



プログラミング言語 第六回

担当: 篠沢 佳久
栗原 聡

2019年 5月20日



本日の内容

- 繰り返し(2)
 - 回数の決まった繰り返し(復習)
 - 無限の繰り返し
 - while True:
- 繰り返しについての練習問題

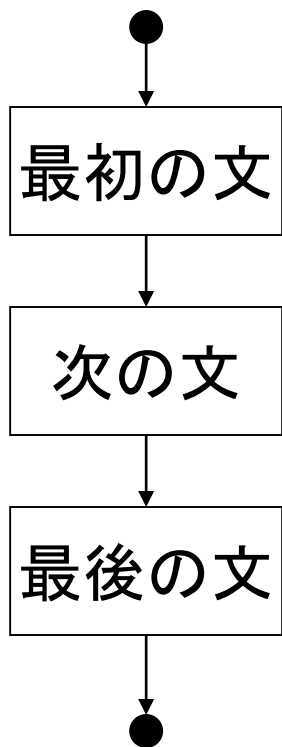


前回の復習

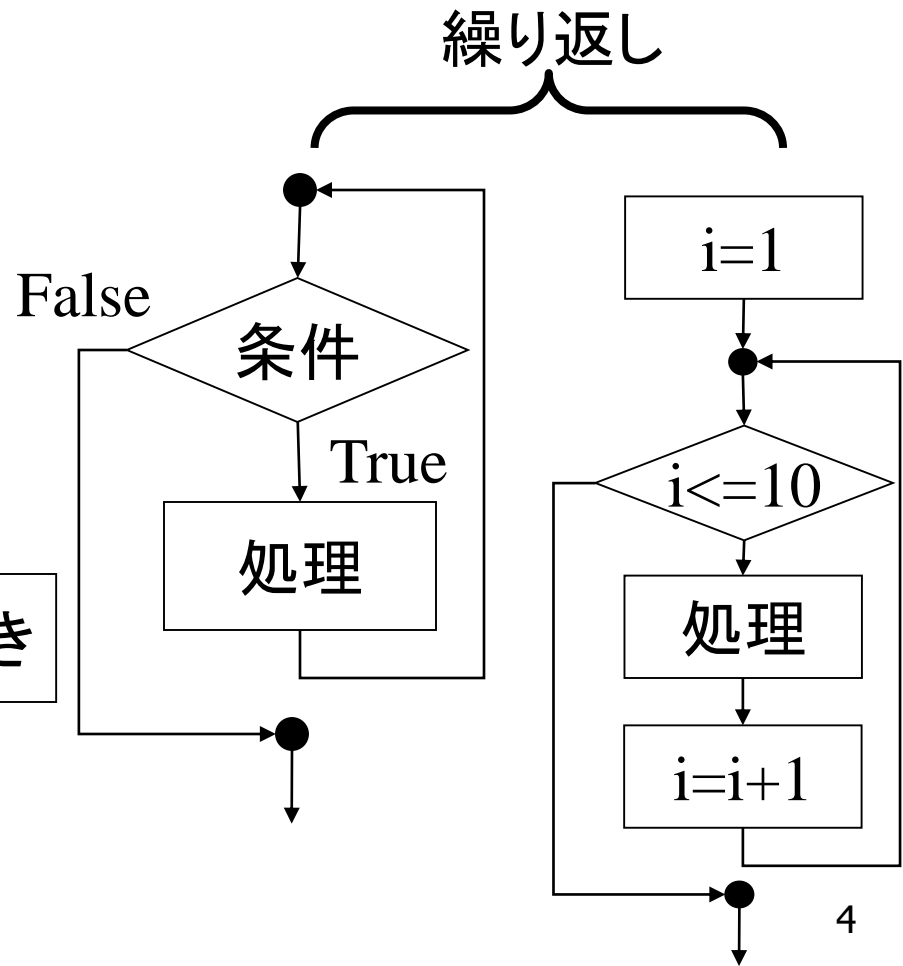
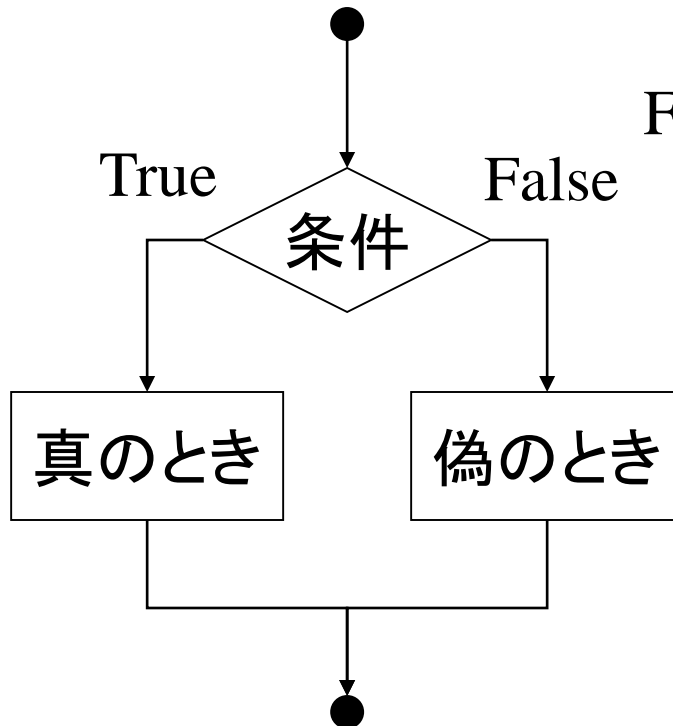
制御構造
繰り返し(1)

制御構造(復習)

文の並び(接続)



If文(分岐)





回数の決まった繰り返し

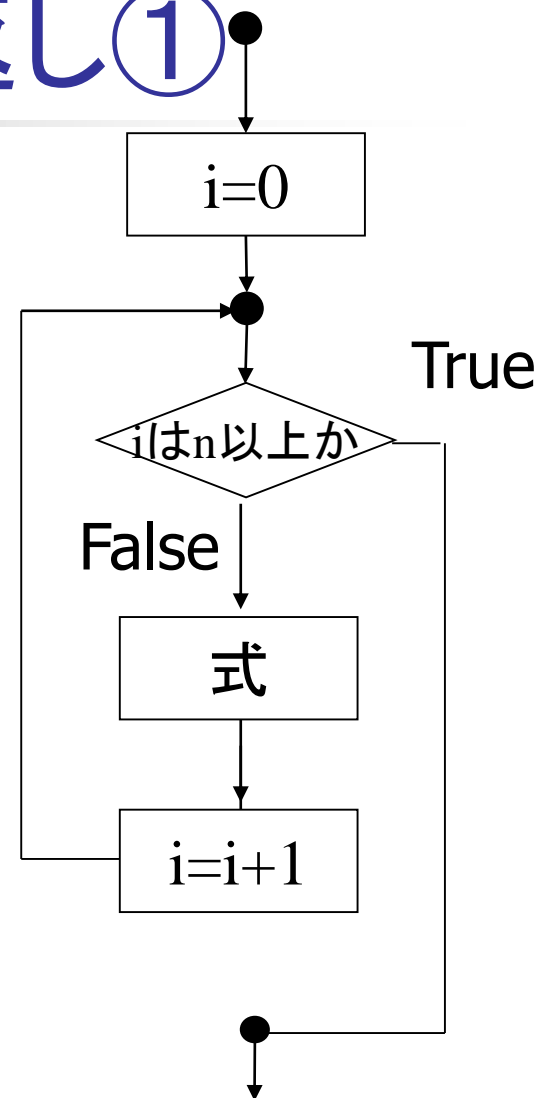
for文

回数の決まった繰り返し①

```
for i in range(n):  
    式
```

n回、式が繰り返される

変数iには0からn-1まで代入される





回数の決まった繰り返し②

10回繰り返す

```
for i in range(10)
    print( "やっほ～ " )
    print( "          Yee-ha! " )
```

iには0,1,2,...,9が代入される

```
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
やっほ～
    Yee-ha!
```



回数の決まった繰り返し③

「こんにちは」を10回表示させるプログラム

```
for i in range(10):  
    print( i , "回目のこんにちは" )
```

*i*には0,1,2,...,9が代入される

*i*は1,2,3,...,10ではないことに
注意して下さい

```
>python sample.py  
0 回目のこんにちは  
1 回目のこんにちは  
2 回目のこんにちは  
3 回目のこんにちは  
4 回目のこんにちは  
5 回目のこんにちは  
6 回目のこんにちは  
7 回目のこんにちは  
8 回目のこんにちは  
9 回目のこんにちは
```


回数の決まった繰り返し③'

```
for i in range(10):  
    print( i , "回目のこんにちは" )
```

```
>python sample.py  
0 回目のこんにちは  
1 回目のこんにちは  
2 回目のこんにちは  
3 回目のこんにちは  
4 回目のこんにちは  
5 回目のこんにちは  
6 回目のこんにちは  
7 回目のこんにちは  
8 回目のこんにちは  
9 回目のこんにちは
```

変数*i*には0から代入されていく

*i*は0,1,2,...,9

変数*i*には9まで代入されていく

回数の決まった繰り返し④

「1～10」回目のこんにちはと表示させるには？

```
for i in range(10):  
    print( i+1 , "回目のこんにちは" )
```

```
>python sample.py  
1回目のこんにちは  
2回目のこんにちは  
3回目のこんにちは  
4回目のこんにちは  
5回目のこんにちは  
6回目のこんにちは  
7回目のこんにちは  
8回目のこんにちは  
9回目のこんにちは  
10回目のこんにちは
```

変数*i*には0から代入されていく
i+1 → 「1回目のこんにちは」

変数*i*には9まで代入されていく
i+1 → 「10回目のこんにちは」

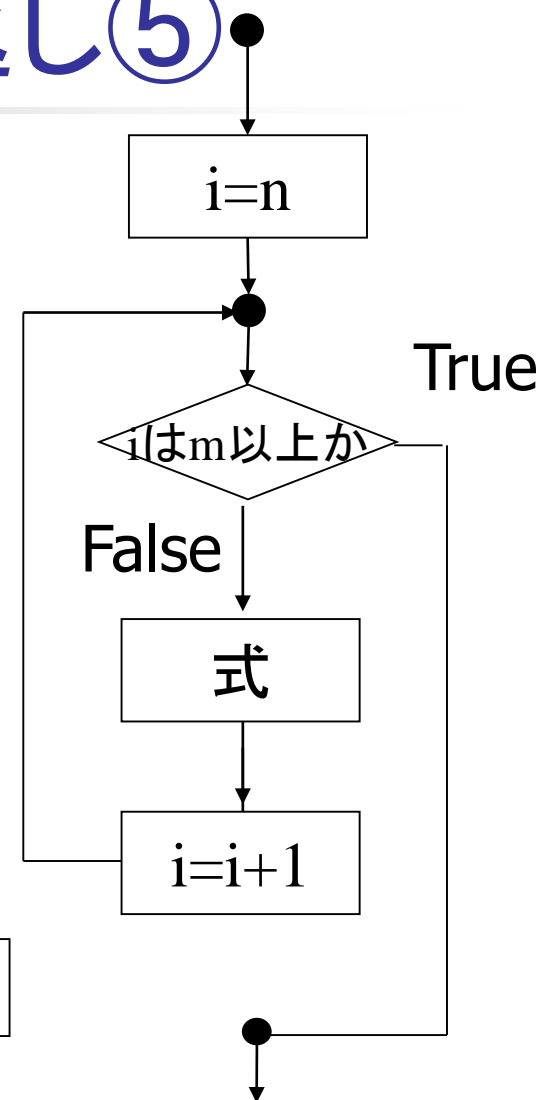
回数の決まった繰り返し⑤

```
for i in range(n,m):  
    式
```

(m-n-1)回, 式が繰り返される

変数iにはnからm-1まで代入される

mまでではないことに注意して下さい



回数の決まった繰り返し⑥

「1～10」回目のこんにちはと表示させるには？

```
for i in range(1,11):  
    print( i , "回目のこんにちは" )
```

```
>python sample.py  
1回目のこんにちは  
2回目のこんにちは  
3回目のこんにちは  
4回目のこんにちは  
5回目のこんにちは  
6回目のこんにちは  
7回目のこんにちは  
8回目のこんにちは  
9回目のこんにちは  
10回目のこんにちは
```

変数iには1から代入されていく

iは1,2,...,9,10

変数iには10まで代入されていく

変数iは11にならないことに注意



回数の決まった繰り返し⑦

0から9までの合計を求めるプログラム

```
total = 0
```

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

```
    total += i
```

```
print( "合計は" , total )
```

変数iは0から9まで1ずつ加算

```
>python sample.py  
合計は45
```



回数の決まった繰り返し⑧

1から10までの合計を求めるには？

```
total = 0
```

```
for i in range(1,11):
```

```
    total += i
```

← 変数iは1から10まで1ずつ加算

```
print( "合計は" , total )
```

```
>python sample.py  
合計は55
```

回数の決まった繰り返し⑨

```
for i in range(10):  
    print( i , "番目" )
```

0 番目
1 番目
2 番目
3 番目
4 番目
5 番目
6 番目
7 番目
8 番目
9 番目

i=0

i=1

i=9

```
for i in range(10):  
    print( "*" * i )
```

*
**

i=0

i=1

i=9

0個表示→何も表示しない

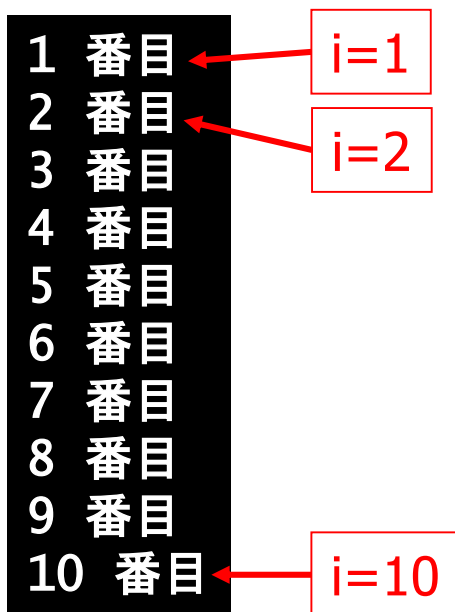
1個表示

iは0,1,2,...,9

9個表示

回数の決まった繰り返し⑩

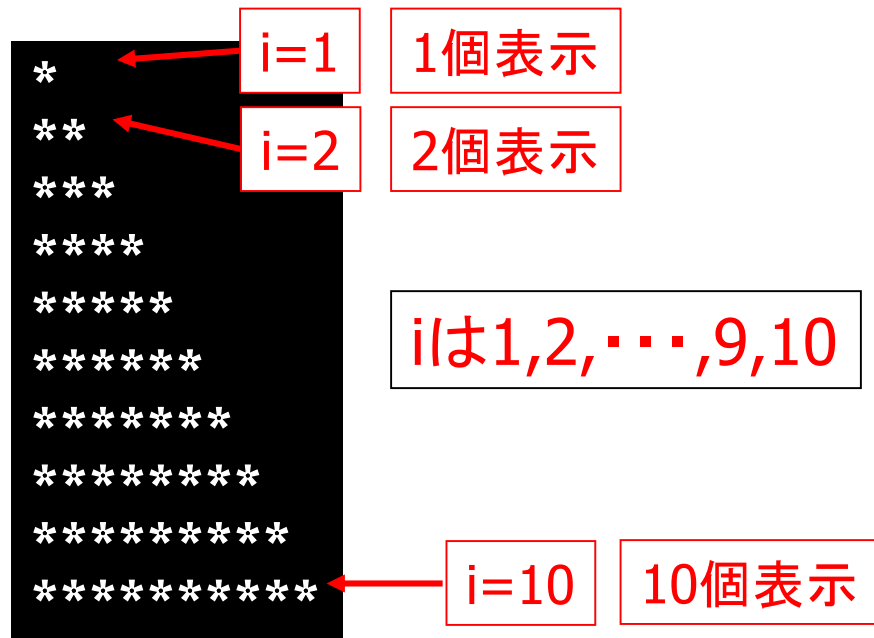
```
for i in range(1,11):  
    print( i , "番目" )
```



1 番目
2 番目
3 番目
4 番目
5 番目
6 番目
7 番目
8 番目
9 番目
10 番目

$i=1$
 $i=2$
 $i=10$

```
for i in range(1,11):  
    print( "*" * i )
```



*
**

$i=1$ 1個表示
 $i=2$ 2個表示
 $i=10$ 10個表示

i は1,2,...,9,10

回数の決まった繰り返し⑪

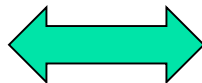
- なぜ, このような出力になるのでしょうか

```
for i in range(5):  
    print( " "*(10-i) , "*" * i , sep="" )
```

空白

空白を入れず表示

出力結果



i	10-i	空白の数	*の数
0	10	10	0
1	9	9	1
2	8	8	2
3	7	7	3
4	6	6	4



回数の決まった繰り返し⑫

- どのような出力になるでしょうか

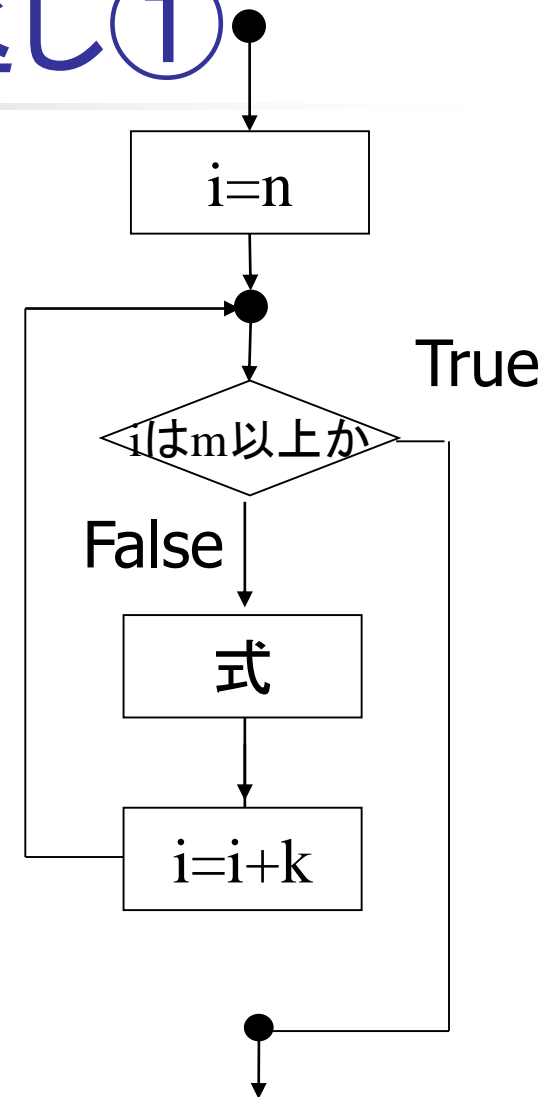
```
for i in range(10):  
    print( " "*(10-i) , "*" , sep="" )
```

```
for i in range(10):  
    print( " "*(10-i), "*", " "*i*2 , "*" , sep="")
```

範囲の決まった繰り返し①

```
for i in range(n,m,k):  
    式
```

変数*i*には*n*から*k*ずつ加算されながら
、*m*を超えない値まで代入





範囲の決まった繰り返し②

```
for i in range(1,10,2):  
    print( i )
```

```
1  
3  
5  
7  
9
```

```
for i in range(2,10,2):  
    print( i )
```

```
2  
4  
6  
8
```

10は表示されないことに注意



範囲の決まった繰り返し②'

```
for i in range(-5,5,2)  
    print( i )
```

```
-5  
-3  
-1  
1  
3
```

5は表示されないことに注意

```
for i in range(5,-5,-2)  
    print( i )
```

```
5  
3  
1  
-1  
-3
```

-5は表示されないことに注意



範囲の決まった繰り返し②"

```
for i in range(5,-5,2)  
    print( i )
```

何も印字されない

```
for i in range(1,10,-2)  
    print( i )
```

何も印字されない

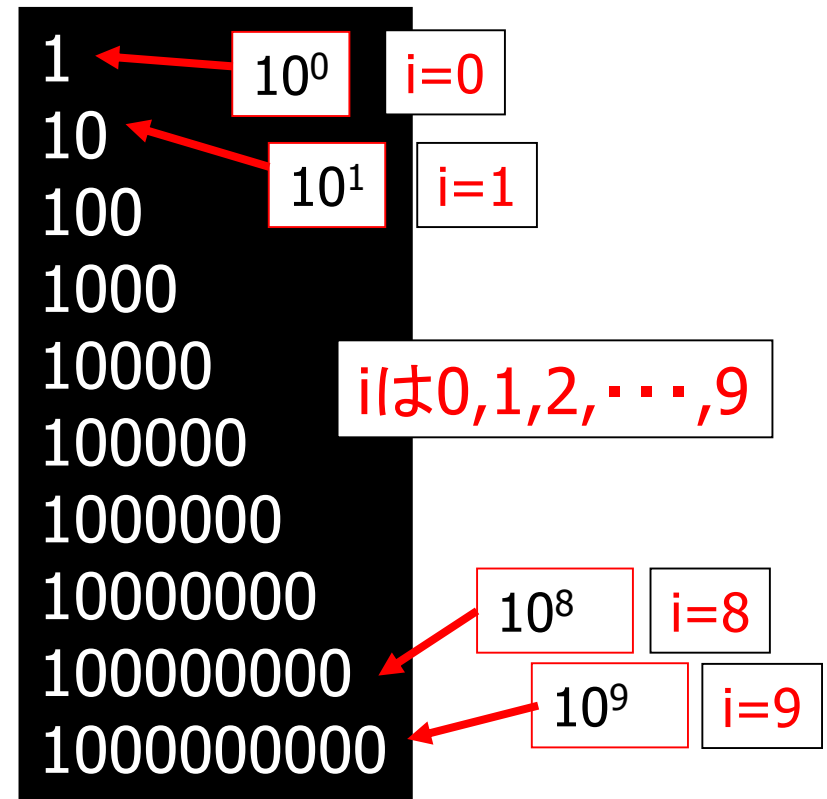
範囲の決まった繰り返し③

10のべき乗を表示するプログラム

```
for i in range(10):  
    print( 10 ** i )
```

```
for i in range(0,10):  
    print( 10 ** i )
```

```
for i in range(0,10,1):  
    print( 10 ** i )
```



範囲の決まった繰り返し④

10のべき乗を表示するプログラム

```
for i in range(0,11,2):  
    print( 10 ** i )
```

1 10^0 $i=0$
100 10^2 $i=2$
10000
1000000
100000000
10000000000 10^{10} $i=10$

i は0,2,4,6,8,10

```
for i in range(1,10,2):  
    print( 10 ** i )
```

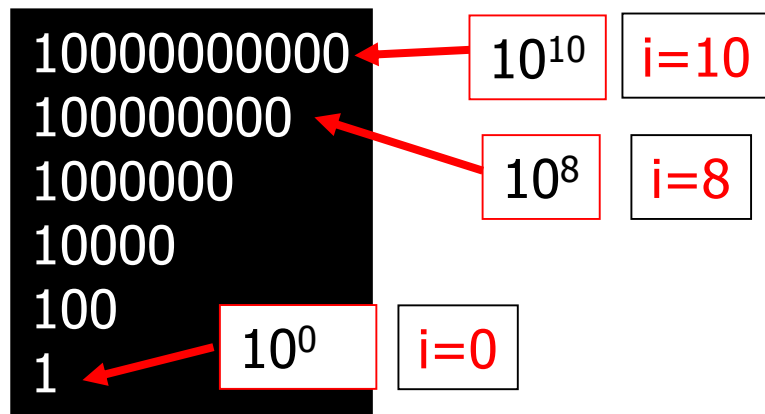
10 10^1 $i=1$
1000 10^3 $i=3$
100000
10000000
1000000000 10^9 $i=9$

i は1,3,5,7,9

範囲の決まった繰り返し④'

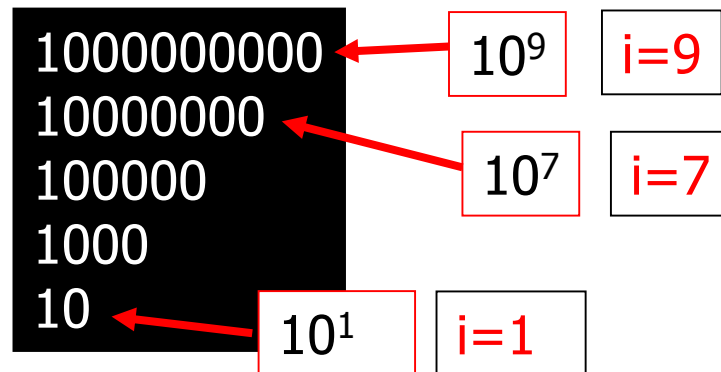
10のべき乗を表示するプログラム

```
for i in range(10,-1,-2):  
    print( 10 ** i )
```



iは10,8,6,4,2,0

```
for i in range(9,0,-2):  
    print( 10 ** i )
```



iは9,7,5,3,1

範囲が決まっている繰り返し⑤

100以下の整数の合計値
を求めるプログラム

```
total = 0
for i in range(1,101):
    total += i
print( "合計値" , total )
```

100以下の偶数の合計値
を求めるプログラム

```
total = 0
for i in range(2,101,2):
    total += i
print( "合計値" , total )
```

100としないこと

合計値 5050

iは1,2,...,99,100

合計値 2550

iは2,4,6,...,98,100



刻み幅に小数値を使用したい場合①

0から1まで0.1刻みで二乗の計算を行なう

```
for i in range(0.0,1.1,0.1):  
    print( i , " の二乗 " , i * i )
```

Traceback (most recent call last):

File "C:¥Users¥shino¥Desktop¥sample.py", line 1, in <module>

for i in range(0.0,1.0,0.1):

TypeError: 'float' object cannot be interpreted as an integer

刻み幅は整数でなければならない



刻み幅に小数値を使用したい場合①

0から1まで0.1刻みで二乗の計算を行なう

$i=0,1,2,\dots,10$

```
for i in range(0,11,1):  
    print( i / 10.0 , " の二乗 " , (i / 10.0)**2 )
```

iを10.0で除算

iを10.0で除算



刻み幅に小数値を使用したい場合①'

0から1まで0.1刻みで二乗の計算を行なう

```
for i in range(0,11,1):  
    print( i / 10.0 , " の二乗 " , (i / 10.0)**2 )
```

0.1刻みにしたい場合

$i=0,1,2,\dots,10$ として, この値を
10で割る

```
0.0 の二乗 0.0  
0.1 の二乗 0.010000000000000002  
0.2 の二乗 0.040000000000000001  
0.3 の二乗 0.09  
0.4 の二乗 0.160000000000000003  
0.5 の二乗 0.25  
0.6 の二乗 0.36  
0.7 の二乗 0.48999999999999994  
0.8 の二乗 0.64000000000000001  
0.9 の二乗 0.81  
1.0 の二乗 1.0
```



刻み幅に小数値を使用したい場合②

0から1まで, 刻み幅0.1ごと

```
for i in range(0,11,1):  
    print( i / 10.0 , " の二乗 " , (i / 10.0)**2 )
```

0から1まで, 刻み幅0.01ごと

```
for i in range(0,101,1):  
    print( i / 100.0 , " の二乗 " , (i / 100.0)**2 )
```



刻み幅に小数値を使用したい場合③

0から10まで, 刻み幅0.5ごと

```
for i in range(0,21,1):  
    print( i / 2.0 , " の二乗 " , (i / 2.0)**2 )
```

0から10まで, 刻み幅0.25ごと

```
for i in range(0,41,1):  
    print( i / 4.0 , " の二乗 " , (i / 4.0)**2 )
```



無限の繰り返し

```
while True:  
    break
```




無限の繰り返し①

while True:
式

式が永久に実行される
停止するために **break** を
用いる

while True:
if 論理式:
 break
次の式

論理式を満たした場合の
み停止する(whileブロック
の次の式を実行する)



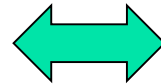
無限の繰り返し②

while True:

if 論理式:

break

次の式



while True:

if 論理式: break

次の式

```
while True:
    print( "こんにちは" )
```

無限に「こんにちは」と表示される

停止するにはCtrlキーを押しながらc

無限に「こんにちは」と表示される

停止するにはCtrlキーを押しながらc





無限の繰り返し

- while True: の場合, 式が無限に繰り返される
- 停止させるためには, 条件文とbreak文で設定しなければならない

無限の繰り返し③

10回で「こんにちは」の表示をやめるには？

```
i = 0
```

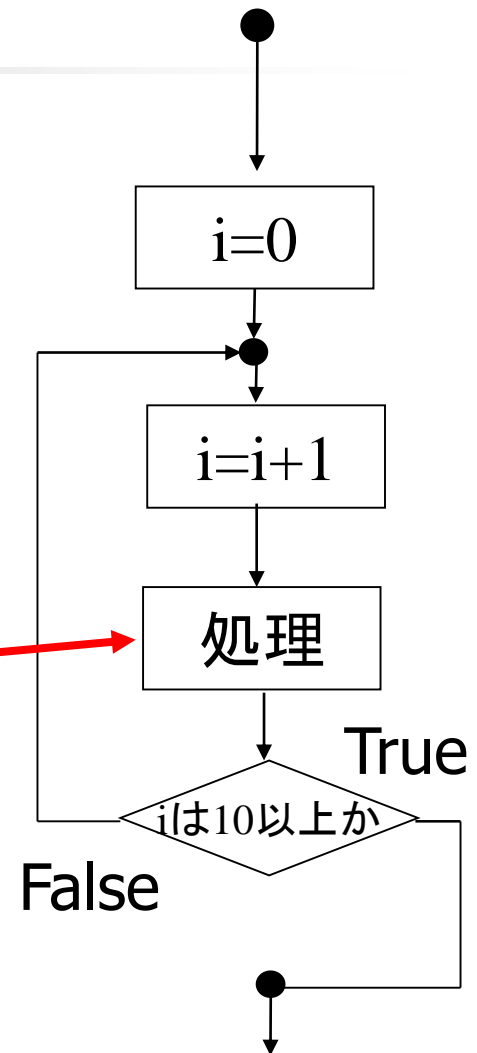
```
while True:
```

```
    i = i+1
```

```
    print( i , "回目のこんにちは" )
```

```
    if i >= 10:
```

```
        break
```





無限の繰り返し③

10回で「こんにちは」の表示をやめるには？

```
i = 0
```

```
while True:
```

```
    i = i+1
```

```
    print( i , "回目のこんにちは" )
```

```
    if i >= 10:
```

```
        break
```

i が10以上になったら停止する

```
> python sample.py
```

```
1回目のこんにちは
```

i=1

```
2回目のこんにちは
```

i=2

```
3回目のこんにちは
```

```
4回目のこんにちは
```

```
5回目のこんにちは
```

```
6回目のこんにちは
```

```
7回目のこんにちは
```

```
8回目のこんにちは
```

```
9回目のこんにちは
```

i=9

```
10回目のこんにちは
```

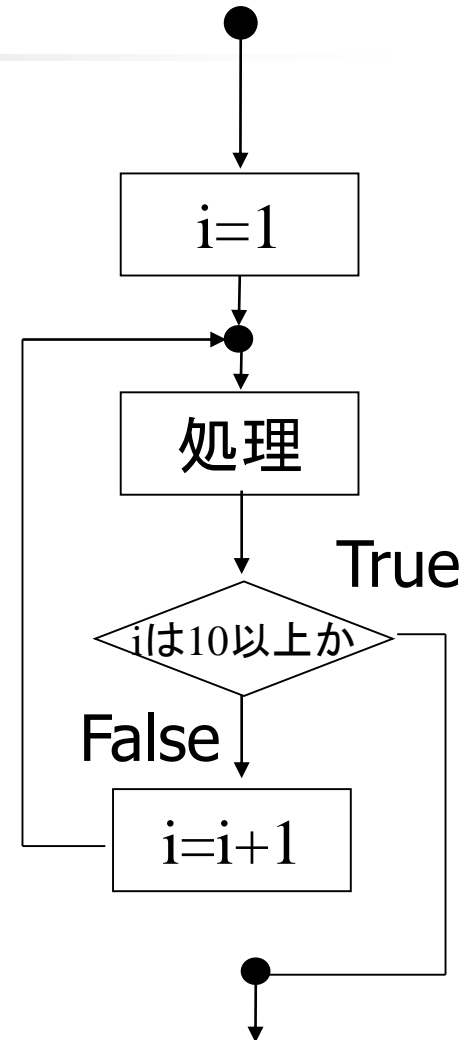
i=10

無限の繰り返し③'

10回で「こんにちは」の表示をやめるには？

```
i = 1
while True:
    print( i , "回目のこんにちは" )
    if i >= 10:
        break
    i = i+1
```

前のページと同じ動作をします





間違えやすいミス①

10回で「こんにちは」の表示をやめるには？

```
while True:
    i = i+1
    print( i , "回目のこんにちは" )
    if i >= 10:
        break
```

Traceback (most recent call last):

File "C:¥Users¥shino¥Desktop¥sample.py", line 2, in <module>

i = i+1

NameError: name 'i' is not defined

変数 i が定義されていない



間違えやすいミス②

10回で「こんにちは」の表示をやめるには？

```
i = 0
while True:
    print( i , "回目のこんにちは" )
    if i >= 10:
        break
```

変数 *i* を1ずつ増やしていない
→ *i*は0のままのためプログラム
は終了しない(無限ループ)

0回目のこんにちは
0回目のこんにちは
0回目のこんにちは
0回目のこんにちは
0回目のこんにちは
0回目のこんにちは
0回目のこんにちは
0回目のこんにちは

間違いやすいミス③

10回で「こんにちは」の表示をやめるには？

```
i = 0
while True:
    print(i, "回目のこんにちは")
    if i >= 10:
        break
    i += 1
```

i の初期値を0として開始

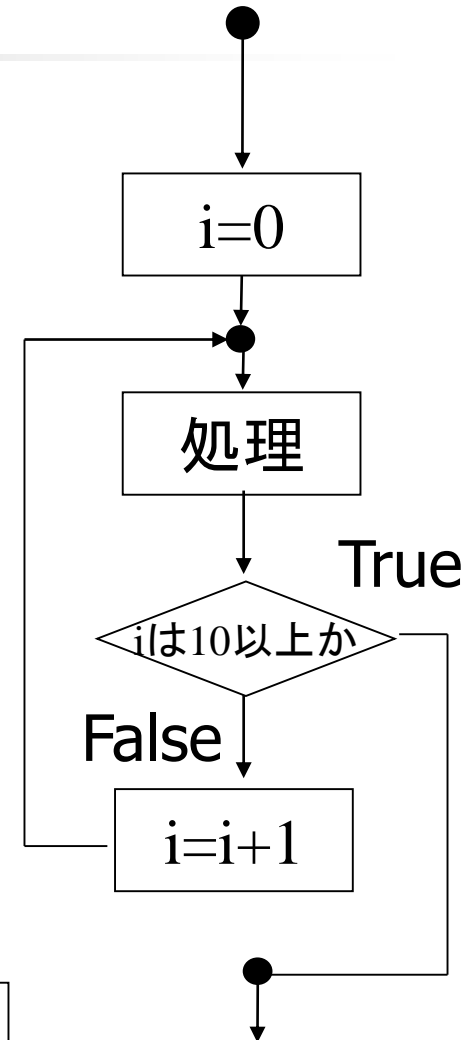
0回目のこんにちは
1回目のこんにちは
2回目のこんにちは
3回目のこんにちは
4回目のこんにちは
5回目のこんにちは
6回目のこんにちは
7回目のこんにちは
8回目のこんにちは
9回目のこんにちは
10回目のこんにちは

無限の繰り返し④

10のべき乗を表示するプログラム

```
i = 0
while True:
    print( 10 ** i )
    if i >= 10:
        break
    i = i + 1
```

```
1 i=0
10 i=1
100 i=2
1000
10000
100000
1000000
10000000
100000000 i=9
1000000000 i=10
```

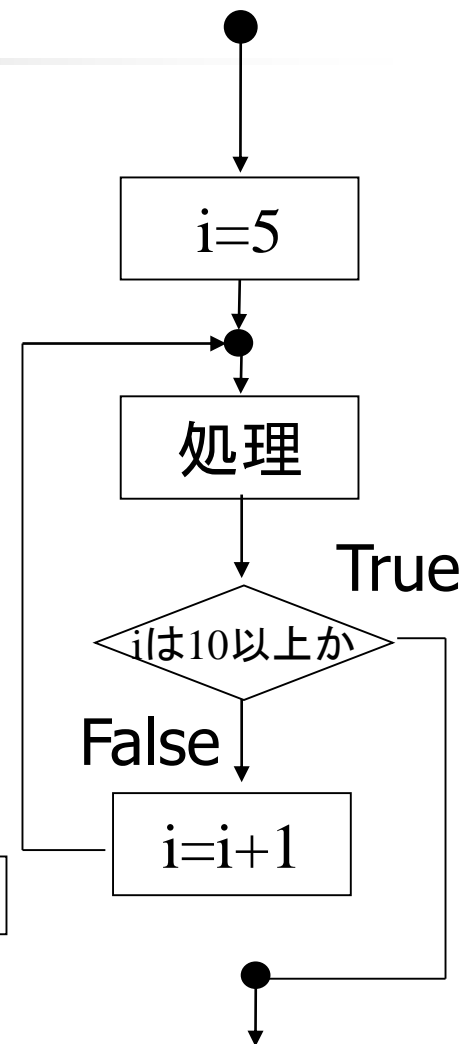


無限の繰り返し④'

10のべき乗を表示するプログラム

```
i = 5
while True:
    print( 10 ** i )
    if i >= 10:
        break
    i = i + 1
```

100000	i=5
1000000	i=6
10000000	
100000000	
1000000000	i=9
10000000000	i=10

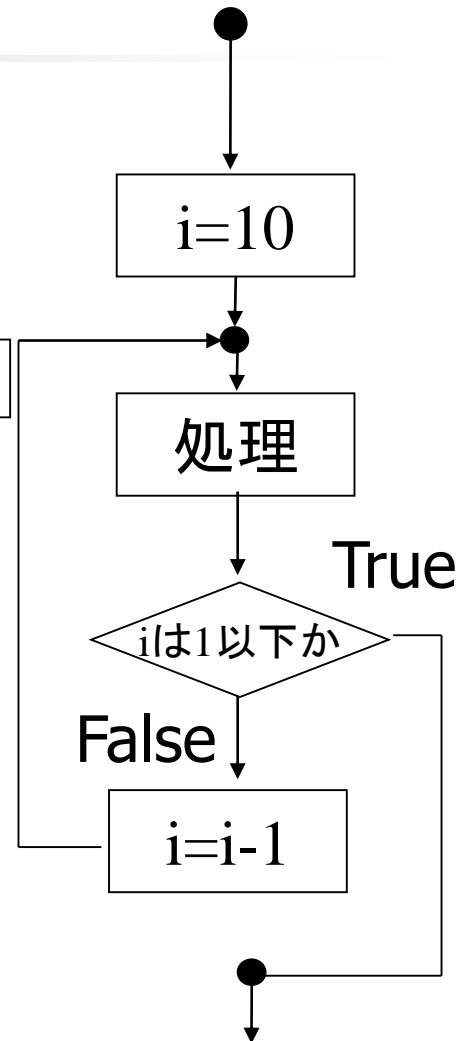


無限の繰り返し④"

10のべき乗を表示するプログラム

```
i = 10
while True:
    print( 10 ** i )
    if i <= 1:
        break
    i = i-1
```

```
10000000000 i=10
1000000000 i=9
100000000
10000000
1000000
100000
10000
1000
100 i=2
10 i=1
```



無限の繰り返し⑤

100以下の偶数を表示するプログラム

```
i = 1
while True:
    if i % 2 == 0:
        print( i )
```

```
    if i >= 100:
        break
    i = i+1
```

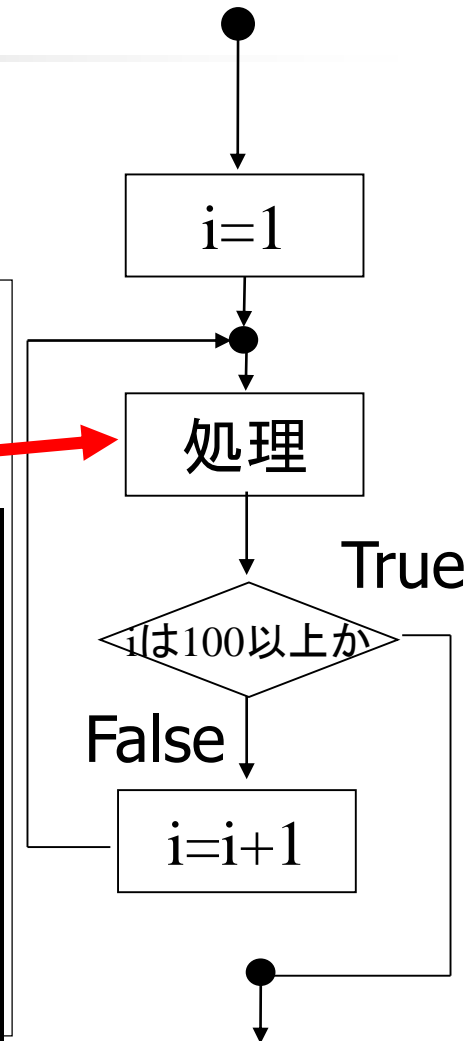
2
4
6
8
10
12

98
100

i=2

i=4

i=100



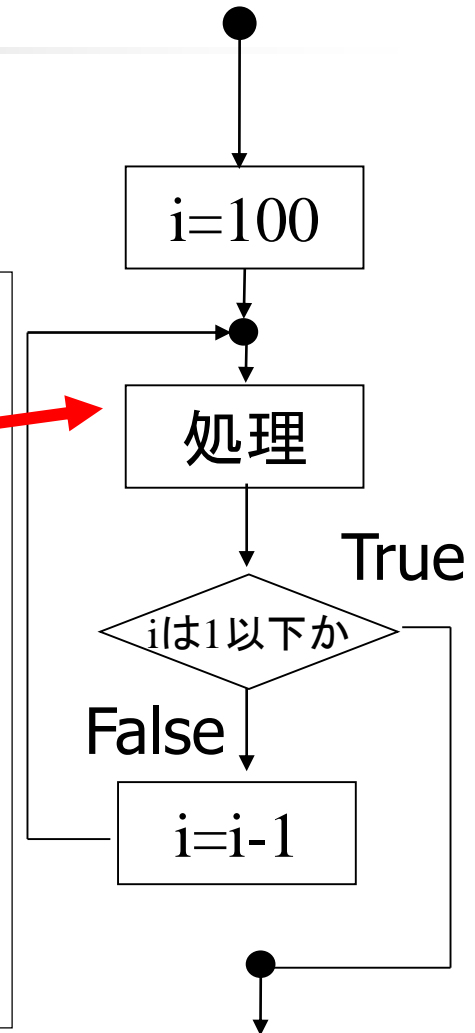
無限の繰り返し⑤'

100以下の偶数を表示するプログラム

```
i = 100
while True:
    if i % 2 == 0:
        print( i )
```

```
    if i <= 1:
        break
    i = i-1
```

100	i=100
98	i=98
96	
94	
92	
4	i=4
2	i=2



無限の繰り返し⑤"

100以下の偶数を表示するプログラム

```
i = 2
```

```
while True:
```

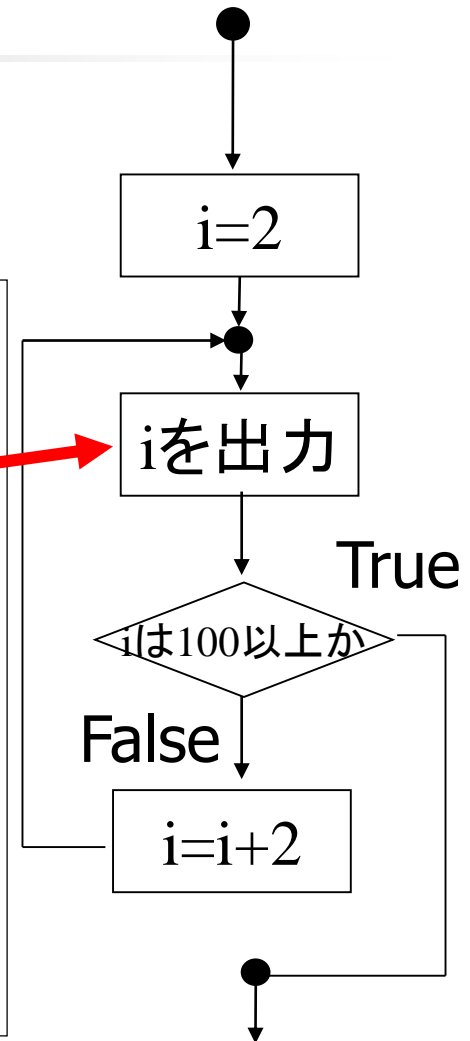
```
    print( i )
```

```
    if i >= 100:
```

```
        break
```

```
    i = i+2
```

i に2ずつ加算



無限の繰り返し⑤'''

100以下の偶数を表示するプログラム

```
i = 1
```

```
while True:
```

```
    print( 2*i )
```

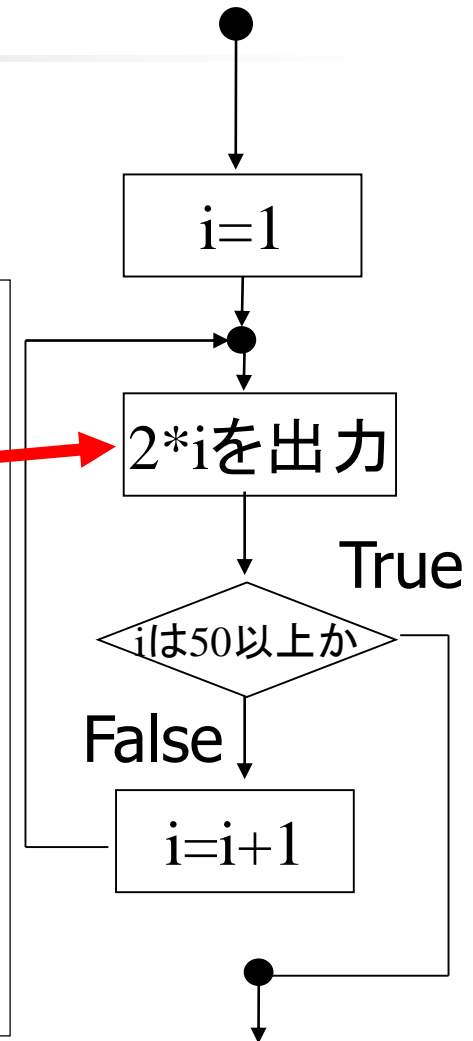
```
    if i >= 50:
```

```
        break
```

```
    i = i+1
```

iの2倍の値を出力

iに1ずつ加算



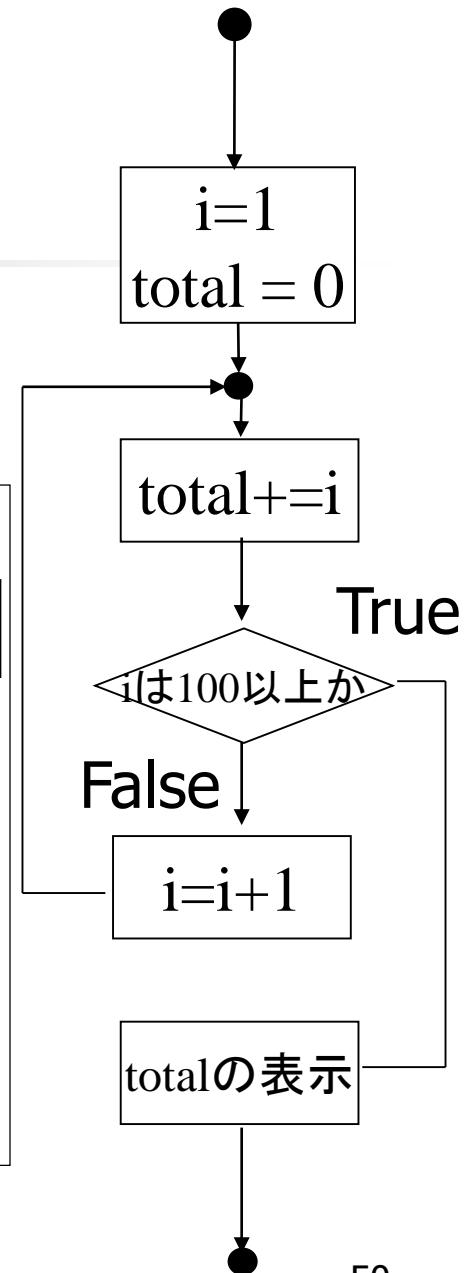
無限の繰り返し⑥

100以下の整数の和を求めるプログラム

```
i = 1
total = 0
while True:
    total += i
    if i >= 100:
        break
    i = i + 1
print( "合計は" , total )
```

iが100以上の場合、停止する

```
>python sample.py
合計は5050
```

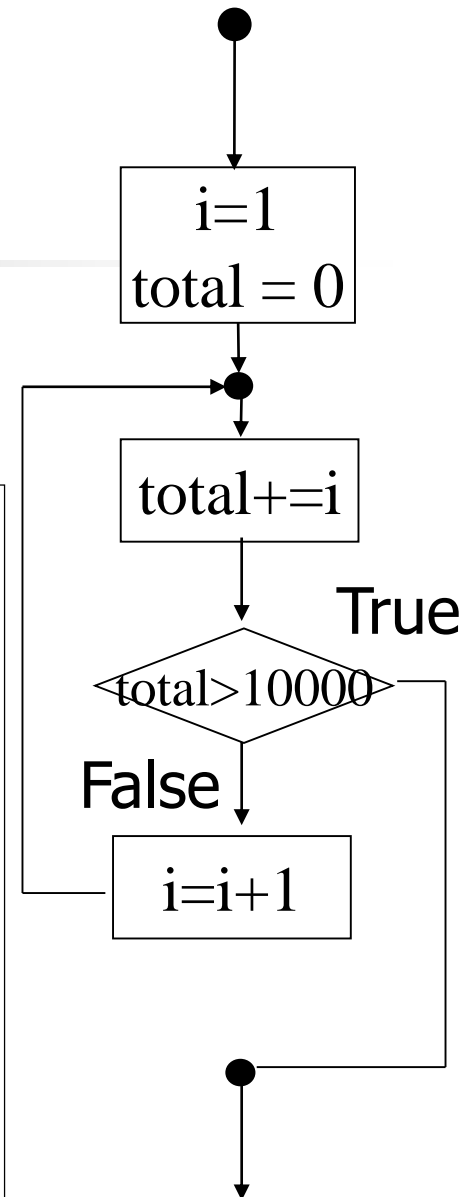


無限の繰り返し⑥'

どういうプログラムでしょうか

```
i = 1
total = 0
while True:
    total += i
    if total > 10000:
        break
    i = i + 1
print( i , total )
```

totalの値が10000を
越えたら停止

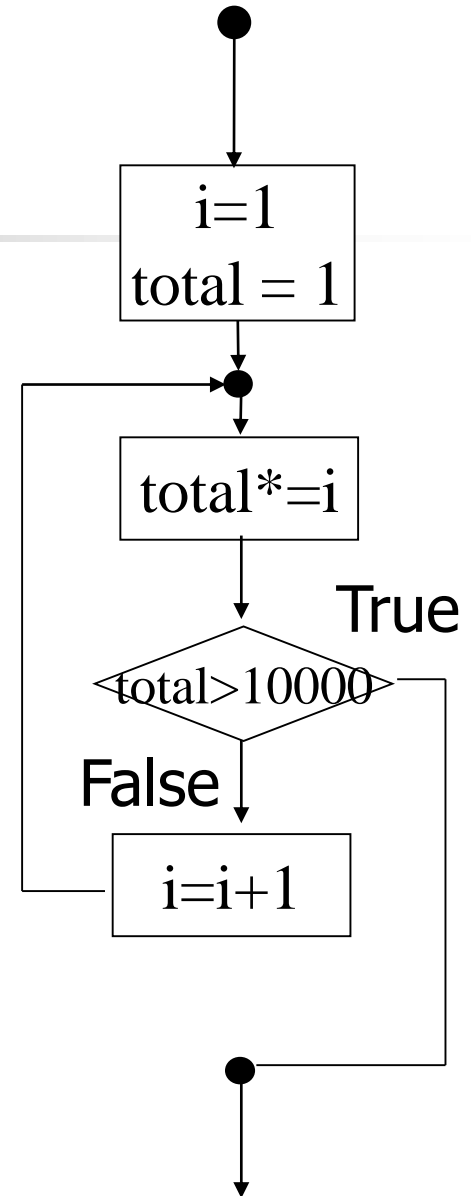


無限の繰り返し⑥"

どういうプログラムでしょうか

```
i = 1
total = 1
while True:
    total *= i
    if total > 10000:
        break
    i = i + 1
print( i , total )
```

totalの値が10000を
越えたら停止



無限の繰り返し⑥'''

どういうプログラムでしょうか

```
i = 100
```

```
count = 0
```

```
while True:
```

```
    i = int( i / 2 )
```

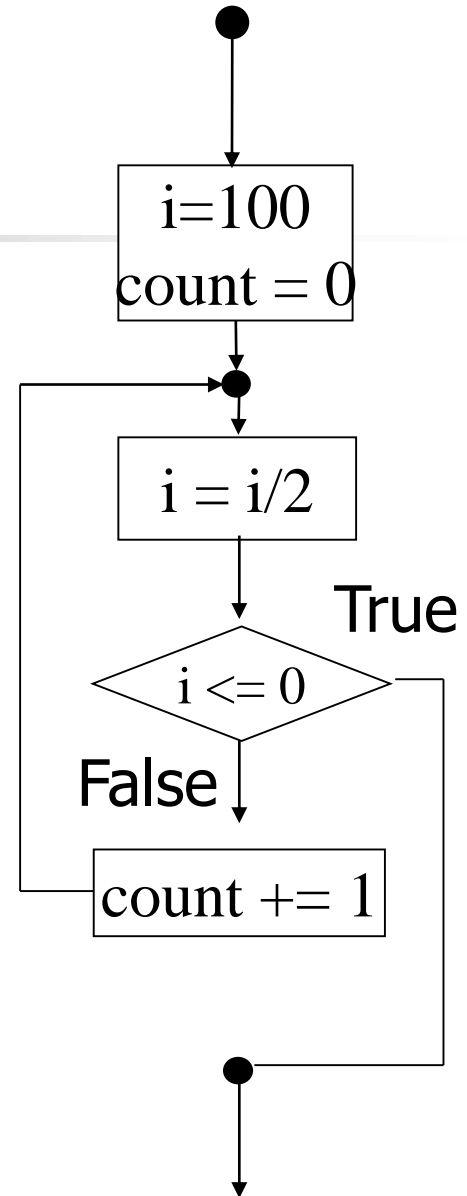
```
    if i <= 0:
```

```
        break
```

```
    count = count + 1
```

```
    print( count )
```

iの値が0以下の場合
、停止





無限の繰り返しのまとめ①

- **while True:**

式

- 上記の「式」を無限に繰り返す。無限個のコピーを作ると考えてもよい。ただし、いきなり作るのではなく、必要があったら作るのですが。
- しかし、いずれにせよ、無限に作られるのは困る。
- 途中で止めなければ意味がない。
- 途中で止める道具(これも文だが、まったく文らしくない)が **break** です。

```
i = 0
while True:
    print( "やっほ~ " )
    if i>=10:
        break
    puts( "          Yee-ha! " )
    i = i+1
```

無限の繰り返しのまとめ②

a から b まで c ずつ加算しながら
繰り返し処理を行なう

```
i = a
```

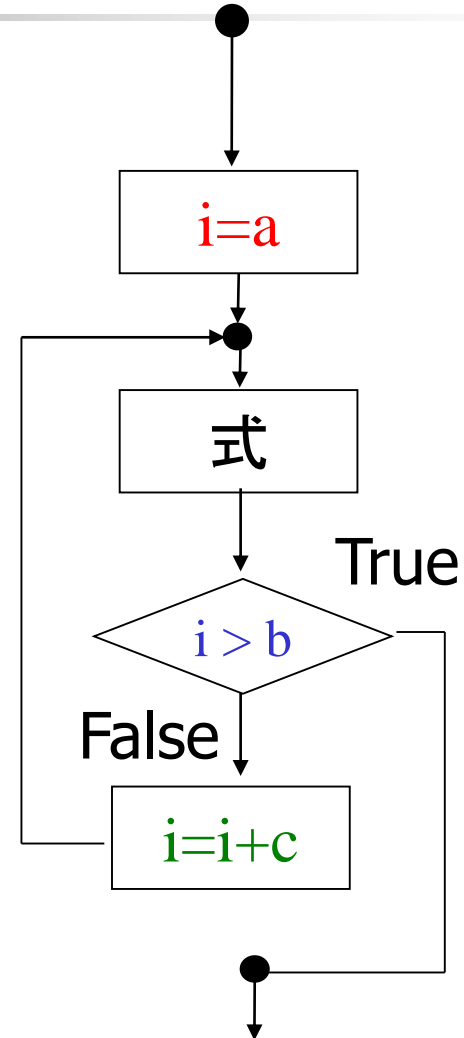
```
while True:
```

```
    式
```

```
    if i > b:
```

```
        break
```

```
    i += c
```

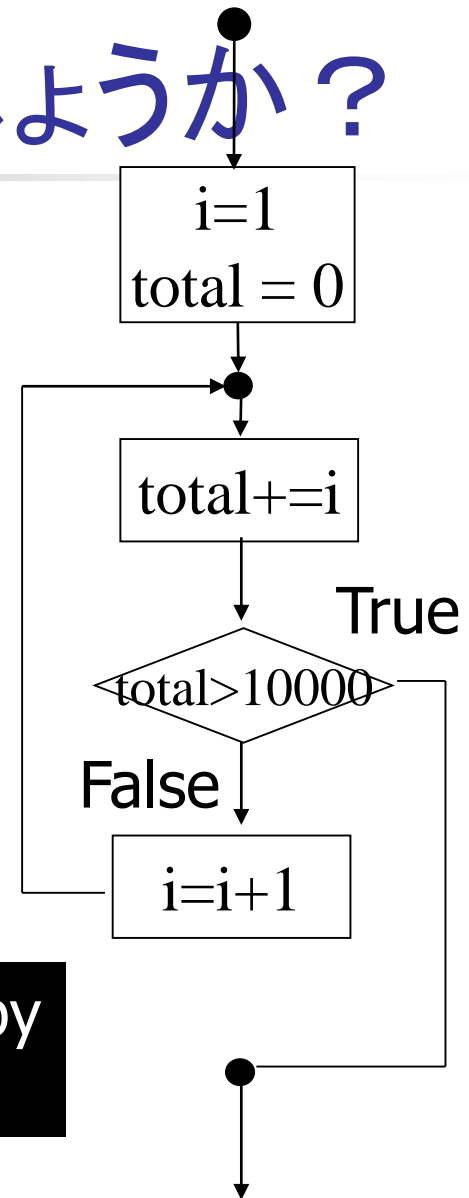


for文では書けるでしょうか？

1からnまでの合計が10,000以上で終了

```
i = 1
total = 0
while True:
    total += i
    if total > 10000:
        break
    i = i + 1
print( i , total )
```

```
> python sample.py
141 10011
```





for文では書けるでしょうか？

無理やり書くと...

```
total = 0
```

```
n = 0
```

```
for i in range(1, 10000):
```

```
    total += i
```

```
    if total > 10000:
```

```
        n = i
```


```
        break
```

```
print( n , total )
```

上限が不明なため、大きな値としておく



for文においてもbreakでループから抜け出すことが可能



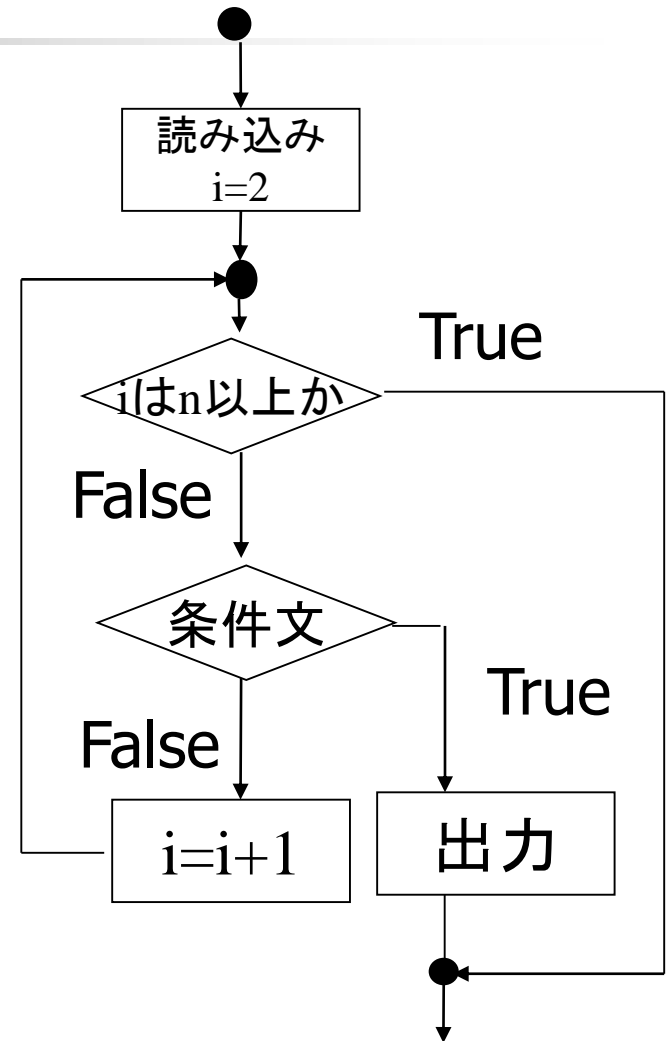
break文

何を調べているプログラムでしょう

```
n=int(input())  
for i in range(2,n):  
    if n % i == 0:  
        print( "〇〇ではありません" )
```

break

break でループを抜け出ることも可能





繰り返しの場合分け

- 同じ処理をある回数だけ行ないたい
 - for, while
 - ただし, for文の方が記述が容易
- ある条件が成立するまで, 同じ処理を繰り返したい
 - while
 - for文では書けない場合がある

while文の場合の注意

while文の場合

```
i=0.0
while True:
    print( i , " " , i**2 )
    if i == 1.0:
        break
    i += 0.1
```

実行結果

→ 終了しない
(Ctrl+Cで強制終了)

0.1ずつ加算

0.1を10回加算しても1.0とはならない(なぜでしょうか)

```
Command Prompt
722.5000000000957 522006.2500001383
722.6000000000957 522150.76000013837
722.7000000000958 522295.2900001384
722.8000000000958 522439.8400001385
722.9000000000958 522584.4100001385
723.0000000000958 522729.0000001386
723.1000000000959 522873.61000013864
723.2000000000959 523018.2400001387
723.3000000000959 523162.8900001387
723.4000000000959 523307.56000013876
723.500000000096 523452.2500001388
723.600000000096 523596.9600001389
723.700000000096 523741.69000013894
723.800000000096 523886.440000139
723.900000000096 524031.2100001391
```

while文の場合の注意

while文の場合

```
i=0.0
while True:
    print( i , " " , i**2 )
    if i >= 1.0:
        break
    i += 0.1
```

0.1ずつ加算

1を越えて出力

1.0になっていません

```
0.0 0.0
0.1 0.010000000000000002
0.2 0.040000000000000001
0.30000000000000004 0.09000000000000002
0.4 0.16000000000000003
0.5 0.25
0.6 0.36
0.7 0.48999999999999994
0.7999999999999999 0.6399999999999999
0.8999999999999999 0.8099999999999998
0.9999999999999999 0.9999999999999998
1.0999999999999999 1.2099999999999999
```



刻み幅に小数を使う場合の注意

```
i=0
while True:
    print( i/10.0 , " " , (i/10.0)**2 )
    if i >= 10:
        break
    i += 1
```

iを1ずつ加算する

1.0で停止

```
>python sample.py
0.0 0.0
0.1 0.010000000000000002
0.2 0.040000000000000001
0.3 0.09
0.4 0.160000000000000003
0.5 0.25
0.6 0.36
0.7 0.48999999999999994
0.8 0.64000000000000001
0.9 0.81
1.0 1.0
```

刻み幅に小数を使う場合の注意

```
i=0.0
while True:
    print( i , " " , i**2 )
    if i >= 1.0:
        break
    i += 0.1
```

0.1を10回加算しても1とはならない



```
i=0
while True:
    print( i/10.0 , " " , (i/10.0)**2 )
    if i >= 10:
        break
    i += 1
```

刻み幅は必ず整数とする



標準入力と繰り返し

標準入力(復習)

「5」を入力

```
>python sample.py
```

5

1回目のこんにちは

2回目のこんにちは

3回目のこんにちは

4回目のこんにちは

5回目のこんにちは

```
n = int( input() )
```

```
i = 0
```

```
while True:
```

```
    i = i+1
```

```
    print( i , "回目のこんにちは" )
```

```
    if i >= n:
```

```
        break
```

input により 文字列の"5"
int により整数の5に変換される



簡単に書くには...①

```
a = int(input())  
b = int(input())  
c = int(input())  
d = int(input())  
e = int(input())  
sum = a+b+c+d+e  
print( "合計は" , sum , "です" )
```

5個の整数を読み込み, 合計を出力

```
>python sample.py  
43  
34  
2  
1  
1  
合計は81です
```

簡単に書くには...②

```
sum = 0
i = 0
while True:
    x = int(input())
    sum += x
    i += 1
    if i >= 5:
        break
print( "合計は" , sum , "です" )
```

変数xに整数を入力

5回入力したら停止

```
>python sample.py
4
1
4
6
7
合計は22です
```

簡単に書くには...②'

for文でも書けます

```
sum = 0
for i in range(5):
    x = int(input())
    sum += x
print( "合計は" , sum , "です" )
```

変数xに整数を入力

```
>python sample.py
4
1
4
6
7
合計は22です
```

入力の繰り返し①

```
max = 0
```

```
i = 0
```

```
while True:
```

```
    x = int(input())
```

```
    if x > max:
```

```
        max = x
```

```
    i += 1
```

```
    if i >= 5:
```

```
        break
```

```
print( "最大は" , max , "です" )
```

変数xに整数を入力

5回入力したら停止

```
>python sample.py
```

```
23
```

```
12
```

```
68
```

```
45
```

```
23
```

```
最大は68です
```

入力の繰り返し①'

for文でも書けます

```
max = 0
for i in range(5):
    x = int(input())
    if x > max:
        max = x
print( "最大は" , max , "です" )
```

変数xに整数を入力

```
>python sample.py
23
12
68
45
23
最大は68です
```

入力の繰り返し②

```
sum = 0  
while True:
```

```
    x = int(input())
```

```
    sum += x
```

```
    if sum > 100:
```

```
        break
```

```
print( "合計は" , sum , "です" )
```

変数xに整数を入力

sumの値が100を越えたら停止

```
>python sample.py  
34  
23  
17  
56  
合計は130です
```



入力回数が分からない場合

- 前々頁のプログラムは入力が5回
- 前頁のプログラムは条件文によって停止
- 入力回数が分からない場合はどうすればよいか
 - 特定の文字(Enterなど)を入力した場合のみ入力を終了させるようにする

繰り返し入力できるようにするためには①

```
while True:
    line = input( "何か文字を入れて下さい(Enterで終了)¥n" )
    if line == "":
        break
    print( line )
```

Enterで終了

```
>python sample.py
何か文字を入れて下さい(Enterで終了)
24
24
何か文字を入れて下さい(Enterで終了)
abcd
abcd
何か文字を入れて下さい(Enterで終了)
sdds
sdds
何か文字を入れて下さい(Enterで終了)
```



キーボードからの入力

キーボードでEnterキーを入力した場合

```
while True:
    line = input( "何か文字を入れて下さい(Enterで終了)¥n" )

    if line == "":
        break
    print( line )
```

lineにはEnter(改行)は削除され空白文字が入る

lineには空白文字が入っているため break が実行され停止する

繰り返し入力できるようにするためには②

```
while True:
    line = input( ( "何か文字を入れて下さい(stopで終了)¥n" ) )
    if line == "stop":
        break
    print( line )
```

stopで終了

```
>python sample.py
何か文字を入れて下さい(stopで終了)
32
32
何か文字を入れて下さい(stopで終了)
stop
```

繰り返し入力できるようにするためには③

```
sum = 0
while True:
    line =input( "整数を入れて下さい(Enterで終了)¥n" )
    if line == "":
        break
    print( line )
    sum += int( line )
    print( "合計値は" , sum )
```

空白かどうか→空白の場合は停止

文字型の変数line
→整数に型変換

```
sum = 0
while True:
    line =input( "整数を入れて下さい(Enterで終了)¥n" )
    if line == "":
        break
    print( line )
    sum += int( line )
print( "合計値は" , sum )
```

```
>python sample.py
整数を入れて下さい(Enterで終了します)
32
32
整数を入れて下さい(Enterで終了します)
23
23
整数を入れて下さい(Enterで終了します)
11
11
整数を入れて下さい(Enterで終了します)

合計値は66
```

繰り返し入力できるようにするため には④

while True:

line = input("Enter your score: ")

if line=="":

break

score = float(line)

if score >= 70:

if score >= 80:

grade = "A"

else:

grade = "B"

else:

if score >= 60:

grade = "C"

else:

grade = "D"

print("Your score " , score , "corresponds to" , grade)

改行キー(Enter)のみ入力
された場合、停止

> python sample.py

Enter your score: 90

Your score 90.0 corresponds to A

Enter your score: 40

Your score 40.0 corresponds to D

Enter your score:



練習問題

練習①～④

(簡単な人は練習⑤も試してみてください)

実行画面通りに表示させてなくて結構です

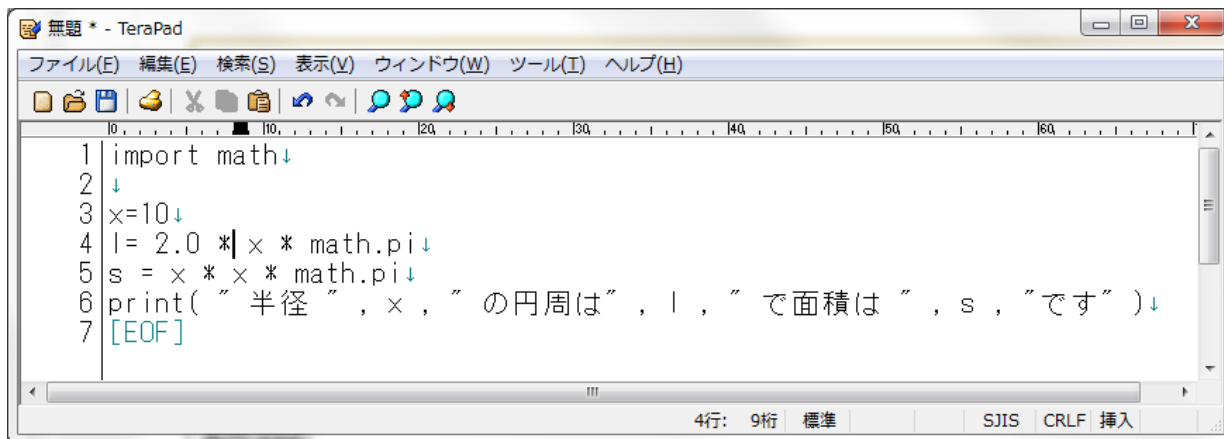


練習問題を提出する上での注意

- プログラムについて
 - エディターをキャプチャーした図ではなく、テキストのままWordに貼り付けて下さい
- 実行結果について
 - 対話型シェルで一文ごと実行するのではなく、pythonコマンドで実行して下さい

練習問題を提出する上での注意

(×)



```
1 import math↓
2 ↓
3 x=10↓
4 l= 2.0 *| x * math.pi↓
5 s = x * x * math.pi↓
6 print( " 半径 " , x , " の円周は" , l , " で面積は " , s , " です" )↓
7 [EOF]
```

The screenshot shows a TeraPad window with a menu bar (File, Edit, Search, View, Window, Tools, Help) and a toolbar. The code is written in a monospaced font. Line numbers 1 through 7 are on the left. Blue arrows point to the end of lines 1, 2, 3, 4, 5, and 6. A vertical bar is placed between the space and 'x' in line 4. The status bar at the bottom indicates '4行: 9桁 標準' and 'SJIS CRLF 挿入'.

(○)

```
import math
x=10
l= 2.0 * x * math.pi
s = x * x * math.pi
print( " 半径 " , x , " の円周は" , l , " で面積は " , s , " です" )
```

練習問題を提出する上での注意

(×)

```
コマンド プロンプト - python
C:\Users\shino>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> x=10
>>> l= 2.0 * x * math.pi
>>> s = x * x * math.pi
>>> print( " 半径 ", x, " の円周は ", l, " で面積は ", s, " です" )
 半径  10   の円周は 62.83185307179586   で面積は  314.1592653589793   です
>>>
```

(○)

```
C:\Windows\system32\cmd.exe
C:\Users\shino\Desktop>python 4-2.py
 半径  10   の円周は 62.83185307179586   で面積は  314.1592653589793   です
C:\Users\shino\Desktop>
```



練習①

- 1から100までの整数のうち3の倍数を表示するプログラムをfor, while True:を用いて書きなさい.
- 実行画面は一部の結果のみ貼り付けて下さい.

```
> python 6-1.py  
3  
6  
9  
12  
15  
18
```



練習②

- 1から100までの整数の二乗和を印字するプログラムをwhile True:を用いて書きなさい

二乗和

$$\sum_{i=1}^{100} i^2$$

```
> python 6-2.py  
自乗和は338350
```



練習③

- 正の整数をキーボードから読み込み，積を印字するプログラムを書きなさい．負の整数を読み込むまで，入力は続くものとします．

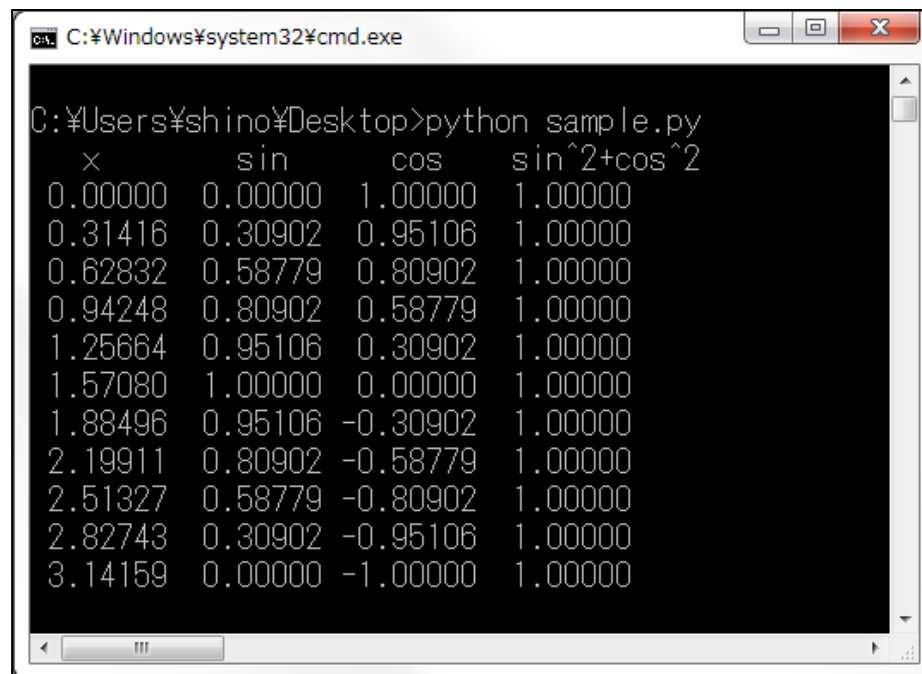
```
>python 6-3.py
2
4
5
7
3
-1
積は840です
```

負の整数を入力したら
入力は停止



練習④

- 0から π まで, 0.1π 刻みでsin, cos, $\sin^2 + \cos^2$ を印字するプログラムを書きなさい



```
C:\Windows\system32\cmd.exe

C:\Users\shino\Desktop>python sample.py

x          sin      cos      sin^2+cos^2
0.00000    0.00000    1.00000    1.00000
0.31416    0.30902    0.95106    1.00000
0.62832    0.58779    0.80902    1.00000
0.94248    0.80902    0.58779    1.00000
1.25664    0.95106    0.30902    1.00000
1.57080    1.00000    0.00000    1.00000
1.88496    0.95106   -0.30902    1.00000
2.19911    0.80902   -0.58779    1.00000
2.51327    0.58779   -0.80902    1.00000
2.82743    0.30902   -0.95106    1.00000
3.14159    0.00000   -1.00000    1.00000
```



練習⑤

- 57ページのスライドのプログラムは素数でないことを判定するプログラムです

```
n=int(input())
for i in range(2,n):
    if n % i == 0:
        print( "素数ではありません" )
        break
```



練習⑤

- このプログラムを改良し，整数 n が素数かどうかを判定するプログラムに書き直して下さい

```
>python 6-5.py  
9999973  
9999973は素数です
```

```
>python 6-5.py  
123456789  
123456789は素数ではありません
```




練習問題

- 練習問題①から④を行ないなさい
- (簡単な人は練習⑤も試してみてください)
- プログラムと実行結果をワープロに貼り付けて, keio.jp から提出して下さい