

●if でよく出てくる比較演算子

| python | | DNCL |
|--------|-------|------|
| == | 等しい | == |
| != | 等しくない | != |
| > | 大なり | > |
| >= | 以上 | >= |
| < | 小なり | < |
| <= | 以下 | <= |

練習 1 input を使い数字を入力できるようにしよう。a に入った数字がコンピューターは文字か数字か判断できません。そこで a に入ったものが数字だと判断させるようにします。int を使います

```
a=input("点数は?")
ten=int(a)
print(ten)
```

変数「ten」の中に
int(a)を入れます
☆int は中のものを数値型
に変換してくれます

練習 2 条件によって表示が変わるようにしよう。
今回は入れた点数が 80 点以上なら「合格」、
そうでなければ「不合格」となるようにします。

```
a=input("数字は?")
ten=int(a)
print(ten)
```

```
if ten>=80:
    print("合格")

else:
    print("不合格")
```

条件で分けるときは if~else 文
使います。

```
if(条件式):
    処理 A

else:
    処理 B
```

☆if、else の後に print 文を入れる場合は字下げ(tab キー)が必要です。
字下げで if 文の中にあるという意味になります。

字下げなし

```
if ten>20:
print("合格")
```

字下げあり

```
if ten>=20:
    print("不合格")
```

練習 3 さらに処理を3つに分けてみよう。
80 点以上は「合格」、70 点以上は「もう少し」、
そうでなければ「不合格」となるようにします。

```
a=input("数字は？")
ten=int(a)
print(ten)
```

```
if ten>=80:
    print("合格")

elif: ten>=70:
    print("合格")

else:
    print("不合格")
```

複数の条件に分けるときは
elif を使います

```
if(条件式):
    処理 A

elif(条件式):
    処理 B

else:
    処理 C
```

基礎課題 1 入力された数字が偶数かどうか判定できるプログラムを作ろう

ヒント①偶数になる条件は 2 で割って余りが 0 のとき
ヒント②それ以外は奇数
ヒント③print の中身を変えてみよう。

重要 Python と DNCL の比較

Python

```
a=input("数字は？")
b=int(a)
print(b)

if b>=80:
    print("合格")

else:
    print("不合格")
```

DNCL

- (1) a = 【外部からの入力】
- (2) もし b が 80 以上ならば：
- (3) | 表示する ("合格")
- (4) そうでなければ：
- (5) | 表示する ("不合格")

☆目標☆

Python で繰り返しを覚える

練習 1 「1、2、3」と順番に表示されるプログラム

```
for i in range (1,4):
    print(i)
```

- ・繰り返しは変数 i を使います
- ・range(開始、終了)という意味
- ・(1,4) は 1 から 4 未満の数字まで繰り返すという意味→ 1 から 3 回まで

●繰り返しはトレースするのが大事

| 繰り返し回数 | i の値 |
|--------|------|
| 1 回目 | |
| 2 回目 | |
| 3 回目 | |

●for の後に input や print 文を入れる場合は字下げ(tab キー)が必要です。
字下げで for 文の中にあるという意味になります。

字下げなし

```
for i in range(1,4):
print(i)
```

字下げあり

```
for i in range(1,4)
    print(i)
```

練習 1 input を表示してみよう。

```
input("点数を入力してください")
```

練習 2-1 input が 3 回繰り返し表示されるようにします。

```
for i in range (1,4):
    input("点数を入力してください")
```

練習 2-2 練習 2-1 と別の方法で input が 3 回繰り返し表示されるようにします。

```
for i in range (1,4,1):
    input("点数を入力してください")
```

- ・range(開始、終了、増分)
- ※増分が 1 つずつ増える場合は省略可能

練習3 3回繰り返した後に成功だねと表示させるプログラムを組んでいきます。

```
for i in range (1,4):  
  
    input("点数を入力してください")  
  
    print("成功だね")
```

※print 文を字下げしてしまうと for の中にあるとみなされるので 3回成功だねと表示されます。
なので字下げはしないでください

練習4 input に入れた数字が足されていくプログラムを作ります。

```
gokei=0  
  
for i in range (1,4):  
  
    a=input("点数を入力してください")  
    b=int(a)  
    gokei=gokei+b  
  
print(gokei)
```

- ・変数 gokei を 0 にしておきます。
- ・gokei=b は input に入れた数字を足していくプログラムです。
- ・a、b、gokei は for 文の中にいれるので字下げをしてください。
- ・最後に足した gokei を表示します。

●ちなみに gokei の値を追っていく(トレース)すると

| 繰り返し回数 | gokei+ | b の値(input に入った値) |
|--------|--------|-------------------|
| 1 回目 | | |
| 2 回目 | | |
| 3 回目 | | |

| 最終 gokei 値 |
|------------|
| |
| |
| |

超重要 Python と DNCL の比較

Python

```
gokei=0  
  
for i in range (1,11):  
  
    a=input("点数を入力してください")  
    b=int(a)  
    gokei=gokei+b  
  
print(gokei)
```

DNCL

- (1) gokei = 0
- (2) i を 1 から 10 まで 1 ずつ増やしながら繰り返す：
- (3) | a=【外部からの入力】
- (4) | gokei= gokei + a
- (5) 表示する (gokei)

注意 Python の【for i in range (1,10)】は 10 を含まない!

DNCL の【i を 1 から 10 まで 1 ずつ増やしながら繰り返す：】は 10 を含む

繰り返し 問題

① 1 から 4 までの数字の合計を求めるプログラム

- (1) `sum = 0`
- (2) `i` を 1 から 4 まで 1 ずつ増やしながら繰り返す :
- (3) `└ sum = sum + i`
- (4) 表示する (`sum`)



| 実行回数 | i | sum |
|------|---|-----|
| 1 | 1 | 1 |
| 2 | 2 | ① |
| 3 | ② | ③ |
| 4 | ④ | ⑤ |

② 1 から 9 までの整数のうち奇数だけを合計するプログラム

- (1) `sum = 0`
- (2) `i` を 1 から 9 まで 2 ずつ増やしながら繰り返す :
- (3) `└ sum = sum + i`
- (4) 表示する (`sum`)



| 実行回数 | i | sum |
|------|---|-----|
| 1 | 1 | 1 |
| 2 | 3 | ① |
| 3 | ② | ③ |
| 4 | ④ | 16 |
| 5 | 9 | ⑤ |

③ 「3、2、1」と数字を順番に表示した後、成功と表示するプログラム

- (1) i を 3 から まで減らしながら繰り返す：
- (2) └ 表示する
- (3) 表示する（"成功"）

④ データを 5 回入力し、その合計を求めるプログラム

- (1) sum = 0
- (2) i を 1 から まで 1 ずつ増やしながら繰り返す：
- (3) └ a = 【外部からの入力】
- (4) └ sum = sum +
- (5) 表示する（sum）

もし使うなら・・・外部からの入力は適当に入れてください

| 実行回数 | ②が入る | sum |
|------|------|-----|
| 1 | | 1 |
| | | |
| | | |
| | | |
| | | |

⑤ 1 から 5 までの数字の中に偶数がどれだけあるのかを求めるプログラム

- (1) count = 0
- (2) i を 1 から まで 1 ずつ増やしながら繰り返す：
- (3) └ もし $i \% 2 == 0$ ならば
- (4) └ count = + 1
- (5) 表示する（count）

●もし終わった人は⑤を python で書いてみよう