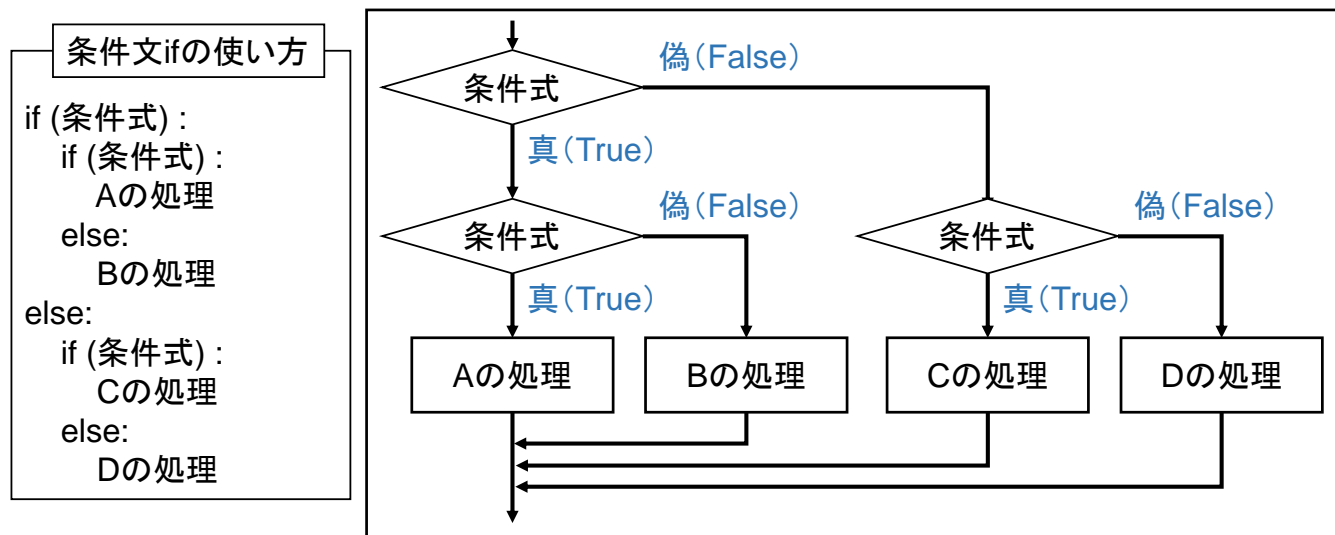


入れ子になっている条件処理

これまでの条件処理では、1つの「if - else -」の処理を書いていましたが、if、else のそれぞれの中に条件処理を書くこともできます。このような場合は、新たにインデントを下げて、1つのブロックとして書く必要があります。



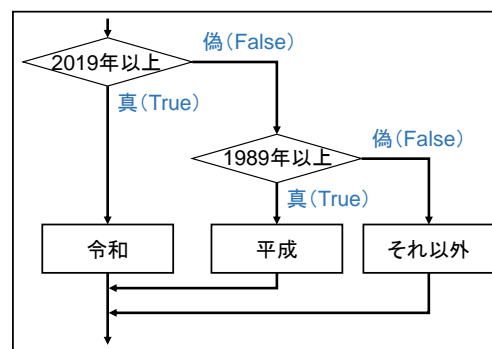
(例 1) 次のプログラムを入力し、実行してみよう。(テストの判定)

```

1 math = int(input("数学の点数:"))
2 english = int(input("英語の点数:"))
3
4 if(math >= 40):
5     if(english >= 40):
6         print("数学と英語どちらも合格です")
7     else:
8         print("数学は合格で英語は不合格です")
9 else:
10    if(english >= 40):
11        print("英語は合格で数学は不合格です")
12    else:
13        print("数学と英語どちらも不合格です")
    
```

(問 1) 右のフローチャートを参考にして、
西暦から和暦を判定するプログラムを書いてみましょう。
1989 年以降であれば平成、2019 年以降であれば令和、
それ以外は対象外とするプログラムを作ってみよう。

HINT : 条件式で「>、>=」を使う場合は、
より大きい数値から判定するとうまういきます。
※逆に「<、<=」を使う場合は、小さい数値から
判定するとうまういきます。



論理演算子を使った条件処理

複雑な条件処理は、入れ子構造を使うことで、実現することもできますが、プログラムが長くなりがちです。複数の条件を 1 つにまとめることのできる「論理演算子」というものがあります。

演算子	書き方	例	意味
and	A and B	a == 40 and b == 40	a=40 かつ b=40 (数学では $A \cap B$)
or	A or B	a == 40 or b == 40	a=40 または b=40 (数学では $A \cup B$)
not	not A	not a == 40	a=40 ではない (数学では \bar{A})

(例 2) 次のプログラムを入力し、実行してみましょう。

(例 1) を改良してどちらも 40 点以上で合格、そうでなければ合格とするプログラム

1	math = int(input("数学の点数:"))
2	english = int(input("英語の点数:"))
3	
4	if(math >= 40 and english >= 40):
5	print("合格です")
6	else:
7	print("不合格です")

(問 2) (例 2) を改良して、数学と英語どちらかが 40 点なら合格を出力するプログラムを書いてみよう。

論理演算子を使えば、入れ子構造にしなくても短いプログラムを作ることができます。しかし、論理演算子で作ったプログラムは細かい部分まで確認するようなプログラムは困難です。例えば、例 1 では、1 つ 1 つの点数をチェックして場合分けできましたが、例 2 では、どちらも合格か不合格かのみ判断ができ、数学だけ合格していたなどの判定は難しいです。

発展 1 年齢を入力して、映画館の料金を出力するプログラムを書いてみましょう。ある映画館は次のとおりの料金です。

- ・ 3 歳未満：無料
- ・ 12 歳以下：こども料金の 500 円
- ・ 18 歳以下：1000 円
- ・ 60 歳未満：1800 円
- ・ 60 歳以上は 1200 円

発展 2 (例 1) のプログラムに物理の点数も入れるように書き換えてみてください。合格の条件は、3 教科がすべて 40 点以上の場合、または 3 教科の合計点が 120 点以上の場合として、それ以外は不合格です。

発展 3 三角形の 3 辺の入力し、次の三角形の種類から最も適切なものを判定・表示するプログラムを作ってみよう。 三角形の種類は、正三角形・直角三角形・二等辺三角形・普通の三角形・三角形ではない

HINT：直角三角形の条件は「2 辺の長さの 2 乗の和は残りの長さの 2 乗と等しい (三平方の定理)」

HINT：三角形の成立条件は次の①かつ②

- ① すべての辺の長さが 0 より大きい
- ② どの 2 辺の和も残りの辺より大きい