

プログラムの基本構造1/2

この時間の目標

- プログラムの基本構造
- S プログラムの基本構造がよく理解でき、手順を工夫して活用しようと思った
- A プログラムの基本構造がよく理解できた
- B プログラムの基本構造が理解できた
- C プログラムの基本構造が理解できなかった

プログラミングを始めよう

- プログラムの開発環境を用意する
 - Googleコラボトリーが使える人
 1. Googleコラボトリーを開く
 2. [ファイル]→[ノートブックを新規作成]
 3. 左上のファイル名を[22j1-211.ipynb]に変更
 - Googleコラボトリーが使えない人
 1. [スタート]→[Anaconda 3(64bit)] → [Anaconda Prompt(Anaconda)]とクリック
 2. 表示された黒い画面に IDLE と入力して[Enter]キーを押す
 3. 白い画面が実行環境が表示される

演算

- 計算や文字列操作など
- 演算子

	演算子	数学での表記
足し算	+	+
引き算	-	-
掛け算	*	×
割り算	/	÷

	演算子	例	結果
商	//	9//4	2
余り	%	9%4	1
累乗	**	2**3	8

演算

- 大小の比較など
- 比較演算子

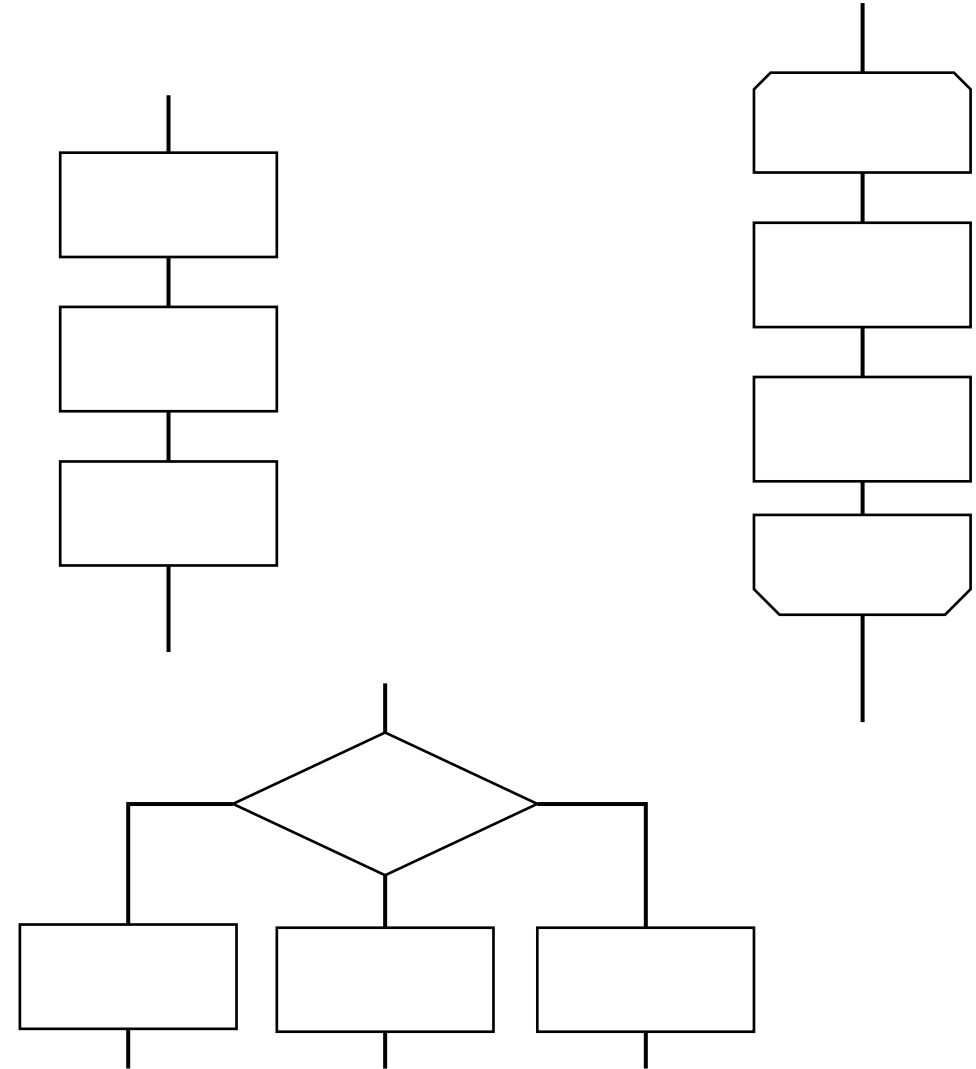
演算子	意味
$A < B$	AはBより小さい
$A \leq B$	AはB以下
$A > B$	AはBより大きい
$A \geq B$	AはB以上
$A == B$	AとBは等しい
$A != B$	AとB は等しくない

値の扱い方

- 数には名前がないので変数を用いる
- 変数を用いて抽象化
- 変数の処理方法を記述するのがプログラミング
 - 変数が決まれば答えが決まる

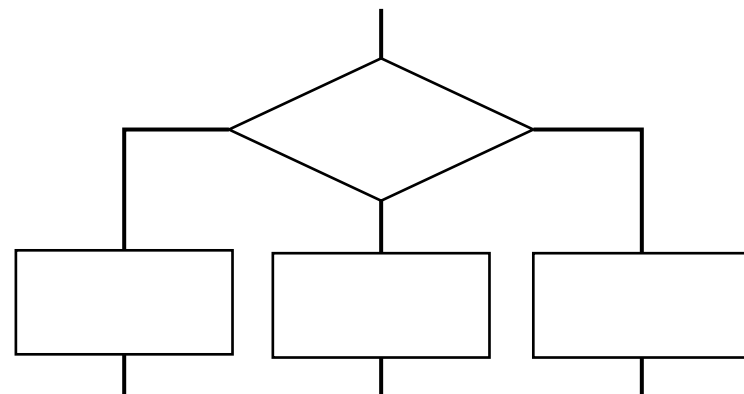
制御構造

- 順次構造
 - 順番に実行
- 分岐構造
 - 条件で処理を選択
- 反復構造
 - 条件に従い繰り返し処理



分岐構造

- 条件によって命令を選択
 - if文を使用
 - 処理2・処理3は省略可
 - elifは複数使用可



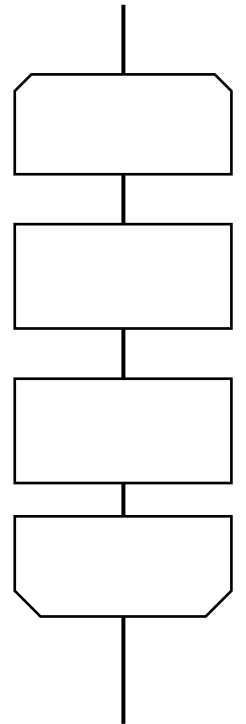
```
if 条件1:  
    処理1  
elif 条件2:  
    処理2  
else:  
    処理3
```


反復構造

- 条件に従い処理を繰り返し実行
- for文とwhile文がある
 - 繰り返し回数を指定するときはfor文
 - 条件を満たすまで繰り返す時はwhile文

for カウンタ 範囲:
処理

while 条件:
処理



inputの使い方 文字列の扱い

情報オリンピック予選に必要な「技」

input()関数の使い方

- 文字を入力させる
 - input()
 - 数を入力しても文字扱い
- 数値を入力させる
 - int(input())
- スペースで区切って複数データを入力させる
 - input().split()
 - リストとして入力される

文字列の扱い

- 文字列はリストと同じように扱える
- 例

プログラム

```
moji='ABCDEF'  
print(moji)  
print(moji[3])  
print(moji[1:3])
```

実行例

```
ABCDEF  
D  
BC
```