

python3講習

4 S 野口史遠

第2回目

制御構造ツール

環境

- python 3.9 ,3.10 ,3.11
- ThinkPad L380 ubuntu22.04.3tls
- ThinkPad P1 Gen 3 Windows10

if文

- if文は条件に応じて異なるコードブロックを実行します。
- 条件がTrueの場合、ifブロックが実行されます。

例:

```
x = 10

if x > 5:
    print("xは5より大きい")
elif x == 5:
    print("xはちょうど5です")
else:
    print("xは5より小さい")
```

xは5より大きい

for文

- for文はイテラブルオブジェクト内の要素を繰り返し処理します。
- リスト、タプル、文字列などのイテラブルオブジェクトに対して使用できます。

例:

```
fruits = ["りんご", "バナナ", "オレンジ"]  
for fruit in fruits:  
    print(fruit)
```

```
りんご  
バナナ  
オレンジ
```

range関数

- range()関数は指定された範囲の整数のシーケンスを生成します。
- 通常、forループと組み合わせて使用されます。

例:

```
for i in range(5):  
    print(i)  
0
```

```
1  
2  
3  
4
```

break文とcontinue文，ループにおけるelse文

- break文はループを強制的に終了します。
- continue文は現在のイテレーションをスキップし、次のイテレーションに進みます。
- ループにおけるelse文はループが正常に終了した場合に実行されます。

```
for n in range(2,10):  
    for x in range(2,n):  
        if n % x ==0:  
            print(n, 'equals', x, '*', n//x)  
            break  
    else:  
        print(n, "is a prime number")
```

```
4 equals 2 * 2  
6 equals 2 * 3  
8 equals 2 * 4  
9 equals 3 * 3  
9 is a prime number
```

pass文

- pass文は何もしないステートメントです。
- 一時的にコードのブロックを空にするために使用されます。

例:

```
if True:  
    pass
```


関数の定義

- 関数は複数の文をまとめた再利用可能なコードブロックです。
- defキーワードを使用して関数を定義します。

例:

```
def greet(name):  
    print("こんにちは, " + name)
```

```
greet("Altair")
```

```
こんにちは, Altair
```

キーワード引数

- キーワード引数を使用すると、引数の順序を気にせずに関数を呼び出すことができます。

例:

```
def show_info(name, age):  
    print("名前:", name)  
    print("年齢:", age)  
  
show_info(age=25, name="Bob")
```

```
名前: Bob  
年齢: 25
```

lambda式

- lambdaを使用して無名の小さな関数を作成できます。
- 通常、簡単な操作に使用されます。

例:

```
square = lambda x: x**2  
print(square(4))
```

16

課題

1から10までの整数をループで表示し、偶数の場合には"偶数"と表示し、奇数の場合には"奇数"と表示するプログラムを書いてください.

```
for num in range(1, 11):  
    if num % 2 == 0:  
        print(f"{num}は偶数")  
    else:  
        print(f"{num}は奇数")
```

1は奇数
2は偶数
3は奇数
4は偶数
5は奇数
6は偶数
7は奇数
8は偶数
9は奇数
10は偶数

次回

データ構造とモジュール

おまけ

```
import serial

ser = serial.Serial('/dev/ttyACM0', 9600, timeout=None) # "COM19"
while True:
    line = ser.readline() # ここで一行データを取得するがbyte型
    stripped_str = str(line, 'ascii').strip() # byte型を文字列に変換して前後の空白改行除去
    data = int(stripped_str) # 文字列を数値に解釈し直す
    print(data)
```

```
//Arduino
void setup() {
    Serial.begin(9600);
}

int data = 0;
void loop() {
    data++;
    Serial.println(data);
    delay(1000);
}
```