける処理を付けたら終わり

生徒B：でも、武器が沢山でても、意味はあるのかしら？

生徒A：武器が壊れる世界なのかもしれないけど…余ったら売るとか？

生徒C：出た武器を売却した場合の金額を表示するアプリに変更しておいたぞ！

生徒A：いつのまに…なんか文字のデザインまで変わっているし。これがスタイルシートのか。でも、レアばかり出るし合計金額の表示もおかしいような…

生徒C：バグ修正は任せた！俺は俺の道を行く、別のカスタムをするから！

生徒A：えー

生徒B：頑張ってるね。私も、素材集にあるカフェの画像を使って一日の売上げをシミュレーションするアプリを作ることにするわ

生徒A：僕も早く終わらせて、追加のカスタムをするか見た目を変更しようかな

※スライドを参照して下さい。

## 【解説①】 確率の変更

- └ 条件分岐の比較では「==」以外の記号も使えます
  - └ 「<」や「>」で「未満」や「より大きい」を表します
  - └ 「<=」や「>=」で「以下」や「以上」を表します
- └ 比較で使う記号を「比較演算子」と呼びます
- └ 比較演算子で確率を制御できます（以下例）
  - └ 0～99の100通りの乱数に設定

```
ransu = random.randint(0, 99)
if ransu < 5:
 print('5%の当たり')
else:
 print('95%の外れ')
```

条件分岐と比較演算子を活用して、武器が出る確率を変更します。

これまでは「==」で右辺と左辺が一致するかどうかを比較してきましたが、「<」を使えば、「未満」かどうかを比較することができます。

これを活用すれば、乱数を0～99の100通り生成して、  
0～4の五通りの値が出たときだけ「当たり」  
それ以外の、5～99の95通りは「外れ」にする、  
そんな確率で条件分岐するプログラムが作れます。

## 【実習①】 確率の変更

└ 武器毎の出現確率を変更する

└ 模範例：レア武器である『七支刀』の出現率を10%に変更

```
def play(e = ''):
 ransu = random.randint(0, 99)
 if ransu < 10:
 soeji = 0
 elif ransu < 30:
 soeji = 1
 elif ransu < 60:
 soeji = 2
 else:
 soeji = 3
```

| 添字 | 武器  | 確率 |
|----|-----|----|
| 0  | 七支刀 | 10 |
| 1  | 刀   | 20 |
| 2  | 銅剣  | 30 |
| 3  | 木刀  | 40 |

4通りある武器を、スライドにあるような出現確率で発生するように変更して下さい。

乱数で0～9の10通りの値が出たときには、七支刀が、  
 10～29の20通りの値が出たときには、刀が出ます。  
 刀の出る確率は30%ではなく20%になります。  
 それを意識しながらやってみて下さい。



## 【解説②】 反復で加算

- └ 実は関数の結果を「値」として返すことができます
- └ 今回のプログラムではplay()は武器の金額を返します
- └ 値は変数に代入して再利用できます
  - └ 代入するだけなら『変数 = 関数』のように記述します
- └ 反復と組み合わせれば武器の合計金額が求められます
  - └ 合計を求めるなら『変数 = 変数 + 関数』のように記述します

次の実習では反復を活用して、武器を売却した場合の金額を求めます。

反復処理でplay()関数を実行する度に、売却金額を求めます。  
その売却金額を合計金額に、反復しながら、加算していきます。

## 【実習②】 反復で加算

└ 反復しながら変数goukeiの値を加算する

```
def gacha(e = ''):
 goukei = 0
 for i in range(3):
 goukei = goukei + play()
```

配列の値を追加する実習です。

配列と乱数を書きかえて、  
kyouなども表示できるおみくじアプリを完成させてください。