

情報 I

4-1 Python作法1

4-1-8 反復 for文、while文



4-1-8 反復


① for文 for **変数** in range(**回数**) ~ 処理
for **n** in range(10): # 変数nを10回繰返し、**最後にコロン:必須**
 print(n) # nを表示せよ、**改行後インデント(字下げ)必須**

for n in range(10): # 変数nを10回繰返し
 print(n+1) # n+1を表示せよ

for n in range(1, 10, 3): # 変数nを1~10未満の間で3毎繰返し
 print(n) # nを表示せよ


- ポイント ➡ 反復for文では条件末尾にコロン: 必須
➡ 反復条件内の処理はインデント(字下げ) 必須
➡ rangeの扱い... range(start, stop[, step])
➡ start ~ stop未満でstepごとに連続した数値を返す
start省略→0、step 省略→1






```
for n in range(10):  
    print(n)
```

0
1
2
3
4
5
6
7
8
9



```
for n in range(10):  
    print(n+1)
```

↳ 1
2
3
4
5
6
7
8
9
10



```
for n in range(1, 10, 3):  
    print(n)
```

↳ 1
4
7



② while文 初期値~while 反復条件~処理~繰り返し

```
n=0
while n<10:
    print(n)
    n=n+1
```

変数n 初期値0
変数nを10未満で繰り返し 最後にコロン:必須
nを表示せよ、改行後インデント(字下げ)必須
nに1を加えて処理を繰り返す

```
n=0
while n<10:
    print(n+1)
    n=n+1
```

変数n 初期値0
変数nを10未満で繰り返し 最後にコロン:必須
n+1を表示せよ、改行後インデント(字下げ)必須
nに1を加えて処理を繰り返す

```
n=1
while n<10:
    print(n)
    n=n+3
```

変数n 初期値1
変数nを10未満で繰り返し 最後にコロン:必須
nを表示せよ、改行後インデント(字下げ)必須
nに3を加えて処理を繰り返す





```
n=0
while n<10:
    print(n)
    n=n+1
```



0
1
2
3
4
5
6
7
8
9



```
n=0
while n<10:
    print(n+1)
    n=n+1
```



1
2
3
4
5
6
7
8
9
10



```
n=1
while n<10:
    print(n)
    n=n+3
```



1
4
7



例題 for文 1~10未満の範囲で偶数のみ表記

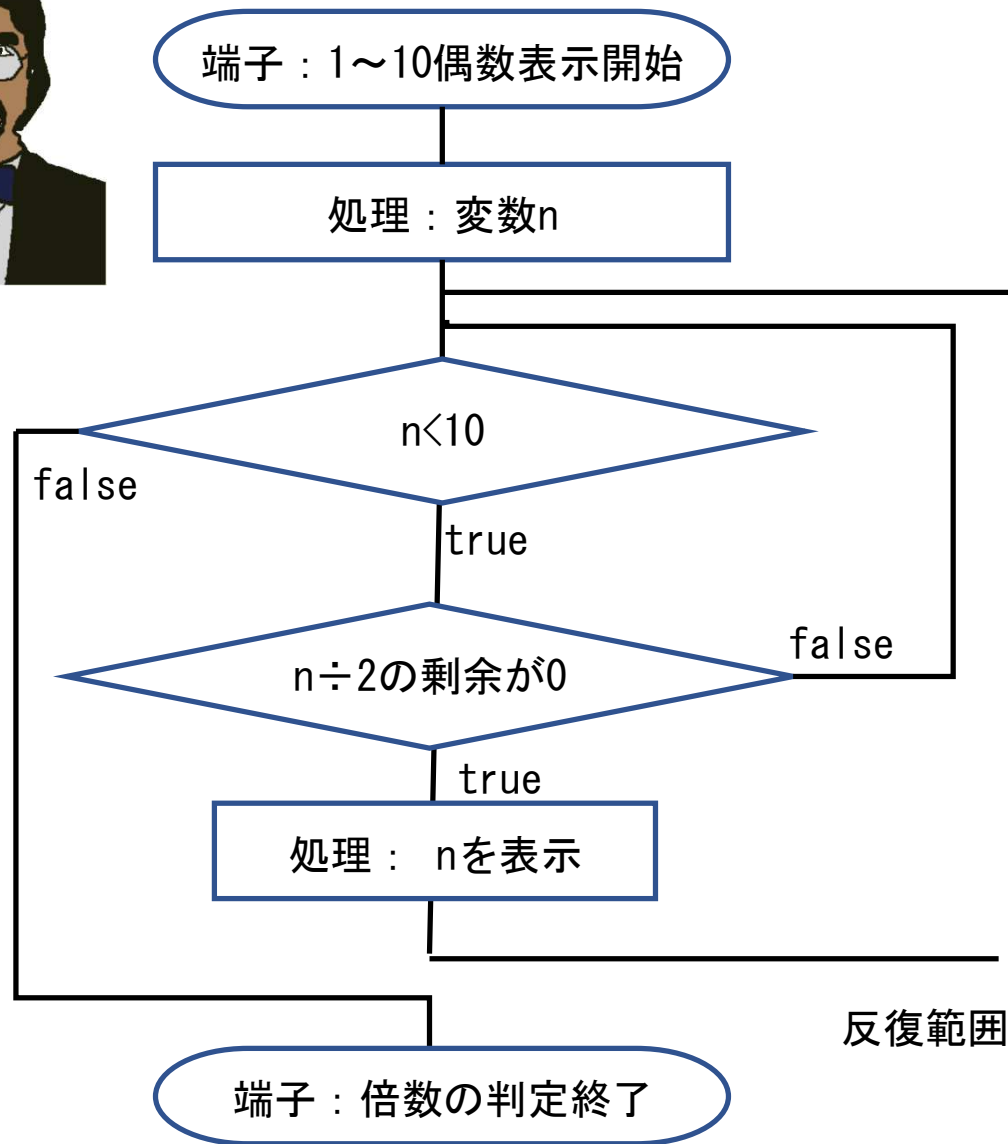
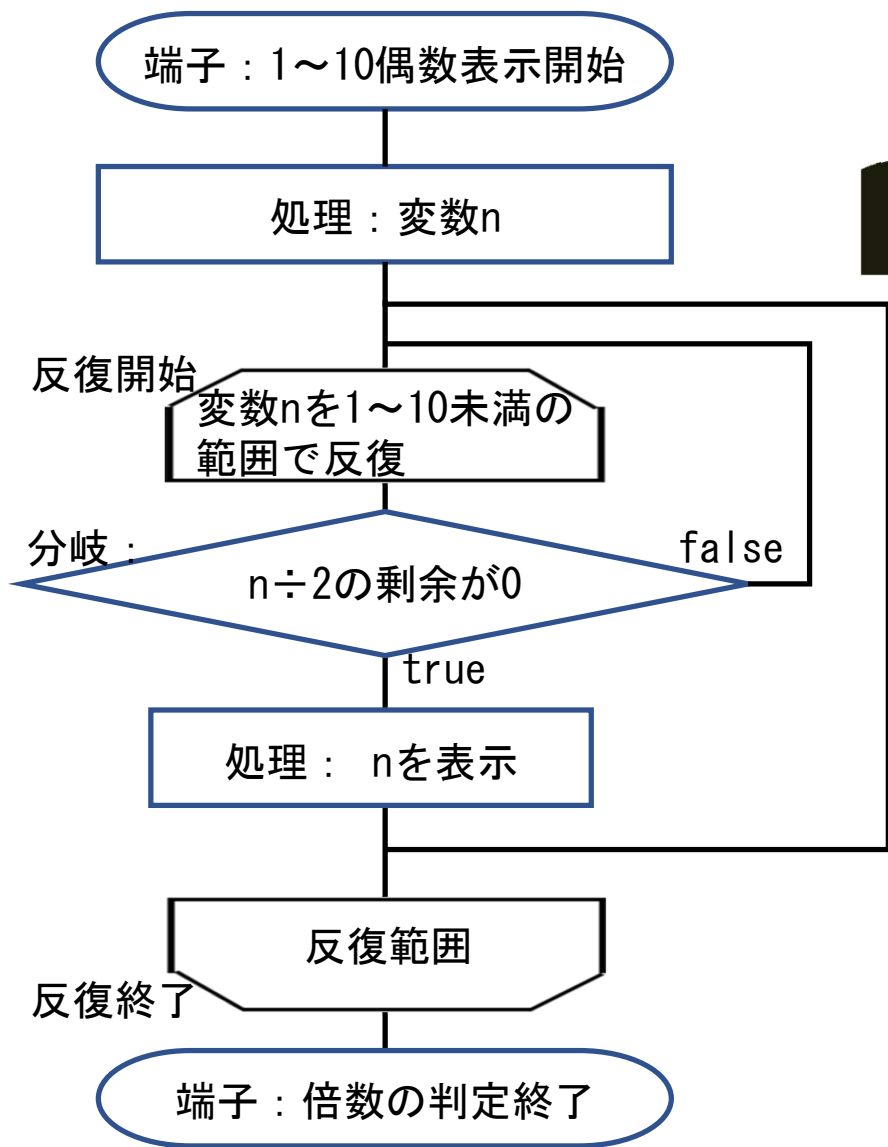
➡2, 4, 6, 8

```
for n in range(1, 10): # 変数nを1~10未満の範囲で反復
    if n%2==0:         # 変数nを2で割った余りが0ならば
        print(n)      # 変数nを表示せよ
```



```
for n in range(1, 10):
    if n%2==0:
        print(n)
```





例題 while文 0~20以下の範囲で初項0, 公差6の数列表記
→0, 6, 12, 18

```
n=0
k=6
while n<=20:
    print(n)
    n=n+k
```

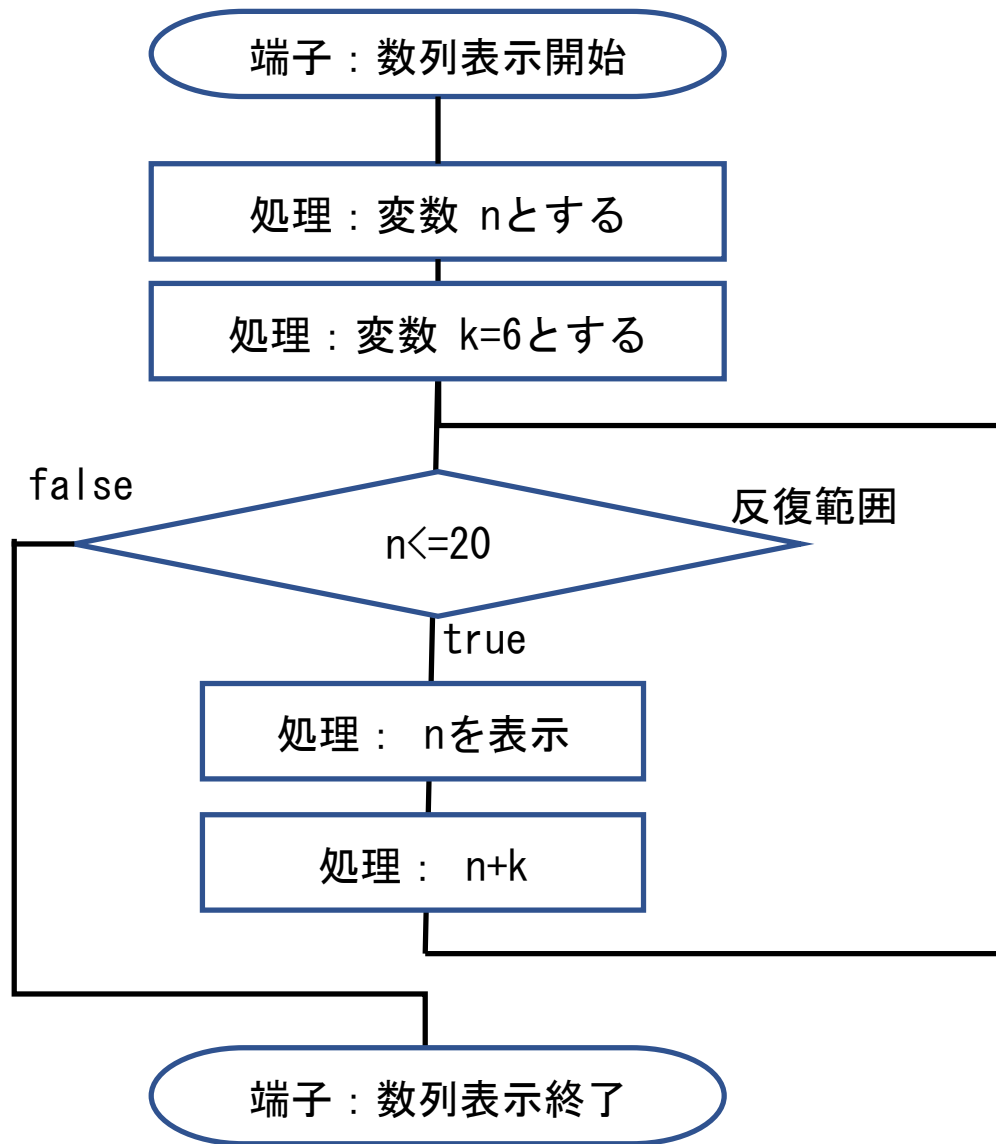
変数n 初期値0
公差k=6とする
変数n を20以下で繰り返し
変数n を表示
変数n にk=6を加えて処理



```
n=0
k=6
while n<=20:
    print(n)
    n=n+k
```

☞
0
6
12
18





例題 while文 1~10未満の範囲で偶数のみ表記
➡2, 4, 6, 8

```
n=0          # 変数n 初期値0
while n<10:  # 変数n を10未満で繰り返し
    if n%2==0: # 変数n を2で割った余りが0ならば
        print(n) # 変数n を表示
        n=n+1    # 変数n に1を加えて処理
```



③ 反復forの入れ子 掛け算九九

```
for m in range(1, 10):          # 変数 m は1~9 (9回) 繰返し
    for n in range(1, 10):      # 変数 n は1~9 (9回) 繰返し
        print(m, ' × ', n, ' = ', m*n) # 右辺は (1~9)*(1~9) 演算
```



```
for m in range(1, 10):
    for n in range(1, 10):
        print(m, ' × ', n, ' = ', m*n)
```

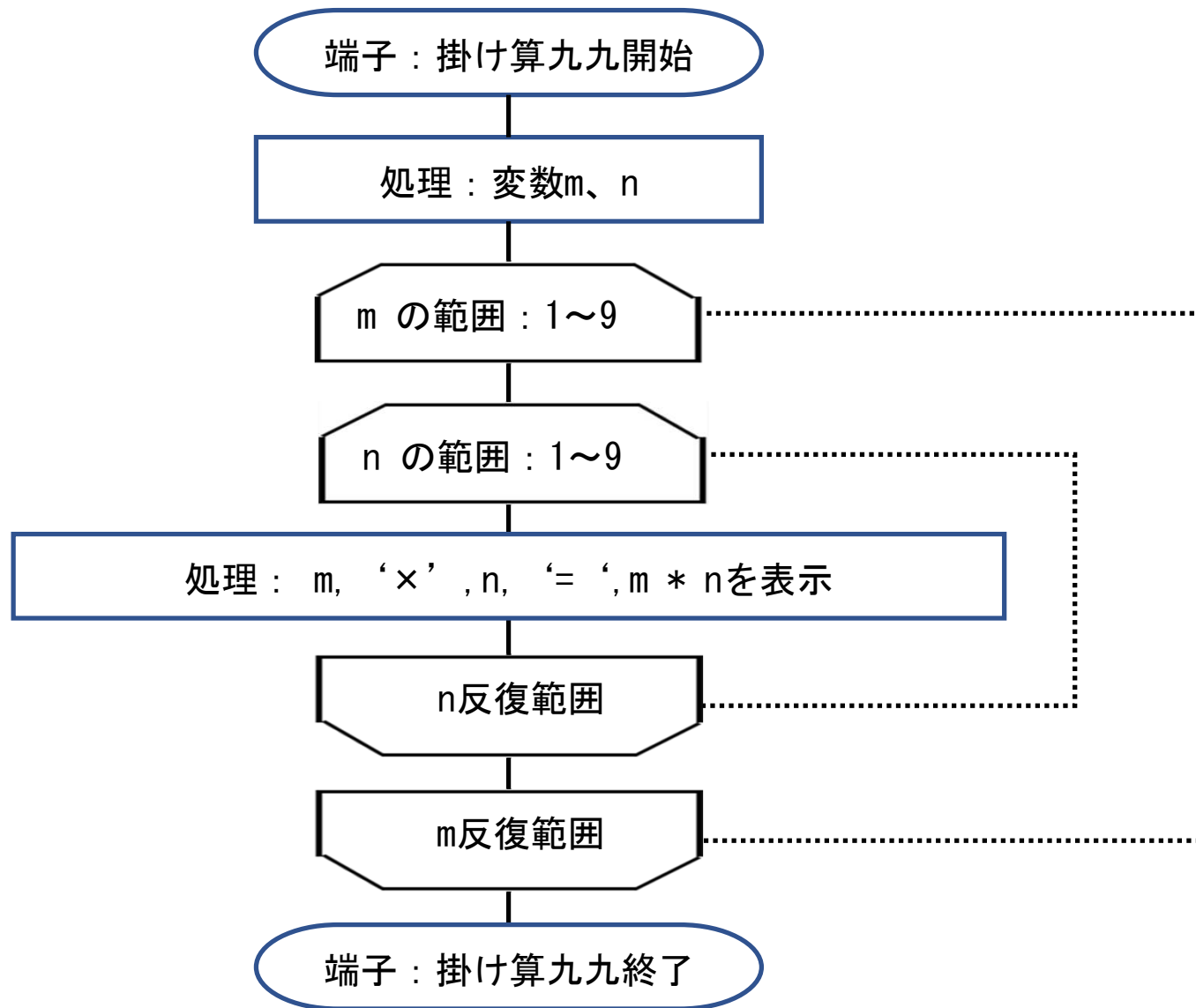
↳

1	×	1	=	1
1	×	2	=	2
1	×	3	=	3
1	×	4	=	4



9	×	2	=	18
9	×	3	=	27
9	×	4	=	36
9	×	5	=	45
9	×	6	=	54
9	×	7	=	63
9	×	8	=	72
9	×	9	=	81





③ 反復whileの入れ子 掛け算九九

```
m=1 # 変数 m 初期値1
while m<=9: # 変数 m を9以下で繰り返す
    n=1 # 変数 n 初期値1
    while n<=9: # 変数 n を9以下で繰り返す
        print(m, '*', n, '=', m*n) # n, '*', m, '=', m*n 表示
        n=n+1 # 変数 n に1を加えて反復
    m=m+1 # 変数 m に1を加えて反復
```





```
m=1
while m<=9:
    n=1
    while n<=9:
        print(m, '*', n, '=', m*n)
        n=n+1
    m=m+1
```

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
```



```
8 * 8 = 64
8 * 9 = 72
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
```



例題 while文 1~10未満の範囲で偶数のみ表記
➡2, 4, 6, 8

```
n=0          # 変数n 初期値0
while n<10:  # 変数n を10未満で繰り返し
    if n%2==0: # 変数n を2で割った余りが0ならば
        print(n) # 変数n を表示
        n=n+1    # 変数n に1を加えて処理
```



④ While による反復の強制終了と前判定・後判定

```
x = 0          # 変数x 初期値0
while x <= 20 : # 変数x を20以下で繰り返し
    print(x)   # xを表示
    x = x + 3  # 変数x に3を加えて繰り返し
    if x > 12: # xが12を超えた場合
        break  # 反復処理を終了する
print('fin')  # 'fin' 表示
```

```
x=0
while x<=20:
    print(x)
    x=x+3
    if x>12:
        break
print('fin')
```

whileは、条件が**true, false**で反復(ループ)を回すか否かを決定する。**true, false**を求める式を**論理式**という。処理の前に条件を設定することを**前判定**という。本例の12を超えると反復を**break**させる例のように、反復中に実行条件や終了条件を判断することも可能で、これを**後判定**という。

```
↳
0
3
6
9
12
fin
```



⑤ 後判定例題 1から10までの数を積算

```
s=0          # 合計値を変数sとし、初期値0
n=1          # 加える数を変数nとし、初期値1
while True:  # 後の設定する条件になるまで反復
    s=s+n    # 合計値初期値s=0にnの初期値1を加算
    n=n+1    # 次にnに1を加えよ
    if n<=10: # nが10以下では
        continue # 処理を継続
    else:     # 10を超えたら
        break    # 反復を終了
print(s)     # 合計値(1+2+3+4+5+6+7+8+9+10)を表示
```

```
s=0
n=1
while True:
    s=s+n
    n=n+1
    if n<=10:
        continue
    else:
        break
print(s)
```

$$s=45+10=55$$

$$n=9+1=10$$

55

このケースで、後判定のbreak条件がないとプログラムは暴走する⇒無限ループ。

