

## Lecture 3

プログラミング演習Ⅰ その3

# 本日の演習の流れ

1. 先週の残りの問題
2. 条件分岐の【if 文】について
3. pythonの対話型シェルについて

授業のホームページ

<http://amth.mind.meiji.ac.jp/courses/PE1/>

# 演習問題の目安時間について

C言語 #7. 10分

C言語 #8. 10分

C言語 #9. 20分

C言語 #10. 20分

C言語 #11. 20分

Python #1. 20分

Python #2. 20分

# 計算演算子の優先順位について

1つの式に複数の演算記号があるときには、次のように優先順位が決められている。

第1順位     $*$  ,  $/$  ,  $\%$

第2順位     $+$  ,  $-$

**7** 以下のコードを実行し、出力を確認しなさい。

```
1 #include <stdio.h>
2 int main(void){
3     int a;
4     a = 5 + 3 * 4 - 2;
5     printf("%d\n",a);
6     a = (5 + 3) * (4-2);
7     printf("%d\n",a);
8     return 0;
9 }
```

## 条件分岐の【if 文】について

8

与えられた条件が成立したとき、あるいは成立しないときに、それぞれ異なった処理を行う分岐処理では、if文を使用する。次のプログラムを実行せよ。

```
#include <stdio.h>
int main(void){
    int i;
    i = 5;
    if(i > 4){
        printf("i is greater than 4.\n");
    }
    return 0;
}
```

解説：関係演算子 (relational operators)

`x == y` // 等しい

`x != y` // 等しくない

`x < y` // より小さい

`x <= y` // 以下

`x > y` // より大きい

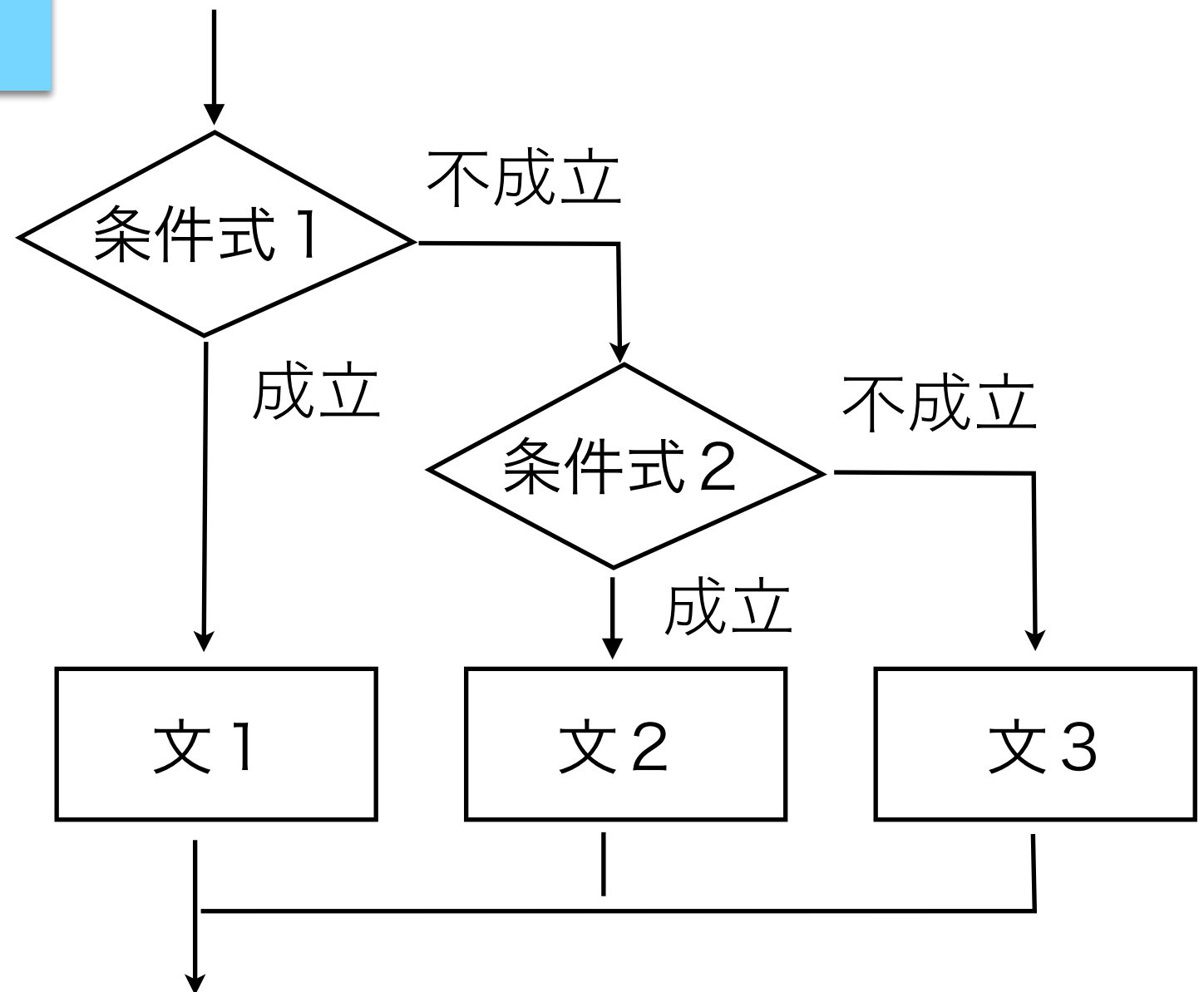
`x >= y` // 以上

# if ~ else if ~ else 文による判断

2つ以上の条件を組み合わせる判断することができる。

< if, else if, else 文の構文 >

```
if (条件式 1) {  
    文 1;  
} else if (条件式 2) {  
    文 2;  
} else {  
    文 3;  
}
```



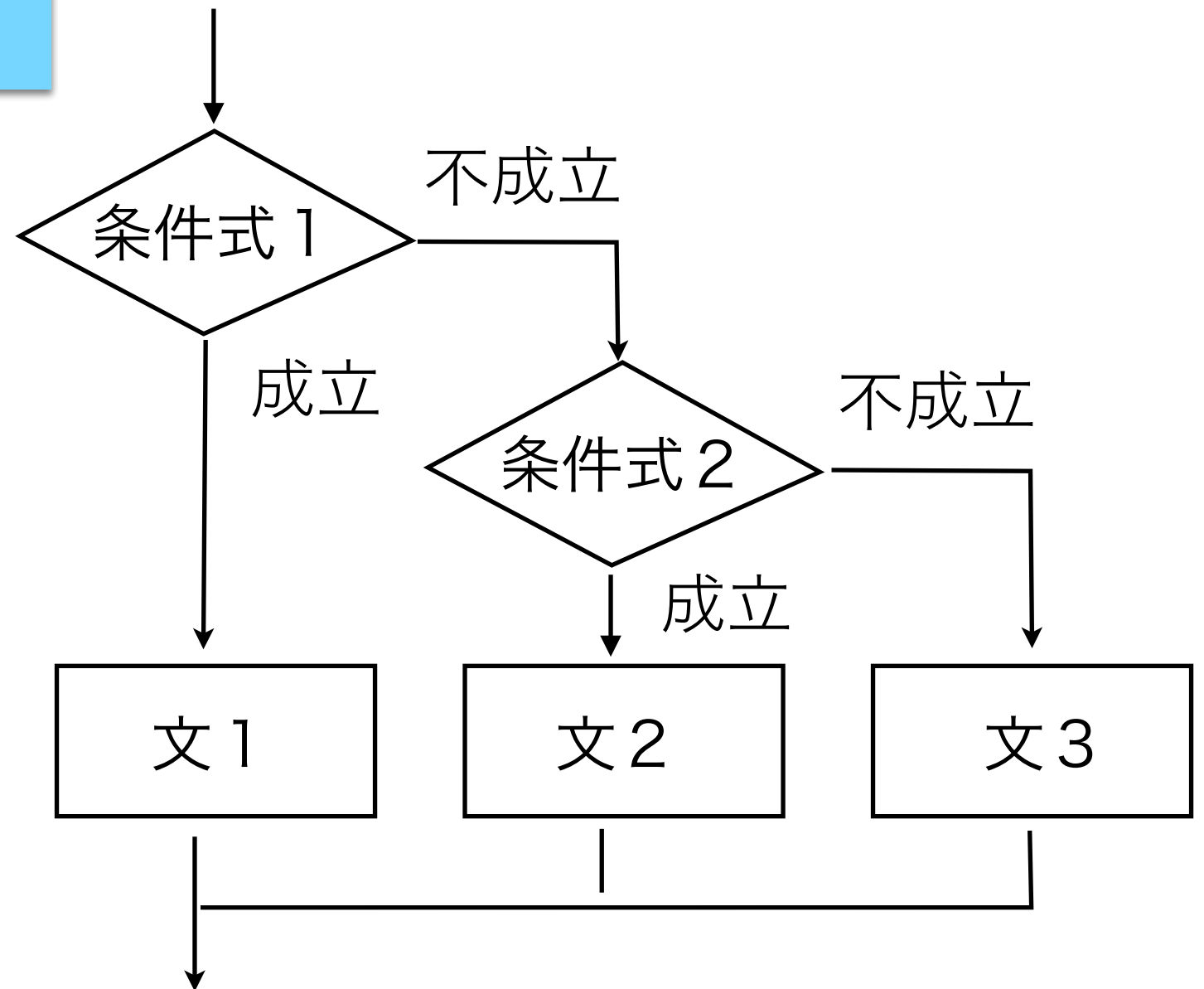
# if ~ else if ~ else 文による判断

2つ以上の条件を組み合わせる判断ができる。

< if, else if, else 文の構文 >

```
if(条件式1){  
    "if block"  
}else if(条件式2){  
    "else if block"  
}else{  
    "else block"  
}
```

「{」と「}」の間をブロック (block) という





# if ~ else if ~ else 文による判断

9

次のプログラムを実行した際、最終的にnはどのような値となるか、確認しなさい。

```
1  #include <stdio.h>
2  int main(void){
3      int n = 10;
4      if(n > 7){
5          n = 5;
6      }else if( n > 4){
7          n = 3;
8      }else{
9          n=0;
10     }
11     printf("the value of n is: %d\n",n); // nの値を出力する
12     return 0;
13 }
```

//変数nに初期値として10を代入  
//もしnが7より大きいなら  
//nに5を代入する  
//nが4より大きい場合  
//nに3を代入する  
//そうでなければ  
//nに0を代入する



# if ~ else if 文による判断

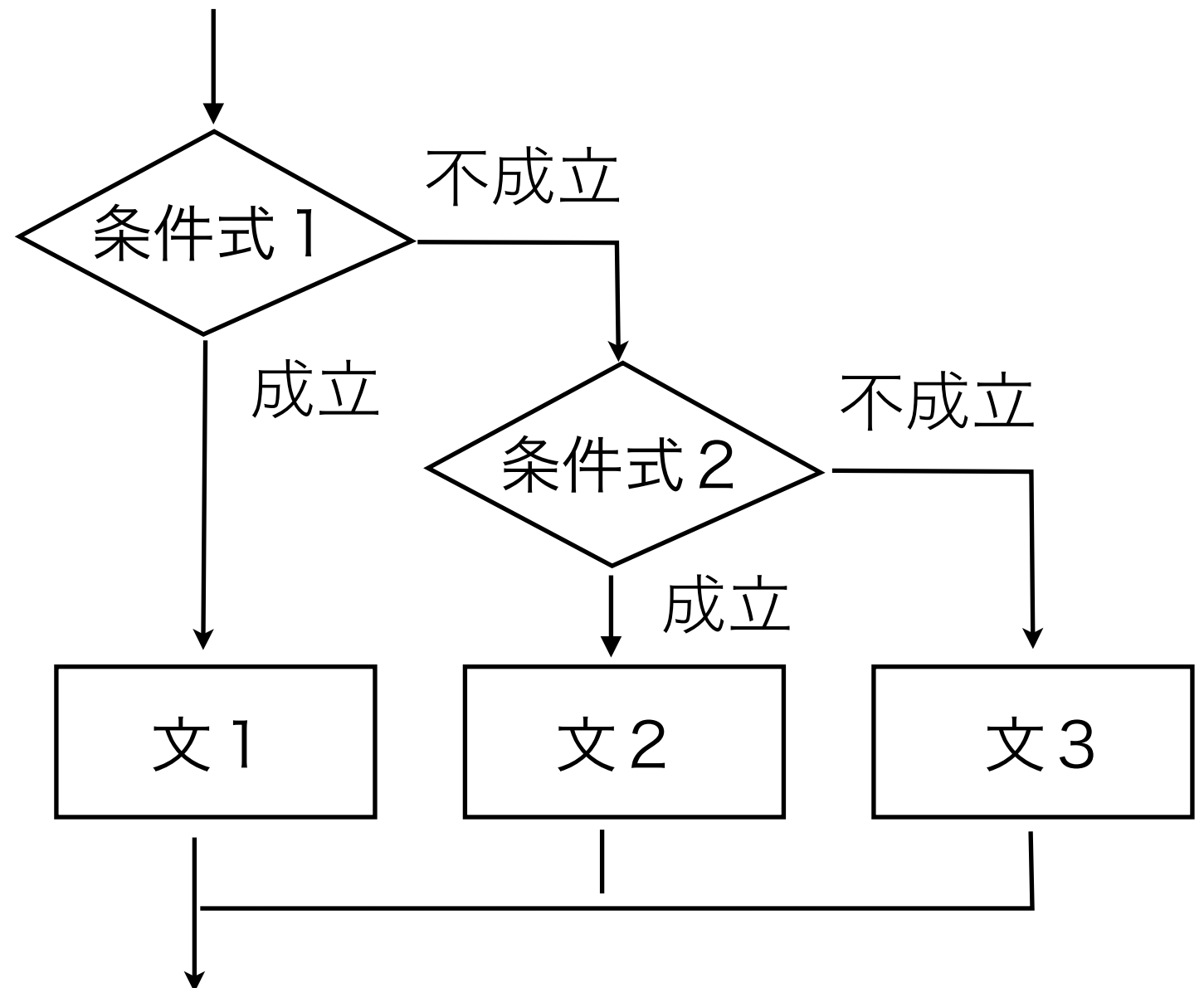
10

if の条件分岐を使い，与えられたnが9より大きいかどうかを判断し，大きいなら10とし，9以下で5より大きい場合はnを8とし，そうでなければ0とするプログラムを作りなさい。

## ヒント

< if, else if, else文の構文 >

```
if(条件式1){  
    文1;  
}else if(条件式2){  
    文2;  
}else{  
    文3;  
}
```



# 条件分岐と計算演算子について

**11** 次のような動作をするプログラムを作成せよ.

1. キーボードから整数の入力を求める.
2. 入力された数値が奇数なら「odd」と表示する.
3. 入力された数値が偶数なら「even」と表示する.

ヒント

「%」を使用すること

# Python とは

Pythonはさまざまな分野で利用されている汎用のプログラミング言語である。動的プログラミング言語（dynamic programming language）の一種で、事前にコンパイルが不要で実行時に機械語に変換する処理系を持つ言語である。

PythonはGuido van RossumがCWI (Centrum Wiskunde & Informatica)勤務時の1989年に開発を開始し、1991年に初期バージョンが公開された。



# Python の特徴

## インタープリタ言語 (Interpreted language)

プログラムを逐次解釈して実行する仕組みになっており、ソースコードをコンパイル作業を必要としない。

## スクリプト言語 (Scripting language)

動的型付け(dynamic typing)言語であること、インタープリタ言語であること、ローレベル (low level) の記述が不要で小規模のプログラム向き、といった特徴を備えた言語を指す。

## グルー言語 (glue language)

python は、よく glue language であると言われている。日本語で言えば「糊言語」である。これは、他の言語のプログラムを容易に結合させることができる言語であることを意味する。

# Why Python?

## IEEE Spectrumのプログラミング言語ランキング

順位	言語
1	Python
2	Java
3	C言語
4	C++
5	R言語

2016年度

順位	言語
1	Python
2	Java
3	C言語
4	C++
5	JavaScript

2020年度

# Python とは

スクリプト言語には処理を簡潔に書けるという特徴がある。簡単な例を見てみよう。c++とc言語で "Hello World" と出力するプログラムを書くと以下のようなになる。

## C++

①

```
1 #include <iostream>
2 using namespace std;
3 int main(void){
4     cout << "hello, World!\n";
5     return 0;
6 }
```

② c++ hello.cpp

③ ./a.out

## c言語

①

```
1 #include <stdio.h>
2 int main(void){
3     printf("Hello World\n");
4     return 0;
5 }
```

② cc hello.c

③ ./a.out

一方、Pythonでは、以下の通り、たったの1行である。

```
print("Hello World");
```

# Pythonで「Hello World」

1

以下のように `hello.py` を作成して、保存し、  
「`python hello.py`」で実行してみよう。

```
1 print("Hello World"); ←———— hello.py
```

出力

```
[ Lec14_pro python hello.py  
Hello World
```


NOTE

Python ではコメントは#を先頭に付けて記述する。

```
1 print("Hello World") # this is a comment
```



# Python の対話型シェル(interactive shell)について

Pythonは電卓として、簡単な計算ができる。特にターミナルで「python」を打てば、Python の対話型シェルが使える。式を入力するとPythonが計算を行う。ENTERを押せば、結果がすぐ表示される。試してみよう。  終了には quit() か ctrl-d

2

```
Lec14_pro python
Python 2.7.10 (default, Feb  7 2017, 00:08:15)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.34)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

数の足し算と引き算は、加算演算子「+」と減算演算子「-」を使う。次の通りである。

```
>>> 1+3
4
>>> 1+3.5
4.5
>>> -1+2.5
1.5
```

# Python の対話型シェル(interactive shell)について

## 2 (continued)

掛け算には, 乗算演算子「\*」を使う.

```
[>>> 3*2
```

```
6
```

```
[>>> 3.5*1.5
```

```
5.25
```

割り算には, 除算演算子「/」を使う.

```
[>>> 3.0 / 2.0
```

```
1.5
```

```
[>>> 3 / 2
```

```
1
```

```
[>>> 6 / 4
```

```
1
```

# Python の対話型シェル(interactive shell)について

## 2 (continued)

余りだけが必要な場合は、乗余演算子（%, モジュロ, modulo）を使う

```
[>>> 9 % 2  
1
```

累乗（べき乗）の計算は「\*\*」を使う.

```
[>>> 2 ** 2  
4
```

```
[>>> 2 ** 10  
1024
```

```
[>>> 1 ** 10  
1
```

# Python の対話型シェル(interactive shell)について

## 2 (continued)

この例から分かるように、括弧を使って計算演算子を組み合わせ、より複雑な式を扱うことができる。

```
>>> 5 + 5 * 5
30
>>> (5+5)*5
50
```

Python標準ライブラリのmathモジュール (module) で一般的な関数 (sin(), cos(), tan(), asin(), acos(), atan()) が入っている。  
ネイピア数eも定義されている。

```
[>>> import math
[>>> math.e
2.718281828459045
[>>> math.sin(math.pi/2)
1.0
```

# Python の対話型シェル(interactive shell)について

## 2 (continued)

さらに複雑なPythonプログラムを設計するために, 変数 (ラベル (label) と呼ぶ) に数を代入する.

```
>>> a = 3
```

```
>>> a + 1
```

```
4
```

```
>>> a
```

```
3
```

```
>>> a = a + 5
```

```
>>> a
```

```
8
```

```
>>> math.sin(a)
```

```
0.9893582466233818
```