

プログラミングの解釈について

「テストが近いけれど提出も早いから友達の写しちゃった！で、これどういうこと？」といった言葉をなくすべく作ってみました。もちろん、突然の思い付きで。

一部、Scratch3.0(通称スクラッチ)のブロックを例として入れてあります。

- RPGについて

- プログラム(基本のみ)

(開発>挿入>フォームコントロール>ボタン(フォームコントロール))

Sub ボタン1_Click()

```
    Dim namee As String
    namee = InputBox("名前を入力して下さい")
    Dim toi1 As Long
    toi1 = InputBox("お城を出た。どちらに進む? (右: 1 左: 2) ")
    If toi1 = 1 Then
        MsgBox "敵が現れた。"
        Dim teki1 As Long
        teki1 = InputBox("どうする? (たたかう: 1 逃げる: 2) ")
        If teki1 = 1 Then
            MsgBox "たたかう武器を持っていない！"
            MsgBox namee & "は敵に倒された...ゲームオーバー！"
        Else
            MsgBox "うまく逃げ切ることができた。"
        End If
    Else
        MsgBox namee & "は武器屋にたどり着いた。"
    End If
End Sub
```

- Sub の名前、なにこれ？

►先生の指示により名前を変えなかったことによるものです。

ボタン1だなんて面白みのない名前をしていますが、それこそ青や赤になるよう指定したり、他にも様々な工夫をしたりすることでクイズの早押しボタンのように使うことなんかもできるでしょう。ですが、今

回はスタートで「▶」を押さなくていいようにするためだけに使ったので、名前もそっけなくていいのです。



左：本来のスタート方法 右：今回のスタート方法

- ・「As ○○」って？

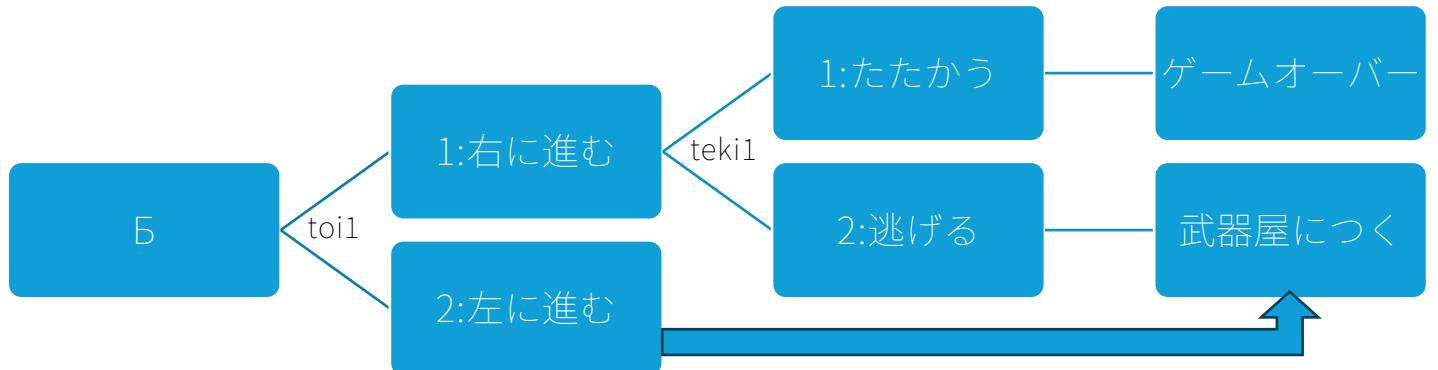
►データ型と呼ばれるもので、どんな使い方をするかを指定します。省略してもよく、その場合は全部入りを指定したことになります。

As Long は長整数型と呼ばれ、約±20 億の範囲の整数を持っておくことができます。toi1 は選択肢が 2 個しかありませんが、大体あと 39 億 9999 万 9998 個ぐらい選択肢を増やしても平気ということです。

As String は文字列型と呼ばれ、20 億文字までの文字を持っておくことができます。これならやたら単語の長い、ドイツ圏の人も安心ですね！

- ・色々並んでてよくわからない…

►図説します。以下の通りになります。先頭のゴチャゴチャは文字”Б”としてまとめました。



- ・素数判定について

- ・プログラム

```
Sub 素数を探せ()
    Dim i, kazu, sosu
    kazu = InputBox("数字を入力してください")
    sosu = 1
    i = 2
    Do While i <= kazu - 1
        If kazu Mod i = 0 Then
            sosu = 0
        End If
        i = i + 1
    Loop
    If sosu = 1 And kazu >> 1 Then
        MsgBox "素数です"
    Else
        MsgBox "素数ではありません"
    End If
End Sub
```

割る数が知りたい数より 1 小さい数になるまで繰り返す
Kazu を i で割ったあまりが 0 のとき
→sosu を 0 にする
カウントを 1 進める
繰り返し

もし「sosu が 0」かつ「kazu が 1 ではない」とき
→素数だと表示する
ではないとき
→素数ではないと表示する
If を終了する

- ・1 は素数じゃないです。どうして sosu=1 にしているの?

►素数かそうでないかの区別のために用いられています。A と B にしても問題はありません。

- ・i = 2 始まりなのはどうして? kazu-1 なのはどうして?

►知りたい値 ÷ 1 と、知りたい値 ÷ 知りたい値を計算することは意味がないからです。

素数の常識に従えば、素数はどんな数字でも割れない数ではなく、1 とその数自身でのみ割れる数です。

しかし、素数でない数字も 1 とその数自身で割ることができるから、意味がないのです。

- ・Mod とは?

►割り算のあまりを出してくれる関数。

ゲームの改造/拡張コンテンツのことではありません。「元の数 Mod 割る数」のように使います。If x Mod y = z Then とすれば、「x を y で割った余りが z の時」という使い方ができます。

を で割った余り

- ・<>って何!? (((; >Δ<))))が カクブル

►演算子というやつの仲間で、≠の意味を持ちます。

顔文字でなければ、電車のパンタグラフでもありません。A と B が等しくないとき、A >> B と表します。じゃあ「≠」でいいじゃないかと思うかもしれません、それではいけません。詳細は差し控えますが、絵

文字と同じような、スペシャルな文字に分類される事が多いので、古いパソコンなんかでは使えませんし。なお、=の仲間を比較演算子といいます。

演算子	+	-	*	/	MOD
普段	+	-	×	÷	なし
意味	加算	減算	乗算	除算	除算のあまり
<hr/>					
演算子	m^n	$m < n$	$m \leq n$	$m > n$	$m \geq n$
普段	m^n	$m < n$	$m \leq n$	$m > n$	$m \geq n$
意味	m の n 乗	m が n より小さい	m が n 以下	m が n より大きい	m が n 以上
					m と n が等しくない

※もちろん実際は m と n のペアではないことが多い。



- Int とは?
- 小数点以下を切り捨てる関数。
Int (5.5)なら 5 という風になります。値をそのまま括弧に入れることはあまりなく、割り算などの計算結果の小数を無視するときなどに使いますが、四捨五入とは異なります。

- ・斜方投射について

- ・プログラム

```
Sub 斜方投射()
```

```
    Dim x, y, vx, vy, t
```

```
    x = 0
```

```
    y = 0
```

最初の座標 グラフで言うなら「(0,0)スタートにします」

```
    vx = 5
```

```
    vy = 10
```

```
    t = 0
```

```
    Sheet1.Cells.ClearContents
```

```
    Cells(1, 1).Value = "x"
```

```
    Cells(1, 2).Value = "y"
```

```
    Do While y >= 0
```

```
        Cells(2 + t, 1).Value = x
```

```
        Cells(2 + t, 2).Value = y
```

```
        x = x + vx
```

```
        y = y + vy
```

```
        vy = vy - 1
```

```
        t = t + 1
```

```
    Loop
```

```
End Sub
```

投げた時点での右へ進む速さ

　　〃 上へ進む速さ

時間のカウント用

Excel の Sheet1 の中身を初期化する

セル A1 に x という文字を代入

セル B1 に y という文字を代入

y が 0 になるまで繰り返す→投げた球が地面につくまで

セル A(2+t)にその時点での x の値を代入

セル B(2+t)にその時点での y の値を代入

1 秒後の x の値 ←プリントでは、1 秒後のボールの座標を

1 秒後の y の値 ($x + V_x, y + V_y$) としているため。

毎秒 1 減速する上向きの速さ

時間カウント

繰り返し

- ・Cells(1, 2)とか、Cells(2+t,2)って何?

►VBA と Excel の連携に関するもの。

VBA は Excel と連携して様々なことができますが、斜方投射のプログラムでは結果を Excel に書かせる機能を使います。一般的な座標は(横,縦)の順ですが、VBA では(縦,横)の順となっているので注意が必要です。(1, 2)の場合、セル B1 のことを指します。(2+t,2)の場合、B1 に入れた y とぶつからないようその次から記録をできるようにしてあります。最初は t=0 なので、2+0=2、つまりセル B2 から記録が開始されます。

- ・どうして上方向のスピードは下がって、なんならマイナスに突入するのに、右方向はそういうのがないの?

►おそらくこれがシミュレーションの一例だからです。

空気抵抗はないが重力はある。そんな感じじゃないでしょうか。

- トランプ占い(基本)について

```
Sub トランプ占い()
    Dim i, card, score
    i = 1
    score = 0
    Do While i <= 3
        card = Rnd
        If card < 0.5 Then
            MsgBox "ハート"
            score = score + 1
        Else
            MsgBox "スペード"
        End If
        i = i + 1
    Loop
    MsgBox "得点は" & score "点です"
    If score = 0 Then
        MsgBox "凶"
    ElseIf score = 1 Then
        MsgBox "小吉"
    ElseIf score = 2 Then
        MsgBox "中吉"
    Else
        MsgBox "大吉"
    End If
End Sub
```

カウント用、スタート地点
スコア(なので最初に)

i が 3 になるまで繰り返す(i=1,2,3 の 3 回)
「カードは乱数(ランダム)です」
もしカード(乱数)が 0.5 未満なら、
ハートと表示して、スコアを 1 点加算する
(カードが 0.5 未満)じゃなかったら
そのまま If を終了

カウントする(1→2,2→3)
くりかえし
(上のを 3 回やったら)そのスコアを表示する

もしスコアが 0 点だったら 凶と表示する
(0点) じゃなくて 1 点なら 小吉と表示する
(1点) じゃなくて 2 点なら 中吉と表示する
どれでもなければ(自動的に 3 点だとわかるので)
3 点と表示する
If を終了
プログラムを終了

- どうして「カード(乱数)が”0.5”未満の時」なの?

►Rnd 関数の仕様を考慮しているからです。

ひたすらに乱数を出す、Rnd 関数は「0 以上 1 未満の数」を出してきます。そのため、そのまんなかである 0.5 が使われるのです。一方、待ち行列ではランダムな数をそのまま見せたいので、整数になるような工夫が必要になってきます。ここでは乱数が処理の過程に過ぎないので、そんな段階を踏む必要なんてありません。

- Elseif とは?Else との違いは?

►Elseif とは、前の If の条件を満たさなかったものから、別の条件のものを取り出すことです。

①カレーライスが食べたい人…最初の If ②カレーライスではなくハヤシライスが食べたい人…Elseif ③今の 2 つではなくシチューライスが食べたい人…Elseif ④この中に食べたいものがいる人…Else

単純に、Else はあまりものと考えてもいいでしょう。



左：単純な If の例 中：Else を使った If の例 右：Elseif を使った If の例

- ・そんな難しい Elseif 使うんじゃなくて、ひたすら If 文作れば良くない?

►はい、おっしゃる通りです。

If score = 0 Then ~ End If → If score = 1 Then ~ End If …とやっていく方がどう考えても考え方は簡単です。が、箇条書きや数学の分数、部屋のケーブルたちがそうであるように、プログラミングの世界にも、できる限りシンプルなつくりのほうが良いという考えがあるのです。とはいえ作りなさいと言われてさつと作ったときにひたすら If 文が書いてあっても間違いにはならないと思います。多分。

流れ

- ① カード(乱数)が 0.5 未満ならハートと表示してスコアを 1 点加算して If を終了する
(0.5 未満)ではないならスペードと表示して If を終了する
- ② を合計 3 回行う
- ③ スコア(score)を表示する
- ④ スコアが 0 なら凶と表示する
0 ではなく 1 なら小吉と表示する
0 でも 1 でもなく 2 なら中吉と表示する
どれでもなければ 3 と表示する

- ・配列探索について

- ・プログラム

Sub 探索()

```

Dim i, n, j, k
n = Array(2, 5, 8, 9, 30, 45, 55, 102, 183, 209)
j = 55
i = 0

For i = 0 To 9 Step 1
    If j = n(i) Then
        k = i + 1
        MsgBox j & "は" & k & "番目です"
    End If
Next i
End Sub

```

または
Do While i = 9
If
End If
i = i + 1
Loop

なお、End If の前に Exit For を入れることで途中終了することができるが、ここでは入れない。

途中終了は、「テスト終わった人から○○していいよ～」というイメージ。

- ・“For i = 0 To 9 Step 1 → Next i” または “Do While i = 9 → Loop” となっているが、なぜ 10 じゃなくて 9 なの?

► n(i)を使っているからです。つまり、

× 「もし j がリスト n の中の i 番目の数と同じだったら」

○ 「もし j がリスト n の中の、位置 i にいる数と同じだったら」ということです。

位置 n(?)	0	1	2	3	4	5	6	7	8	9
何番目	1	2	3	4	5	6	7	8	9	10
配列	2	5	8	9	30	45	55	102	183	209

同じ数字であれば適切な動作を、同じ数字じゃなかったら次の位置にいる数字を確認します。

- ・どうして k = i + 1 にするの?

► 「どこにいるか」よりも「何番目にいるのか」のほうがわかりやすいからです。

配列(リスト)や、「番目」という概念が数字を使っているので、もし位置を A,B,C,D,…と表していたらわかりやすいですが、Excel の世界では 0,1,2,3,…となっています。つまり、そのままだと「0 番目」ができるおかしいので、i に 1 を足すことで正しく「何番目」なのかを判断することができるのです。

- ・そもそも For Next と Do Loop は何が違うの?

► 「(中身)を n 回繰り返す」のが For Next で、「条件 n を満たす間ずっと(中身)を繰り返す」のが Do Loop です。

Do While i = n ~ Loop の場合、どうにかしてカウントを n にして上げる必要があります。



左：For Next の例 右：Do Loop の例

・流れ

探す数字:55 初期値: i=0

- ① リスト n から i(ここでは 0)の位置の数字を見つける
- ② $55 \neq 2$
- ③ 違ったので素通りして i に 1 を足す
- ④ リスト n から i(ここでは 1)の位置にある数字を見つける
- ⑤ $55 \neq 5$
- ⑥ 違ったので素通りして i に 1 を足す
- ⑦ ④⑤⑥をくりかえす
- ⑧ リストの n から i(ここでは 6)の位置にある数字を見つける
- ⑨ $55=55$
- ⑩ 一致したので If の中へ入り、そこまでの i に 1 を足して「k(番目)」とする。
- ⑪ 「j(探している数字)は k 番目です」と表示させる
- ⑫ If 終了
- ⑬ リストの n から i(ここでは 7)の位置にある数字を見つける
- ⑭ $55 \neq 102$
- ⑮ 違ったので素通りして i に 1 を足す
- ⑯ i が 9 までで終了し、i=10 はやらない。
※i = 10 までいくとリストの中に存在しない数を調べることになってしまう

・質問は鴨下まで。