

● プログラミング言語の種類

コンピュータが生まれたきっかけ

その頃の計算方法＝配線をつなぐ、入れ替える

コンピュータが分かる言語＝機械語

機械語と 1 : 1 で命令を書く＝アセンブラ言語

その後、「より人間の言語に近い」プログラミング言語が生まれてくる

FORTRAN、COBOL、BASIC、ALGOL、RPG、LISP、PL/I…

GOTO 文有害論

Pascal、C (C++、C#、Objective-C…), Parl

Python、Visual Basic、Ruby、Java、JavaScript、PHP、Scratch、Swift…

インタプリタ型、コンパイラ型

● アルゴリズム

アルゴリズムは「問題解決のための手順・手法」のこと。



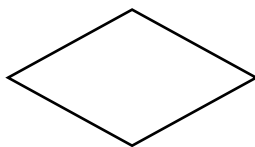
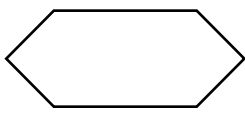
プログラミング言語に依存するものではない（どの言語でも基本的には有効・出来ることがそれぞれの言語で違うので、そのあたりの細かな違いはある）

よって、はじめのうちは「どう解決するのか、紙などを書いてみる」ことを薦める

※流れ図（フローチャート）について（詳しくは第 9 回講義にて）

流れ図とは、アルゴリズムやプロセスを表現するもの。

基本図形の一部の例

	開始、終了など		処理 (広く使われる)
	比較・判断・選択		準備

これを線や矢印で繋いで、「コンピュータにどう処理させるか」を描いていく。

当然ながらであるが、前回「プログラミングとは」で話したような「プログラミング的思考・論理的思考」が加わってくる。

アルゴリズムの作り方を挙げる。

まずは

- ・目的は何か
- ・現状はどうなっているのか
- ・条件があるか

を考えることがとても大事になる。

例：

チョコレートが食べたい（目的）

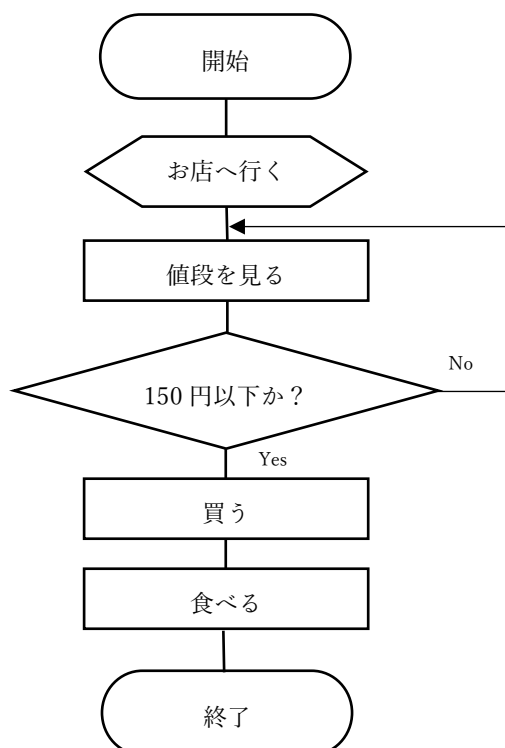
買いに行く時間はある（現状）

後のことを考えると、使えるお金は 150 円（条件）

人間は優秀なので、自然と以下の行動を行う。

- ① 近くのお店へ行く、チョコレートが置いてあるお菓子コーナーへ向かう
- ② チョコレートの値段を見る
- ③ 150 円以下かどうか確認する
- ④ 買える値段であればレジに持って行って買う
- ⑤ おうちに帰って食べる

これを流れ図（フローチャート）で書くと、このような感じになる。



作業分解して
一つ一つ書き出そう！



※プログラミング的思考・論理的思考（ロジカルシンキング）について
論理的思考とは？

物事を整理して、筋道を立てて、矛盾が無いように考えること

必要要素は？

作業や話を進めるうえで筋道が通っている

偏って考えない

物事や作業を適切に分解する（作業分解）

コンピュータは1つ1つのこと（処理・命令）は高速に出来ますが、
組み合わせて考えることは苦手…

人間が教えてあげること→手順（アルゴリズム）を考え、それをプログラミング言語を使って
教えてあげる（プログラミングする）

● Scratch について

プログラミング言語＝コンピュータに指示をする言語

本来は「命令の種類、そのプログラミング言語の文法」を覚える必要がある。

この文法を覚えることを極力省き、論理的思考をし易くした言語が Scratch

公式サイト：<https://scratch.mit.edu/>