

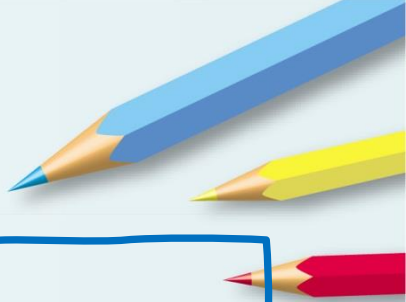
プログラミング教室のテクノロ



Pythonの道

tkinter復習ドリル③

もくじ



- ・ 二次元リストについて（おさらい）
- ・ RPGのマップを表示する
- ・ マップ上にキャラクターを表示する
- ・ キャラクターを動かす
- ・ ゲームをクリアできるようにする

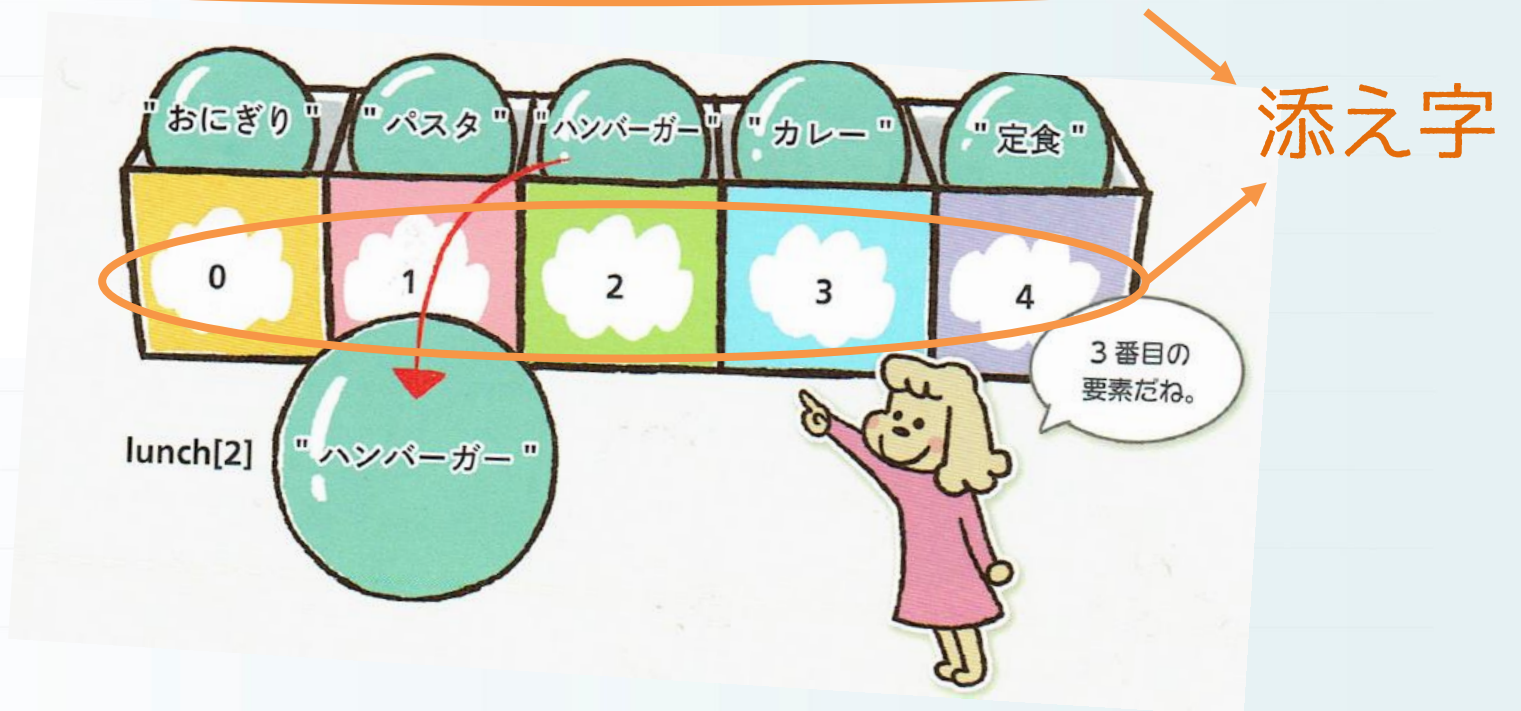
二次元リストについて(おさらい)

迷路のようなデータは二次元のリストで定義する。二次元リストとは横方向(行)と縦方向(列)に添え字を使ってデータを扱うリストのことをいう。

添え字についてのおさらい

```
lunch = ["おにぎり", "パスタ", "ハンバーガー", "カレー", "定食"]  
print(lunch[2])
```

⇒ ハンバーガー
>>>



二次元リストについて(おさらい)

横方向をx、縦方向をyとした場合の添え字の表し方を理解する。

$m[y][x]$
添え字

$x \rightarrow$

$y \downarrow$

$m[0][0]$	$m[0][1]$	$m[0][2]$	$m[0][3]$
$m[1][0]$	$m[1][1]$	$m[1][2]$	$m[1][3]$
$m[2][0]$	$m[2][1]$	$m[2][2]$	$m[2][3]$

例えば右下角の $m[2][3]$ に10を代入する場合、
 $m[2][3] = 10$ と記載する。

1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	1
1	0	1	1	0	0	1	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	1	1	1	1	0	1
1	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1

RPGのマップを表示する

ウィンドウを作成し、マップに表示する画像を読み込む

#tkinterモジュールの呼び出し

```
import tkinter
```

#変数「ウィンドウ」にtkinter.Tk()関数を代入する

```
ウィンドウ=tkinter.Tk()
```

#ウィンドウのタイトルを指定

```
ウィンドウ.title("ダンジョン&パイソン")
```

#ウィンドウのサイズを指定

```
ウィンドウ.minsize(840,454)
```

#ウィンドウの全ての文字のフォントを指定

```
ウィンドウ.option_add("*font",["メイリオ",14])
```

#キャンバス作成

```
キャンバス=tkinter.Canvas(width=620,height=434)
```

```
キャンバス.place(x=10,y=10)
```

RPGのマップを表示する

#キャンバスに矩形を描く

```
キャンバス.create_rectangle(0,0,620,434,fill="gray",  
tag="フィールド")
```

#画像データを読み込み

```
画像=[tkinter.PhotoImage(file="フィールド.png"),  
      tkinter.PhotoImage(file="岩.png"),  
      tkinter.PhotoImage(file="ゴール.png"),  
      tkinter.PhotoImage(file="カギ.png"),  
      tkinter.PhotoImage(file="勇者.png")]
```

#mainloop()関数を実行してウィンドウを表示する

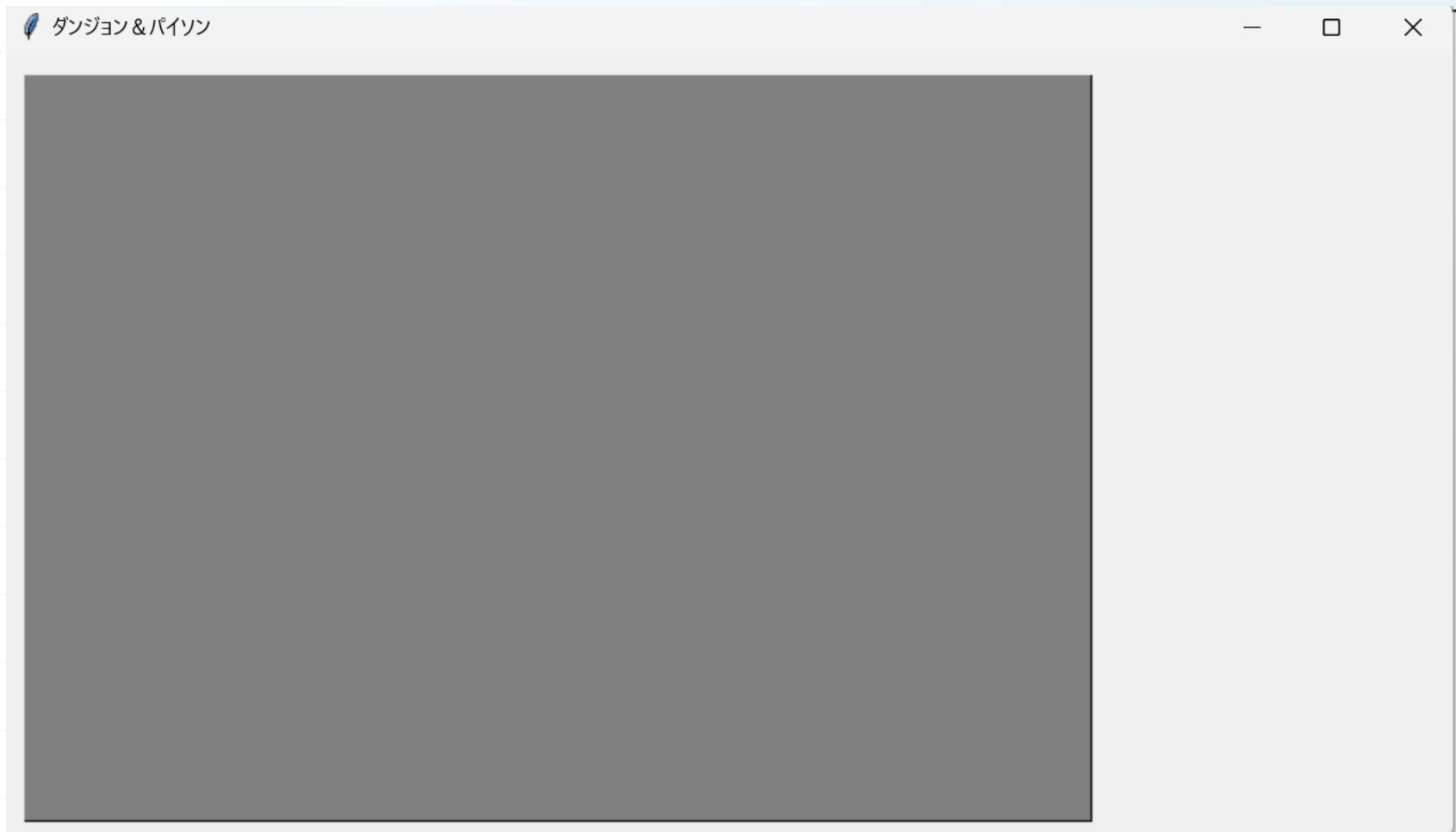
```
ウィンドウ.mainloop()
```

※リストの順番は、フィールド(地面)が0、障害物(岩)が1、ゴールが2、鍵が3、主人公が4となるように設定する。

この数値がマップデータに記録する数値と対応するので、間違えると正しく表示されない。

RPGのマップを表示する

「Run Module」またはF5キーを押してプログラムを実行する。



RPGのマップを表示する

マップデータを用意する。マップを表示する関数を書く。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p=map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
ウィンドウ=tkinter.Tk()
```

```
ウィンドウ.title("ダンジョン&パイソン")
```

```
ウィンドウ.minsize(840,454)
```

```
ウィンドウ.option_add("*font",["メイリオ",14])
```

```
キャンバス=tkinter.Canvas(width=620,height=434)
```

```
キャンバス.place(x=10,y=10)
```

```
キャンバス.create_rectangle(0,0,620,434,fill="gray",tag="フィールド")
```

```
画像=[tkinter.PhotoImage(file="フィールド.png"),
```

```
      tkinter.PhotoImage(file="岩.png"),
```

次のページへ続く

RPGのマップを表示する

```
tkinter.PhotoImage(file="ゴール.png"),  
tkinter.PhotoImage(file="カギ.png"),  
tkinter.PhotoImage(file="勇者.png")]
```

#マップデータ

```
map=[[1,0,1,1,1,1,1,1,1,1],  
      [1,0,0,1,2,0,0,1,3,1],  
      [1,1,0,1,1,1,0,1,0,1],  
      [1,0,0,0,0,0,0,1,0,1],  
      [1,0,1,1,1,1,1,1,0,1],  
      [1,0,0,0,0,0,0,0,0,1],  
      [1,1,1,1,1,1,1,1,1,1]]
```

#ダンジョン()関数を呼び出す

ダンジョン()

#mainloop()関数を実行してウィンドウを表示する

ウィンドウ.mainloop()

RPGのマップを表示する

「Run Module」またはF5キーを押してプログラムを実行する。



マップ上にキャラクターを表示する

主人公のキャラクターを表示させる。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p=map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
#主人公を表示させる
```

```
    キャンバス.create_image(brave_x*62+31,brave_y*62+31,image=画像[4],
```

```
    tag="勇者")
```

```
ウィンドウ=tkinter.Tk()
```

```
ウィンドウ.title("ダンジョン&パイソン")
```

```
ウィンドウ.minsize(840,454)
```

```
ウィンドウ.option_add("*font",["メイリオ",14])
```

```
キャンバス=tkinter.Canvas(width=620,height=434)
```

次のページへ続く

マップ上にキャラクターを表示する

```
キャンバス.place(x=10,y=10)
```

```
キャンバス.create_rectangle(0,0,620,434,fill="gray",tag="フィールド")
```

```
画像=[tkinter.PhotoImage(file="フィールド.png"),  
      tkinter.PhotoImage(file="岩.png"),  
      tkinter.PhotoImage(file="ゴール.png"),  
      tkinter.PhotoImage(file="カギ.png"),  
      tkinter.PhotoImage(file="勇者.png")]
```

#マップデータ

```
map=[[1,0,1,1,1,1,1,1,1,1],  
     [1,0,0,1,2,0,0,1,3,1],  
     [1,1,0,1,1,1,0,1,0,1],  
     [1,0,0,0,0,0,0,1,0,1],  
     [1,0,1,1,1,1,1,1,0,1],  
     [1,0,0,0,0,0,0,0,0,1],  
     [1,1,1,1,1,1,1,1,1,1]]
```

次のページへ続く

マップ上にキャラクターを表示する

#主人公の位置

brave_x = 1

brave_y = 0

#ダンジョン()関数を呼び出す

ダンジョン()

#mainloop()関数を実行してウィンドウを表示する

ウィンドウ.mainloop()

マップ上にキャラクターを表示する

「Run Module」またはF5キーを押してプログラムを実行する。



キャラクターを動かす

ボタン入力に従ってキャラクターを動かしていく。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p = map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
#主人公を表示させる
```

```
    キャンバス.create_image(brave_x*62+31,brave_y*62+31,image=画像[4],tag="勇者")
```

```
    ウィンドウ=tkinter.Tk()
```

```
    ウィンドウ.title("ダンジョン&パイソン")
```

```
    ウィンドウ.minsize(840,454)
```

```
    ウィンドウ.option_add("*font",["メイリオ",14])
```

次のページへ続く

キャラクターを動かす

#キャンバス作成

```
キャンバス=tkinter.Canvas(width=620,height=434)
キャンバス.place(x=10,y=10)
キャンバス.create_rectangle(0,0,620,434,fill="gray",
tag="フィールド")
```

#ボタン配置

```
ボタン上=tkinter.Button(text="↑")
ボタン上.place(x=720,y=150)
ボタン下=tkinter.Button(text="↓")
ボタン下.place(x=720,y=210)
ボタン左=tkinter.Button(text="←")
ボタン左.place(x=660,y=180)
ボタン右=tkinter.Button(text="→")
ボタン右.place(x=780,y=180)
```

次のページへ続く

キャラクターを動かす

#画像データを読み込み

```
画像=[tkinter.PhotoImage(file="フィールド"),  
      tkinter.PhotoImage(file="岩.png"),  
      tkinter.PhotoImage(file="カギ.png"),  
      tkinter.PhotoImage(file="ゴール.png"),  
      tkinter.PhotoImage(file="勇者.png")]
```

#マップデータ

```
map_data=[[1,0,1,1,1,1,1,1,1],  
          [1,0,0,1,2,0,0,1,3,1],  
          [1,1,0,1,1,1,0,1,0,1],  
          [1,0,0,0,0,0,0,1,0,1],  
          [1,0,1,1,1,1,1,1,0,1],  
          [1,0,0,0,0,0,0,0,0,1],  
          [1,1,1,1,1,1,1,1,1,1]]
```

#主人公の位置

```
brave_x = 1  
brave_y = 0
```

次のページへ続く

キャラクターを動かす

#ダンジョン()関数を呼び出す

ダンジョン()

#mainloop()関数を実行してウィンドウを表示する

ウィンドウ.mainloop()

「Run Module」またはF5キーを押してプログラムを実行する。



キャラクターを動かす

coords()関数を使ってキャラクターを移動させる。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p = map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
            キャンバス.create_image(brave_x*62+31,brave_y*62+31,image=画像[4],
```

```
tag="勇者")
```

```
#ボタンが押された時の関数を定義
```

```
#上ボタンが押されたとき
```

```
def クリックボタン上():
```

```
    global brave_y
```

```
    brave_y = brave_y - 1
```

```
    キャンバス.coords("勇者",brave_x*62+31,brave_y*62+31)
```

次のページへ続く

キャラクターを動かす

#下ボタンが押されたとき

```
def クリックボタン下():
```

```
    global brave_y
```

```
    brave_y = brave_y + 1
```

```
    キャンバス.coords("勇者", brave_x*62+31,brave_y*62+31)
```

#左ボタンが押されたとき

```
def クリックボタン左():
```

```
    global brave_x
```

```
    brave_x = brave_x - 1
```

```
    キャンバス.coords("勇者", brave_x*62+31,brave_y*62+31)
```

#右ボタンが押されたとき

```
def クリックボタン右():
```

```
    global brave_x
```

```
    brave_x = brave_x + 1
```

```
    キャンバス.coords("勇者", brave_x*62+31,brave_y*62+31)
```

次のページへ続く

キャラクターを動かす

```
ウィンドウ=tkinter.Tk()
ウィンドウ.title("ダンジョン&パイソン")
ウィンドウ.minsize(840,454)
ウィンドウ.option_add("*font",["メイリオ",14])
キャンバス=tkinter.Canvas(width=620,height=434)
キャンバス.place(x=10,y=10)
キャンバス.create_rectangle(0,0,620,434,fill="gray",tag="フィールド")
```

#ボタン配置

```
ボタン上=tkinter.Button(text="↑")
ボタン上.place(x=720,y=150)
ボタン上["command"]=クリックボタン上
ボタン下=tkinter.Button(text="↓")
ボタン下.place(x=720,y=210)
ボタン下["command"]=クリックボタン下
```

次のページへ続く

キャラクターを動かす

```
ボタン左=tkinter.Button(text="←")
```

```
ボタン左.place(x=660,y=180)
```

```
ボタン左["command"]=クリックボタン左
```

```
ボタン右=tkinter.Button(text="→")
```

```
ボタン右.place(x=780,y=180)
```

```
ボタン右["command"]=クリックボタン右
```

#画像データを読み込み

```
画像=[tkinter.PhotoImage(file="フィールド.png"),  
       tkinter.PhotoImage(file="岩.png"),  
       tkinter.PhotoImage(file="ゴール.png"),  
       tkinter.PhotoImage(file="カギ.png"),  
       tkinter.PhotoImage(file="勇者.png")]
```

#マップデータ

```
map=[[1,0,1,1,1,1,1,1,1],  
     [1,0,0,1,2,0,0,1,3,1],
```

次のページへ続く

キャラクターを動かす

```
[1,1,0,1,1,1,0,1,0,1],  
[1,0,0,0,0,0,0,1,0,1],  
[1,0,1,1,1,1,1,1,0,1],  
[1,0,0,0,0,0,0,0,0,1],  
[1,1,1,1,1,1,1,1,1,1]]
```

#主人公の位置

```
brave_x = 1
```

```
brave_y = 0
```

#ダンジョン()関数を呼び出す

```
ダンジョン()
```

#mainloop()関数を実行してウィンドウを表示する

```
ウィンドウ.mainloop()
```

キャラクターを動かす

「Run Module」またはF5キーを押してプログラムを実行する。



各ボタンをクリックすると移動する

キャラクターを動かす

マップ外に行かないように移動範囲を制限する。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p = map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
#主人公を表示させる
```

```
    キャンバス.create_image(brave_x*62+31,brave_y*62+31,image=画像[4],  
    tag="勇者")
```

```
#移動先のxとy座標がマップの範囲内かをチェック
```

```
def チェック(x,y):
```

```
    global brave_x,brave_y
```

```
    if x >= 0 and x < 10 and y >= 0 and y < 7:
```

```
        brave_x = x
```

```
        brave_y = y
```

```
        キャンバス.coords("勇者",brave_x*62+31,brave_y*62+31)
```

次のページへ続く

キャラクターを動かす

#ボタンが押された時の関数を定義

```
def クリックボタン上():
```

```
    チェック(brave_x,brave_y-1)
```

```
def クリックボタン下():
```

```
    チェック(brave_x,brave_y+1)
```

```
def クリックボタン左():
```

```
    チェック(brave_x-1,brave_y)
```

```
def クリックボタン右():
```

```
    チェック(brave_x+1,brave_y)
```

#ウィンドウ作成

```
ウィンドウ=tkinter.Tk()
```

```
ウィンドウ.title("ダンジョン&パイソン")
```

```
ウィンドウ.minsize(840,454)
```

```
ウィンドウ.option_add("*font",["メイリオ",14])
```

#キャンバス作成

```
キャンバス=tkinter.Canvas(width=620,height=434)
```

次のページへ続く

キャラクターを動かす

```
キャンバス.place(x=10,y=10)
```

```
キャンバス.create_rectangle(0,0,620,434,fill="gray",  
tag="フィールド")
```

#ボタン配置

```
ボタン上=tkinter.Button(text="↑")
```

```
ボタン上.place(x=720,y=150)
```

```
ボタン上["command"]=クリックボタン上
```

```
ボタン下=tkinter.Button(text="↓")
```

```
ボタン下.place(x=720,y=210)
```

```
ボタン下["command"]=クリックボタン下
```

```
ボタン左=tkinter.Button(text="←")
```

```
ボタン左.place(x=660,y=180)
```

```
ボタン左["command"]=クリックボタン左
```

```
ボタン右=tkinter.Button(text="→")
```

```
ボタン右.place(x=780,y=180)
```

```
ボタン右["command"]=クリックボタン右
```

次のページへ続く

キャラクターを動かす

#画像データを読み込み

```
画像=[tkinter.PhotoImage(file="フィールド.png"),  
      tkinter.PhotoImage(file="岩.png"),  
      tkinter.PhotoImage(file="ゴール.png"),  
      tkinter.PhotoImage(file="カギ.png"),  
      tkinter.PhotoImage(file="勇者.png")]
```

#マップデータ

```
map=[[1,0,1,1,1,1,1,1,1],  
     [1,0,0,1,2,0,0,1,3,1],  
     [1,1,0,1,1,1,0,1,0,1],  
     [1,0,0,0,0,0,0,1,0,1],  
     [1,0,1,1,1,1,1,1,0,1],  
     [1,0,0,0,0,0,0,0,0,1],  
     [1,1,1,1,1,1,1,1,1,1]]
```

#主人公の位置

```
brave_x = 1  
brave_y = 0
```

次のページへ続く

キャラクターを動かす

#ダンジョン()関数を呼び出す

ダンジョン()

#mainloop()関数を実行してウィンドウを表示する

ウィンドウ.mainloop()

「Run Module」またはF5キーを押してプログラムを実行する。



マップの外側には出れなくなっていることを確認。
※まだ岩の上には移動する。

キャラクターを動かす

岩の上に行かないように移動範囲を制限する。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p = map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
#主人公を表示させる
```

```
    キャンバス.create_image(brave_x*62+31,brave_y*62+31,image=画像[4],tag="勇者")
```

```
#移動先のxとy座標がマップの範囲内かをチェック
```

```
def チェック(x,y):
```

```
    global brave_x,brave_y
```

```
    if x >= 0 and x < 10 and y >= 0 and y < 7:
```

```
        #mapのリストの中から移動先の位置の要素を取り出す
```

```
        p = map[y][x]
```

次のページへ続く

キャラクターを動かす

#取り出した要素が1ならreturnで関数から脱出

```
if p == 1:
```

```
    return
```

```
    brave_x = x
```

```
    brave_y = y
```

```
    キャンバス.coords("勇者",brave_x*62+31,brave_y*62+31)
```

#ボタンが押された時の関数を定義

```
def クリックボタン上():
```

```
    チェック(brave_x,brave_y-1)
```

```
def クリックボタン下():
```

```
    チェック(brave_x,brave_y+1)
```

```
def クリックボタン左():
```

```
    チェック(brave_x-1,brave_y)
```

```
def クリックボタン右():
```

```
    チェック(brave_x+1,brave_y)
```

#ウィンドウ作成

```
ウィンドウ=tkinter.Tk()
```

次のページへ続く

キャラクターを動かす

```
ウィンドウ.title("ダンジョン&パイソン")
ウィンドウ.minsize(840,454)
ウィンドウ.option_add("*font",["メイリオ",14])
#キャンバス作成
キャンバス=tkinter.Canvas(width=620,height=434)
キャンバス.place(x=10,y=10)
キャンバス.create_rectangle(0,0,620,434,fill="gray",
tag="フィールド")
#ボタン配置
ボタン上=tkinter.Button(text="↑")
ボタン上.place(x=720,y=150)
ボタン上["command"]=クリックボタン上
ボタン下=tkinter.Button(text="↓")
ボタン下.place(x=720,y=210)
ボタン下["command"]=クリックボタン下
ボタン左=tkinter.Button(text="←")
```

次のページへ続く

キャラクターを動かす

```
ボタン左.place(x=660,y=180)
```

```
ボタン左["command"]=クリックボタン左
```

```
ボタン右=tkinter.Button(text="→")
```

```
ボタン右.place(x=780,y=180)
```

```
ボタン右["command"]=クリックボタン右
```

#画像データを読み込み

```
画像=[tkinter.PhotoImage(file="フィールド.png"),
```

```
tkinter.PhotoImage(file="岩.png"),
```

```
tkinter.PhotoImage(file="ゴール.png"),
```

```
tkinter.PhotoImage(file="カギ.png"),
```

```
tkinter.PhotoImage(file="勇者.png")]
```

#マップデータ

```
map=[[1,0,1,1,1,1,1,1,1],
```

```
1,0,0,1,2,0,0,1,3,1],
```

```
1,1,0,1,1,1,0,1,0,1],
```

```
1,0,0,0,0,0,0,1,0,1],
```

次のページへ続く

キャラクターを動かす

```
[1,0,1,1,1,1,1,1,0,1],  
[1,0,0,0,0,0,0,0,0,1],  
[1,1,1,1,1,1,1,1,1,1]]
```

#主人公の位置

```
brave_x = 1
```

```
brave_y = 0
```

#ダンジョン()関数を呼び出す

```
ダンジョン()
```

#mainloop()関数を実行してウィンドウを表示する

```
ウィンドウ.mainloop()
```

キャラクターを動かす

「Run Module」またはF5キーを押してプログラムを実行する。



岩があるところは移動しないことを確認

ゲームをクリアできるようにする

鍵を拾ってゴールできるようにする。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p = map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
#主人公を表示させる
```

```
    キャンバス.create_image(brave_x*62+31,brave_y*62+31,image=画像[4],tag="勇者")
```

```
#移動先のxとy座標がマップの範囲内かをチェック
```

```
def チェック(x,y):
```

```
    global brave_x,brave_y,フラグ
```

次のページへ続く

ゲームをクリアできるようにする

```
if x >= 0 and x < 10 and y >= 0 and y < 7:
```

```
#map_dataのリストの中から移動先の位置の要素を取り出す
```

```
p = map[y][x]
```

```
#取り出した要素が1ならreturnで関数から脱出
```

```
if p == 1:
```

```
    return
```

```
#移動先が鍵(3)だったらフラグにTrueを代入,mapを書き替える
```

```
elif p == 3:
```

```
    フラグ = True
```

```
    map[y][x] = 0
```

```
    キャンバス.delete("all")
```

```
    ダンジョン()
```

```
    brave_x = x
```

```
    brave_y = y
```

次のページへ続く

ゲームをクリアできるようにする

```
キャンバス.coords("勇者",brave_x*62+31,brave_y*62+31)
```

#ボタンが押された時の関数を定義

#上ボタンが押されたとき

```
def クリックボタン上():
```

```
    チェック(brave_x,brave_y-1)
```

#下ボタンが押されたとき

```
def クリックボタン下():
```

```
    チェック(brave_x,brave_y+1)
```

#左ボタンが押されたとき

```
def クリックボタン左():
```

```
    チェック(brave_x-1,brave_y)
```

#右ボタンが押されたとき

```
def クリックボタン右():
```

```
    チェック(brave_x+1,brave_y)
```

次のページへ続く

ゲームをクリアできるようにする

#ウィンドウ作成

```
ウィンドウ=tkinter.Tk()
```

```
ウィンドウ.title("ダンジョン&パイソン")
```

```
ウィンドウ.minsize(840,454)
```

```
ウィンドウ.option_add("*font",["メイリオ",14])
```

#キャンバス作成

```
キャンバス=tkinter.Canvas(width=620,height=434)
```

```
キャンバス.place(x=10,y=10)
```

```
キャンバス.create_rectangle(0,0,620,434,fill="gray",tag="フィールド")
```

#ボタン配置

```
ボタン上=tkinter.Button(text="↑")
```

```
ボタン上.place(x=720,y=150)
```

```
ボタン上["command"]=クリックボタン上
```

```
ボタン下=tkinter.Button(text="↓")
```

次のページへ続く

ゲームをクリアできるようにする

ボタン下.place(x=720,y=210)

ボタン下["command"]=クリックボタン下

ボタン左=tkinter.Button(text="←")

ボタン左.place(x=660,y=180)

ボタン左["command"]=クリックボタン左

ボタン右=tkinter.Button(text="→")

ボタン右.place(x=780,y=180)

ボタン右["command"]=クリックボタン右

#画像データを読み込み

画像=[tkinter.PhotoImage(file="フィールド.png"),

tkinter.PhotoImage(file="岩.png"),

tkinter.PhotoImage(file="ゴール.png"),

tkinter.PhotoImage(file="カギ.png"),

tkinter.PhotoImage(file="勇者.png")]

次のページへ続く

ゲームをクリアできるようにする

#マップデータ

```
map=[[1,0,1,1,1,1,1,1,1,1],  
      [1,0,0,1,2,0,0,1,3,1],  
      [1,1,0,1,1,1,0,1,0,1],  
      [1,0,0,0,0,0,0,1,0,1],  
      [1,0,1,1,1,1,1,1,0,1],  
      [1,0,0,0,0,0,0,0,0,1],  
      [1,1,1,1,1,1,1,1,1,1]]
```

#主人公の位置

```
brave_x = 1
```

```
brave_y = 0
```

#鍵取得フラグ(スタート時は鍵を拾っていないのでFalseを代入しておく)

```
フラグ = False
```

次のページへ続く

ゲームをクリアできるようにする

#ダンジョン()関数を呼び出す

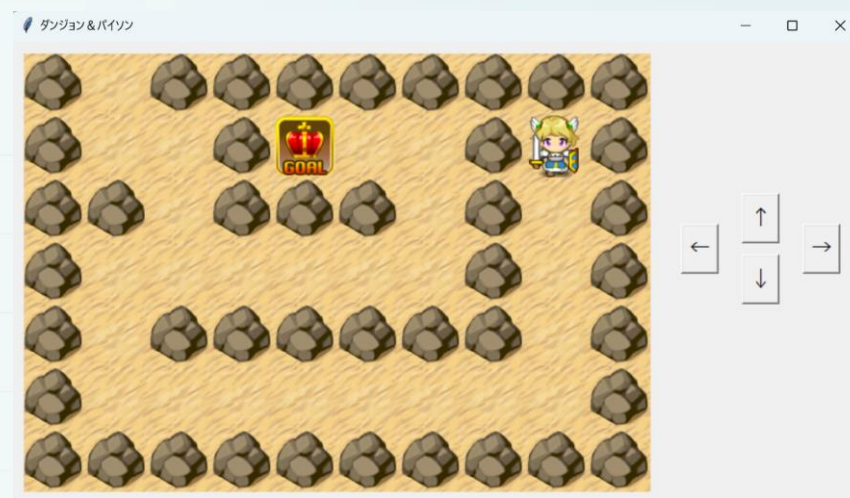
ダンジョン()

#mainloop()関数を実行してウィンドウを表示する

ウィンドウ.mainloop()

ゲームをクリアできるようにする

「Run Module」またはF5キーを押してプログラムを実行する。



鍵に重なってから移動すると
鍵が消えている

ゲームをクリアできるようにする

ゴールに到着したときにフラグを判定する。

```
import tkinter
```

```
#マップの描画
```

```
def ダンジョン():
```

```
    for y in range(7):
```

```
        for x in range(10):
```

```
            p = map[y][x]
```

```
            キャンバス.create_image(x*62+31,y*62+31,image=画像[p])
```

```
#主人公を表示させる
```

```
    キャンバス.create_image(brave_x*62+31,brave_y*62+31,image=画像[4],tag="勇者")
```

```
#移動先のxとy座標がマップの範囲内かをチェック
```

```
def チェック(x,y):
```

```
    global brave_x,brave_y,フラグ
```

次のページへ続く

ゲームをクリアできるようにする

```
if x >= 0 and x < 10 and y >= 0 and y < 7:
```

```
    #mapのリストの中から移動先の位置の要素を取り出す
```

```
    p = map[y][x]
```

```
    #取り出した要素が1ならreturnで関数から脱出
```

```
    if p == 1:
```

```
        return
```

```
    #移動先が鍵(3)だったらフラグにTrueを代入,mapを書き替えて
```

```
    ダンジョン関数を呼び出す
```

```
elif p == 3:
```

```
    フラグ = True
```

```
    map[y][x] = 0
```

```
    キャンバス.delete("all")
```

```
    ダンジョン()
```

次のページへ続く

ゲームをクリアできるようにする

#移動先がゴール(2)だったらフラグがTrueかチェックする

#Trueだったらエンディング関数を呼び出す

```
elif p == 2:
```

```
    if フラグ == True:
```

```
        エンディング()
```

```
    else:
```

```
        return
```

```
    brave_x = x
```

```
    brave_y = y
```

```
    キャンバス.coords("勇者",brave_x*62+31,brave_y*62+31)
```

#ボタンが押された時の関数を定義

#上ボタンが押されたとき

```
def クリックボタン上():
```

```
    チェック(brave_x,brave_y-1)
```

次のページへ続く

ゲームをクリアできるようにする

#下ボタンが押されたとき

```
def クリックボタン下():  
    チェック(brave_x,brave_y+1)
```

#左ボタンが押されたとき

```
def クリックボタン左():  
    チェック(brave_x-1,brave_y)
```

#右ボタンが押されたとき

```
def クリックボタン右():  
    チェック(brave_x+1,brave_y)
```

#エンディング関数を作成

```
def エンディング():  
    キャンバス.delete("all")  
    キャンバス.create_rectangle(0,0,620,434,fill="black")  
    キャンバス.create_text(300,200,fill="white",font=("MS ゴシック",  
15),text=""ゴールおめでとう。)
```

次のページへ続く

ゲームをクリアできるようにする

だが、君の戦いはまだ始まったばかりだ。

．．．．．つづく?""")

#ボタンを無効化

ボタン上["state"]="disabled"

ボタン下["state"]="disabled"

ボタン左["state"]="disabled"

ボタン右["state"]="disabled"

#ウィンドウ作成

ウィンドウ=tkinter.Tk()

ウィンドウ.title("ダンジョン&パイソン")

ウィンドウ.minsize(840,454)

ウィンドウ.option_add("*font",["メイリオ",14])

#キャンバス作成

キャンバス=tkinter.Canvas(width=620,height=434)

次のページへ続く

ゲームをクリアできるようにする



```
キャンバス.place(x=10,y=10)
```

```
キャンバス.create_rectangle(0,0,620,434,fill="gray",tag="フィールド")
```

#ボタン配置

```
ボタン上=tkinter.Button(text="↑")
```

```
ボタン上.place(x=720,y=150)
```

```
ボタン上["command"]=クリックボタン上
```

```
ボタン下=tkinter.Button(text="↓")
```

```
ボタン下.place(x=720,y=210)
```

```
ボタン下["command"]=クリックボタン下
```

```
ボタン左=tkinter.Button(text="←")
```

```
ボタン左.place(x=660,y=180)
```

```
ボタン左["command"]=クリックボタン左
```

```
ボタン右=tkinter.Button(text="→")
```

```
ボタン右.place(x=780,y=180)
```

次のページへ続く

ゲームをクリアできるようにする

ボタン右["command"]=クリックボタン右

#画像データを読み込み

```
画像=[tkinter.PhotoImage(file="フィールド.png"),  
      tkinter.PhotoImage(file="岩.png"),  
      tkinter.PhotoImage(file="ゴール.png"),  
      tkinter.PhotoImage(file="カギ.png"),  
      tkinter.PhotoImage(file="勇者.png")]
```

#マップデータ

```
map=[[1,0,1,1,1,1,1,1,1,1],  
     [1,0,0,1,2,0,0,1,3,1],  
     [1,1,0,1,1,1,0,1,0,1],  
     [1,0,0,0,0,0,0,1,0,1],  
     [1,0,1,1,1,1,1,1,0,1],  
     [1,0,0,0,0,0,0,0,0,1],
```

次のページへ続く

ゲームをクリアできるようにする

```
[1,1,1,1,1,1,1,1,1,1]
```

#主人公の位置

```
brave_x = 1
```

```
brave_y = 0
```

#鍵取得フラグ(スタート時は鍵を拾っていないのでFalseを代入しておく)

```
フラグ = False
```

#ダンジョン()関数を呼び出す

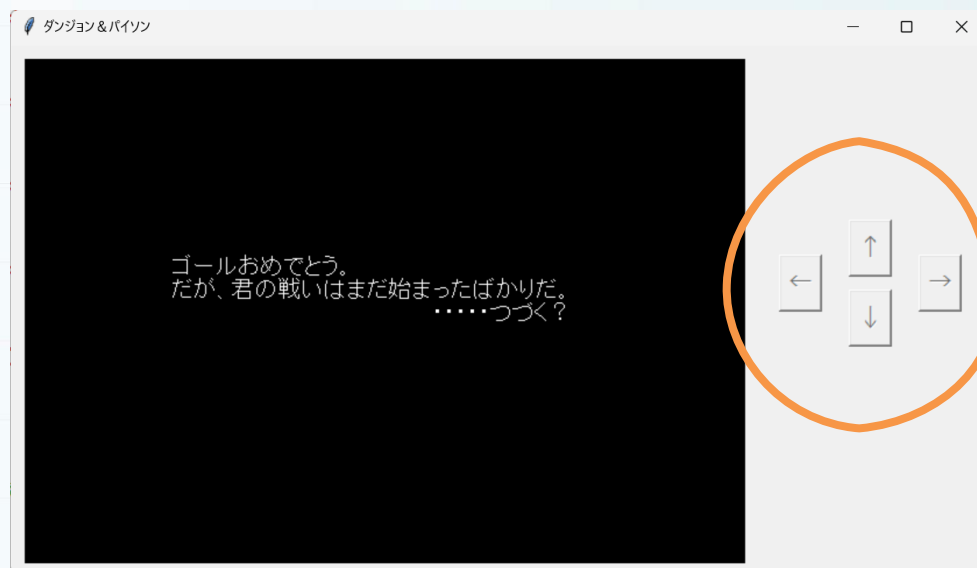
```
ダンジョン()
```

#mainloop()関数を実行してウィンドウを表示する

```
ウィンドウ.mainloop()
```

ゲームをクリアできるようにする

「Run Module」またはF5キーを押してプログラムを実行する。



ボタンは押せなくなる

プログラミング教室の テクノロ

なまえ：