講座

アルゴリズムを学ぼう！

2024年1月22日予定

担当：新田

アルゴリズムとは？

ある問題を解決する方法や、ある目標を完了するための方法が書かれた一連の「手順」

問題を解くための手順や計算方法

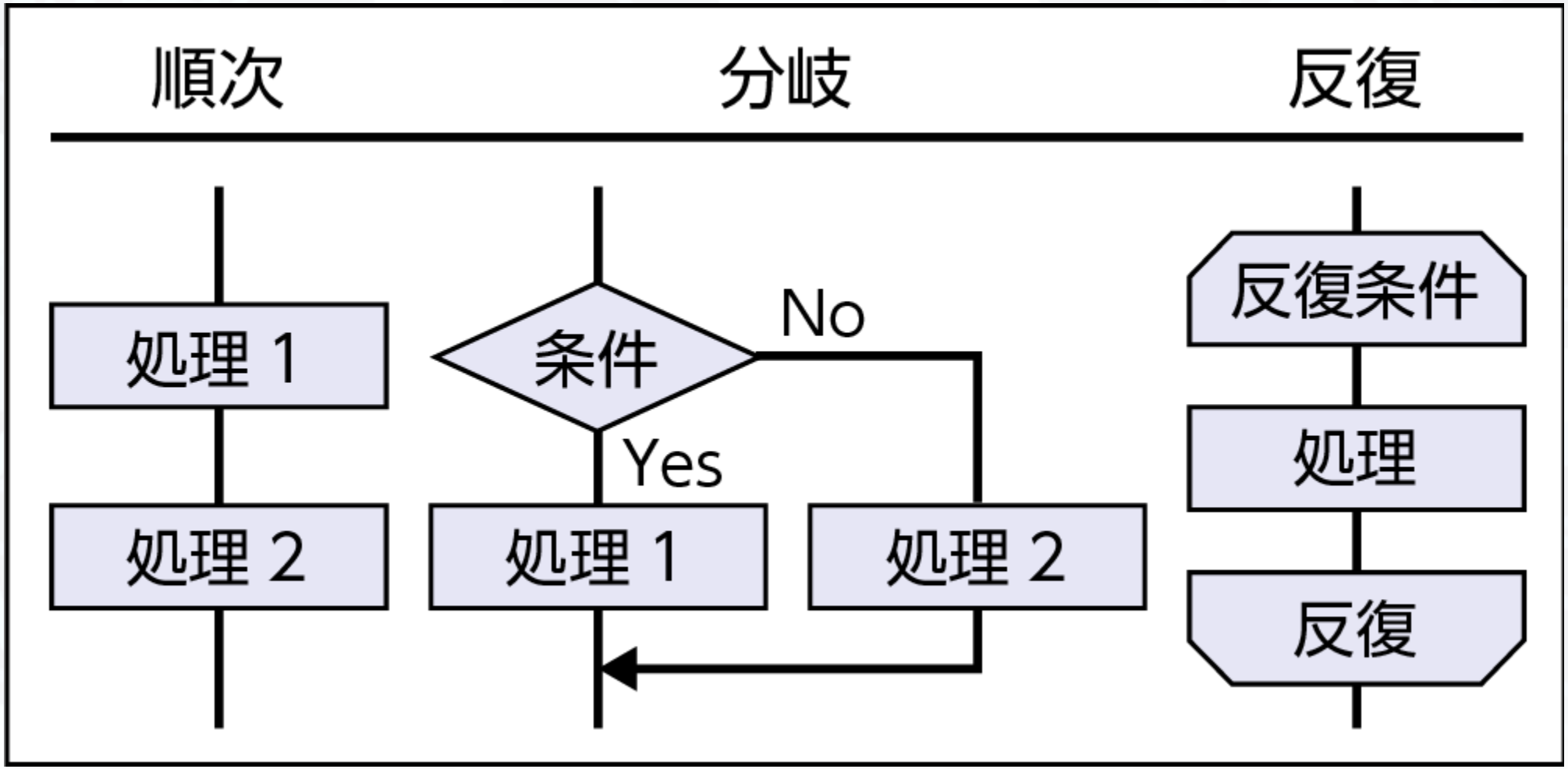
身近な例

- ICカード（電子決済）
- 電車の乗り換え案内
- パスワード認証
- 地図アプリ
- 料理のレシピ
- レジの会計
- おすすめ動画・広告
- マッチングアプリ

フローチャート　アルゴリズムを記述した図

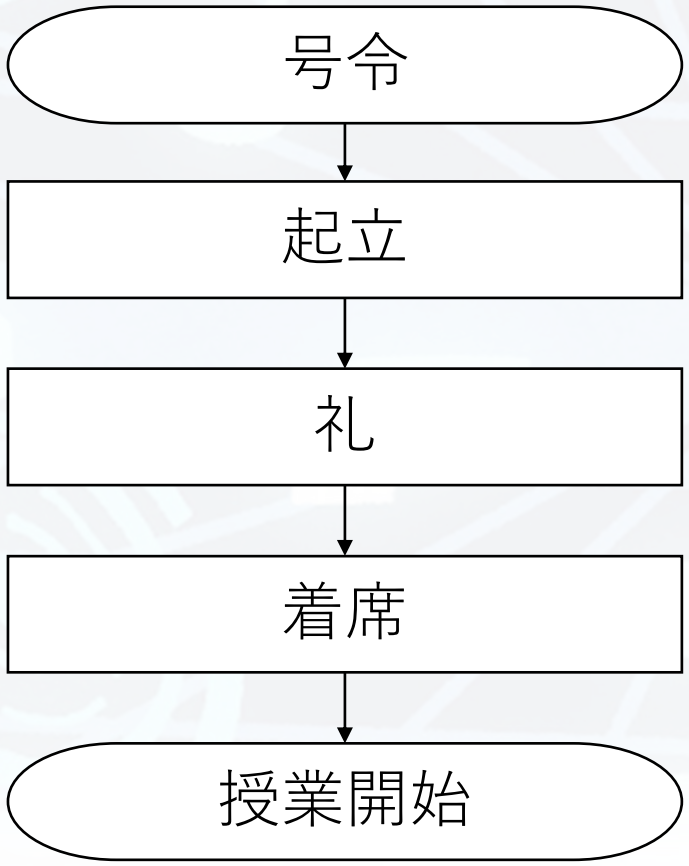
処理  任意の種類の処理機能を表す。	書類  人の読める媒体上のデータを表す。	結合子  ページ内でチャートを分割する
判断  条件によって分岐する。	端子  フローの入り口と出口	結合子  次のページにチャートを分割する
入出力(データ)  ファイルへの入出力を表す。	手操作入力  手で操作して情報を入力する。	表示  ディスプレイへの表示を表す。
定義済み処理  あらかじめ定義された処理のまとまり（関数、サブルーチン）	手作業  人手による任意の処理を表す。	ループ  ２つセットでループの開始と終わり

アルゴリズムの基本構造



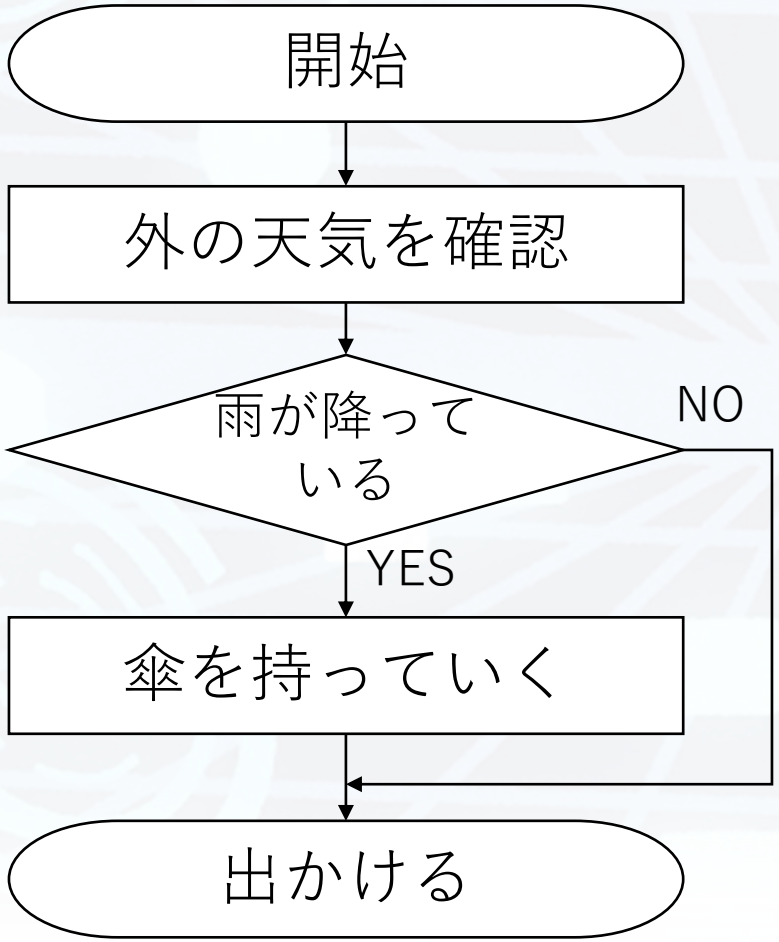
順次構造

ひとつの処理が終わったら次の処理に進む。



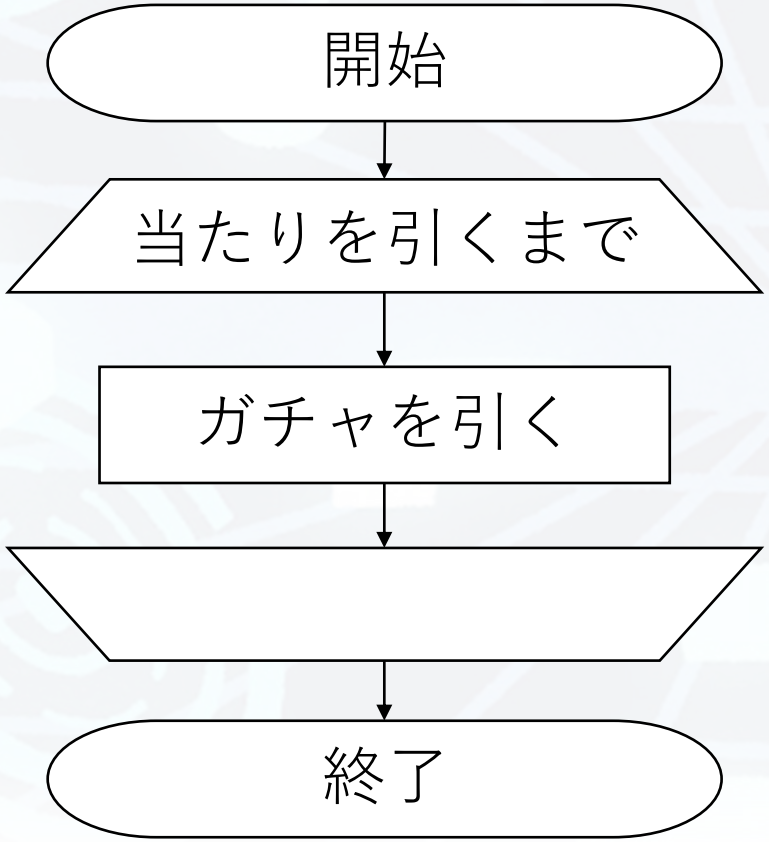
分岐構造

条件に基づいて実行する内容を変える。



反復構造

一定の条件を満たすまで処理を繰り返す。

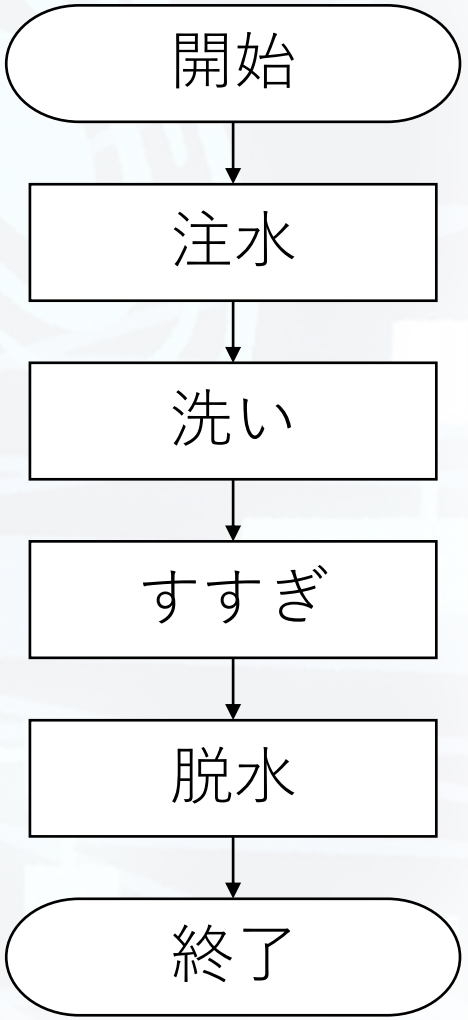


シーケンス制御

あらかじめ定められた手順や
判断によって制御の各段階を
順に進めていく制御。

「前の動作が完了したこと」
「一定時間経過した後」と
いった情報をもとに制御する。

例）洗濯機



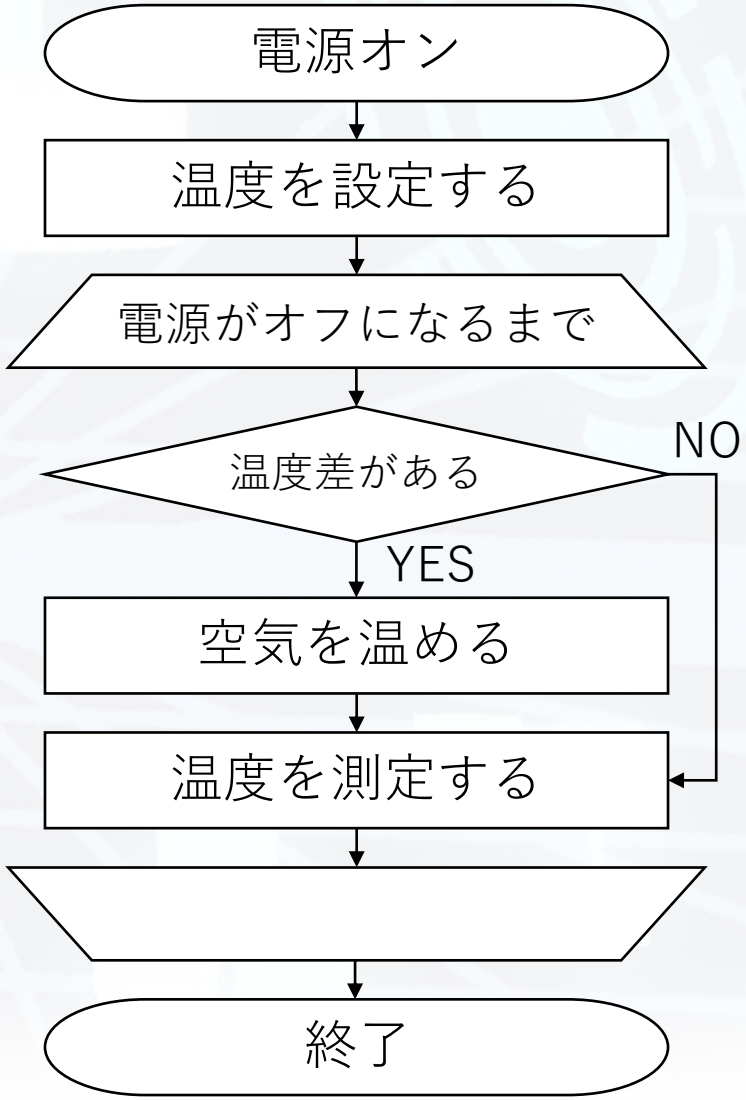
フィードバック制御

目標値と制御値の差を検出して、自動制御を行う。

制御を乱すような外的な作用（外乱）が生じても、その影響を適切に修正するように動作できる。

検出してから応答するまでに多少時間的な遅れが生じる

例）エアコン



利用者サイトのフォームにも・・・

https://rootsakihiro.shop/x/form.php

日報送信フォーム

振り返りの内容は、詳細にお書きください。

Googleアカウントを同時に複数ログインさせると、正常にフォームが送信されませんのでご注意ください。

名前(旧漢字は常用漢字にしてフルネームで入力ください)

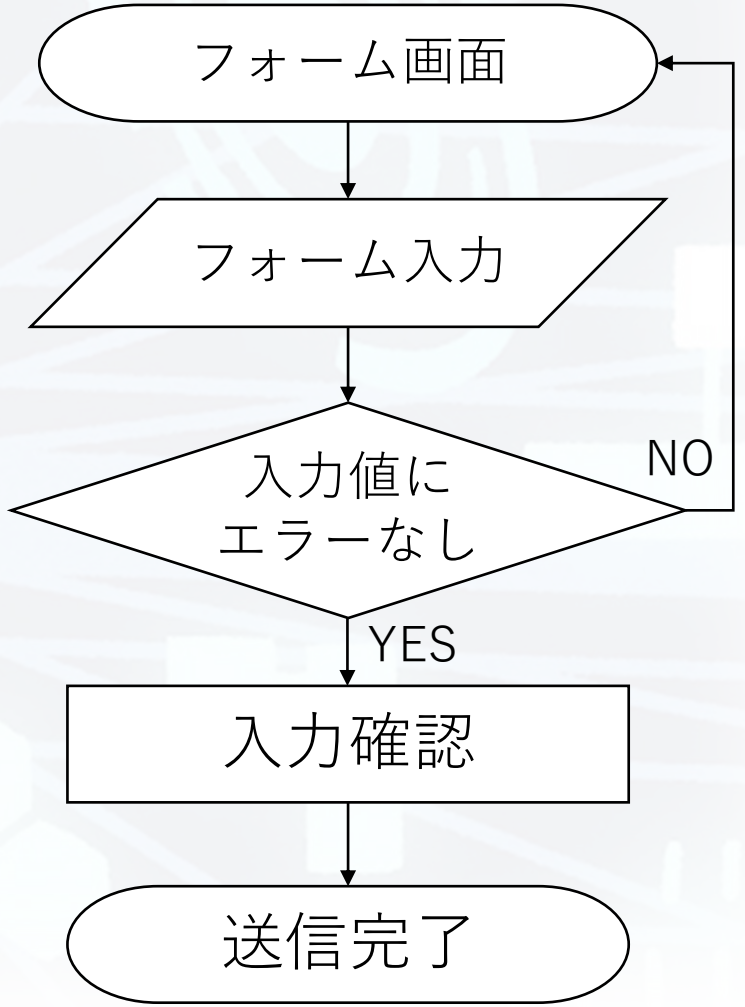
ドラえもん

利用日

2112/09/03

本日の体温(在宅の場合も必ず検温をお願いします)

36.0 ℃



自分の行動をフローチャートにしよう

- モーニングルーティーン
- 通所したときの１日の行動
- 模擬試験
- 休日の過ごし方
- お昼ご飯の決め方
- オリジナル料理のレシピ
- スキンケアの方法
- お金の使い道

：

おすすめツール：draw.io

https://www.drawio.com/integrations

アルゴロジック２

https://algo.jeita.or.jp/prm/2/index.html

◎は使用したコマンドブロック
数がその問題の最小値のとき

３つの基本構造(順次・分岐・反復）をマスターしよう！

０３　方向転換
０６　無限ループ
０８　十字回廊
１１　ELSEを使う
１２　IFを使う２

できる人は応用問題にチャレンジ！

プログラミングしてみよう！

- GoogleColaboratory
  https://colab.research.google.com/notebooks/intro.ipynb

- 簡単なおさらい

  変数の宣言：　　a = 1

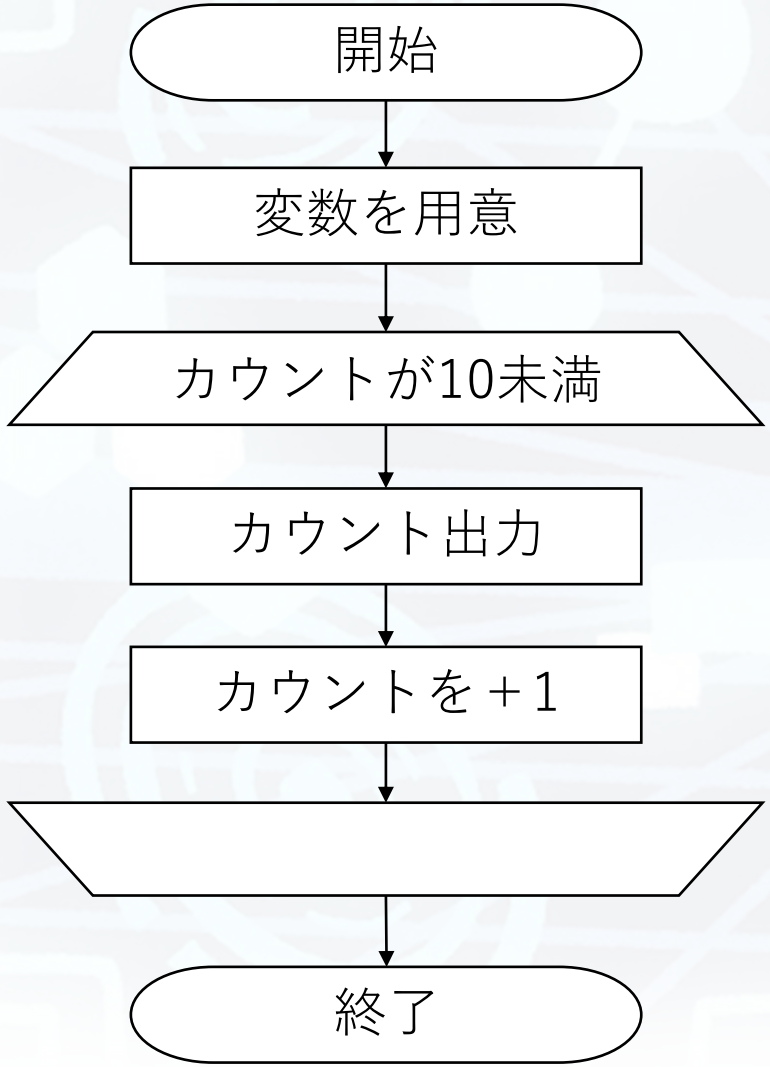
  条件分岐：if　elif　else

  反復処理：while:　for ○ in range(n)

  カウントを増やす：　count = count + 1　(count +=1)

  表示：　print( )

カウントを出力




```
count = 0

while count < 10:
    print(count)
    count = count + 1    # count +=1も可
```

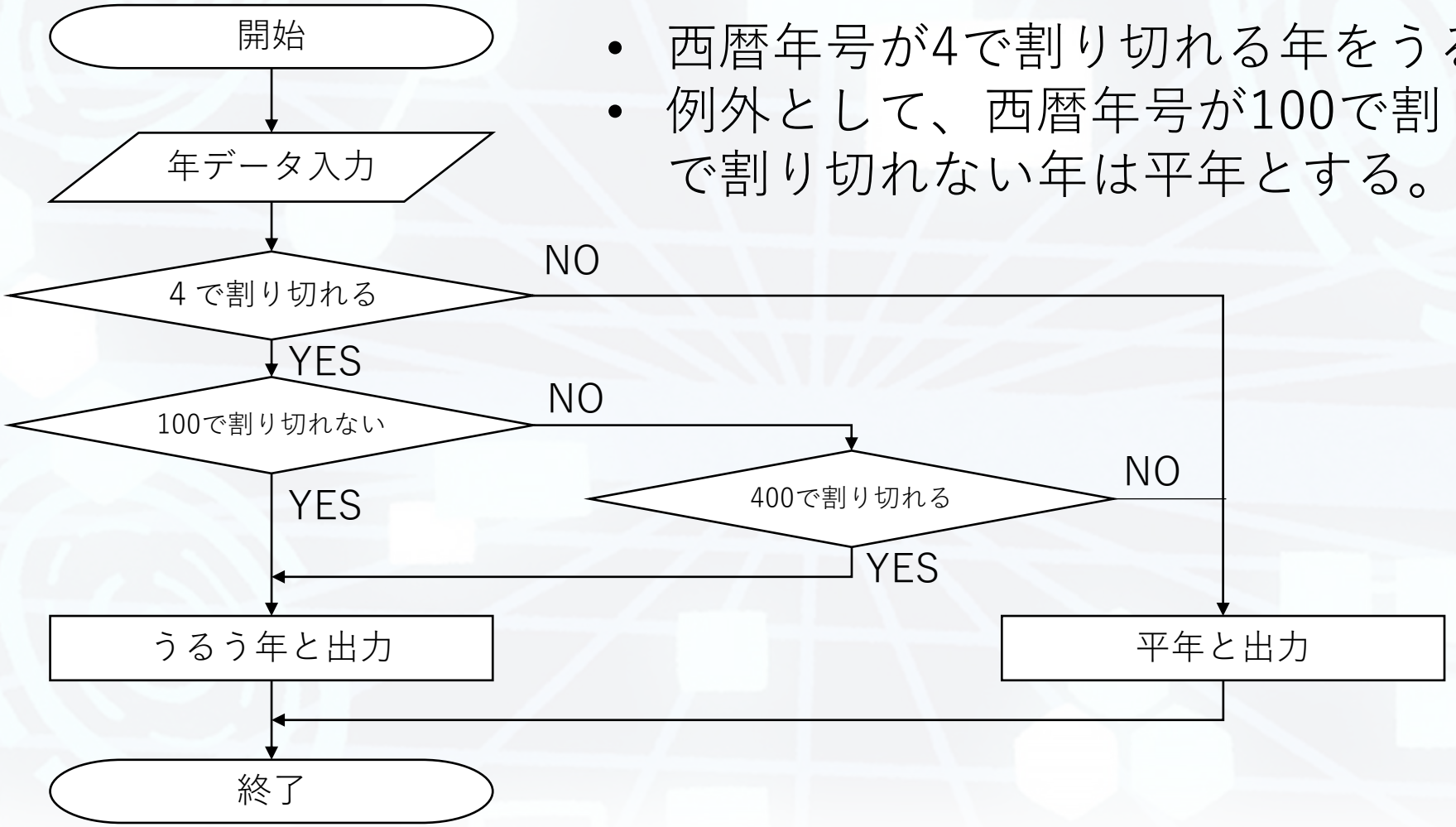
```
count = 0

while count < 10:
    print(count)
    count = count + 1        # count +=1も可
```

うるう年の判定

400年のうち97年がうるう年

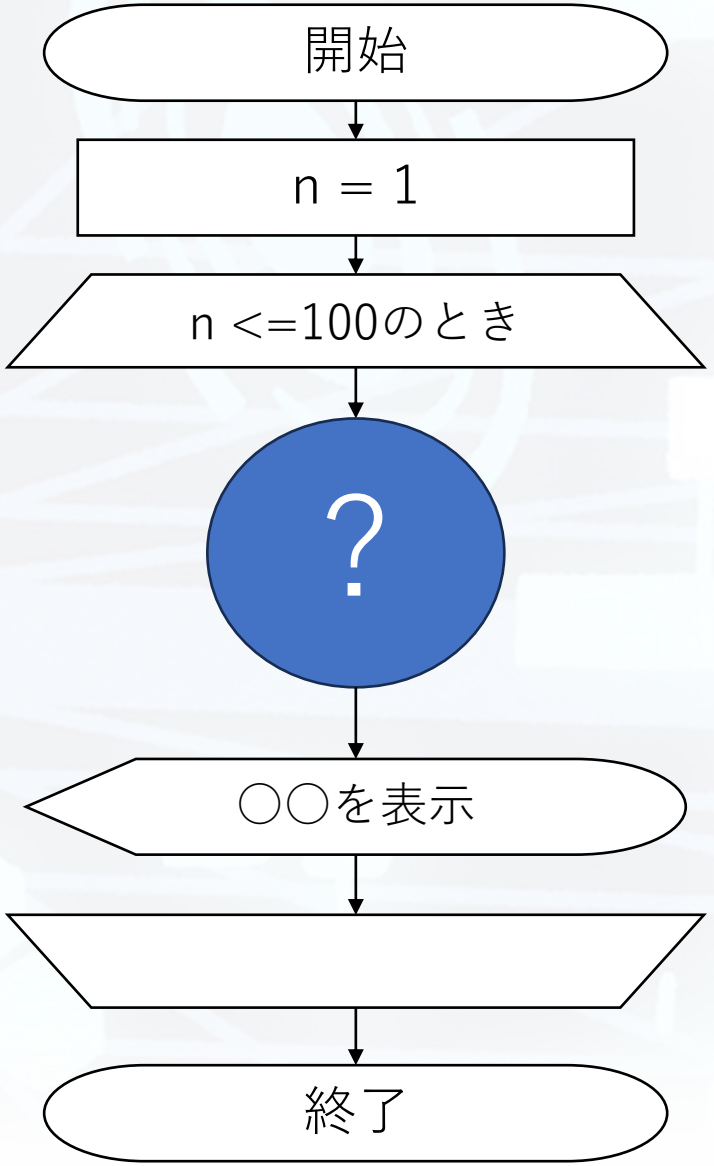
- 西暦年号が4で割り切れる年をうるう年とする。
- 例外として、西暦年号が100で割り切れて400で割り切れない年は平年とする。



FizzBuzz問題

ルール

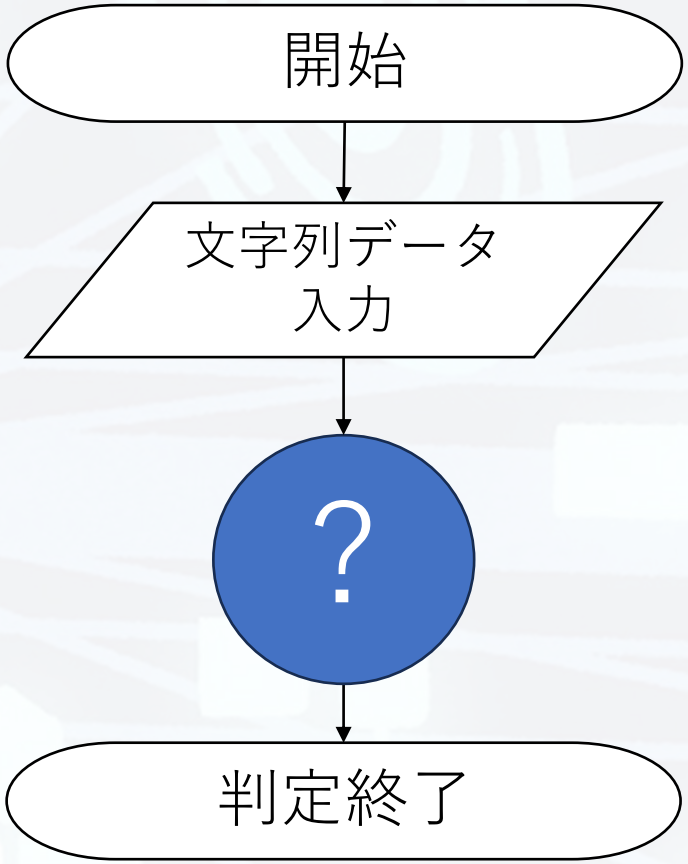
- 1から100までの数字を画面に表示する
- 3の倍数のときは数字の代わりに**Fizz**と表示する
- 5の倍数のときは数字の代わりに**Buzz**と表示する
- 3かつ5の倍数のときは数字の代わりに**FizzBuzz**と表示する



回文判定のアルゴリズム

ルール

- ある文字列が回文（上から読んでも下から読んでも同じになる）かどうかを判定する。
- 文字列の長さはnとする。
- 回文であるときはTrue、回文でないときはFalseを返す。



自動販売機のアルゴリズムを考えよう

- 条件を考えてみよう
  - 支払いは紙幣(1000円札）または硬貨（10円玉、50円玉、100円玉、500円玉)のみにする
  - 商品のまとめ買いはできない

- いろいろなケースを考えよう
  - 商品の在庫がないとき
  - おつり返却ボタンを押されたとき
  - お金以外が投入されたとき
  - 釣銭がないとき

並び替え（ソート）アルゴリズム

- **バブルソート**
- **選択ソート**
- **挿入ソート**
- **ヒープソート**
- **マージソート**
- **クイックソート**
- シェーカーソート
- コムソート
- シェルソート
- ノームソート
- 基数ソート
- イントロソート

【Unity】ソートアルゴリズム12種を可視化してみた
https://qiita.com/r-ngtm/items/f4fa55c77459f63a5228

数多くのアルゴリズムが作られ、今もなお研究され続けている。

どれが一番いいの？

- 平均計算時間
- 最悪計算時間（計算量オーダー）
- メモリの使用量
- 安定性

データ量や、データの構造、時間的コストやメモリのコストなど、
それぞれのケースに合わせて、より良いアルゴリズムを選択する
ことになる。

基本技術者試験

- 2023 年 4 月から通年試験化
- 科目 B 試験は**アルゴリズムとプログラミング**が重要視される。
- 今まで選択式だったプログラミング言語が**擬似言語**に統一される。

サンプル問題

https://www.ipa.go.jp/shiken/syllabus/henkou/2022/ssf7ph000000h5tb-att/fe_kamoku_b_set_sample_qs.pdf

最後に

アルゴリズム的な考え方は
日常生活や仕事でも役に立つ

- 目的を達するまでの道筋が明確になる
- スムーズに作業を進めるためのコツがわかる
- 「計画性」や「効率性」を身につけることができる