

選択構造・反復構造と比較・論理演算 infomaticsI-025

教科書 p.169, 170

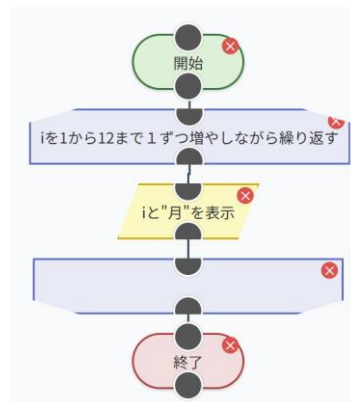
選択構造・反復構造と比較・論理演算

- ・ 選択構造
 - 条件によって処理を分ける構造
 - 「もし〜ならば」「そうでなければ」のように判断して処理が変わる
 - 1 つめの条件から順に評価され、条件が真 (True または 1) であればそのブロックだけが実行される
- ・ 反復構造 1 (条件での反復)
 - 指定した条件が真 (True または 1) であれば処理を繰り返す構造
 - 変数の値を変えながら繰り返すことが多い
- ・ 反復構造 2 (回数での反復)
 - 指定した変数が指定したとおりに変わる回数だけ繰り返す
- ・ 比較演算
 - 値どうしを比べる演算 (等号と不等号)
 - 共通テスト用プログラム表記・Python
 - `a == b` # a と b の値が等しいなら True, 等しくないなら False で置換
 - `a != b` # a と b の値が等しくないなら True, 等しいなら False で置換
 - `a > b` # a が b より大きいなら True, a が b 以下なら False で置換
 - `a < b` # a が b より小さいなら True, a が b 以上なら False で置換
 - `a >= b` # a が b 以上なら True, a が b より小さいなら False で置換
 - `a <= b` # a が b 以下なら True, a が b より大きいなら False で置換
- ・ 論理演算
 - 条件式を組み合わせるときに使う
 - True を 1, False を 0 として, 論理演算をする
 - 共通テスト用プログラム表記・Python
 - (比較演算 a) and (比較演算 b) # 比較演算 a と比較演算 b の論理積
 - (比較演算 a) or (比較演算 b) # 比較演算 a と比較演算 b の論理和
 - not(比較演算 a) # 比較演算 a の否定

基礎練習

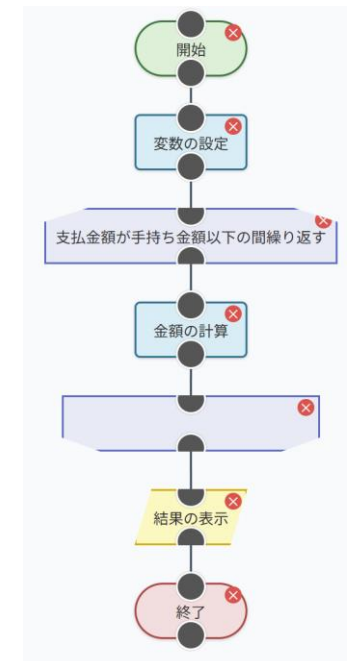
1. 1 月から 12 月までを表示するプログラムは以下のようになる。

- 共通テスト用プログラム表記
 - i を 1 から 12 まで 1 ずつ増やしながら繰り返す:
 - ↳ 表示する(i, "月")
- Python
 - `for i in range(1, 13, 1):`
`print(i, "月")`
- 実行結果
 - 略



2. 手持ちのお金が 2000 円の時, ビッグマックを買える個数を表示するプログラムは以下のようになる。

- 共通テスト用プログラム表記
 - `kingaku = 0`
`temochi = 2000`
`kosuu = 0`
`bigmac = 480`
`kingaku + bigmac <= temochi` の間繰り返す:
 - ↳ `kingaku = kingaku + bigmac`
 - ↳ `kosuu = kosuu + 1`
 - 表示する("ビッグマックを", `kosuu`, "個購入できます")
- Python
 - `kingaku = 0`
`temochi = 2000`
`kosuu = 0`
`bigmac = 480`
`while kingaku + bigmac <= temochi:`
`kingaku = kingaku + bigmac`
`kosuu = kosuu + 1`
`print("ビッグマックを", kosuu, "個購入できます")`



- 実行結果
 - ビッグマックを 4 個購入できます

3. 今日が 28 日または日曜日なら「休み!」, 天気が雨なら「バス投稿」, 雪なら「休みかも!」, それ以外は「自転車登校」と表示するプログラムは以下のようになる。

- 共通テスト用プログラム表記
 - `tenki = "晴"`
`hiduke = 17`
`youbi = "月"`
もし (`hiduke == 28`) or (`youbi == "日"`) ならば:
 - ↳ 表示する("休み!")
 - そうでなくもし `tenki == "雨"` ならば:
 - ↳ 表示する("バス登校")
 - そうでなくもし `tenki == "雪"` ならば:
 - ↳ 表示する("休みかも!")
 - そうでなければ:
 - ↳ 表示する("自転車登校")

- Python
 - `tenki = "晴"`
`hiduke = 17`
`youbi = "月"`
`if (hiduke == 28) or (youbi == "日"):`
`print("休み!")`
`elif tenki == "雨":`
`print("バス登校する")`
`elif tenki == "雪":`
`print("休みかも!")`
`else:`
`print("自転車登校する")`
- 実行結果
 - 自転車登校する

