

プログラミング概論

第4回 2024年10月16日

プログラミングの基本概念(2)

条件分岐

始める前に：

- ①ノートPCの電源を入れてください
- ②Wi-Fi等でインターネットに接続してください
- ③プログラミング概論のwebページを開いておいてください

今回の授業内容

- 復習（変数）
- 条件分岐とは
- 迷路パズルでは
- プログラムを作ってみる

復習

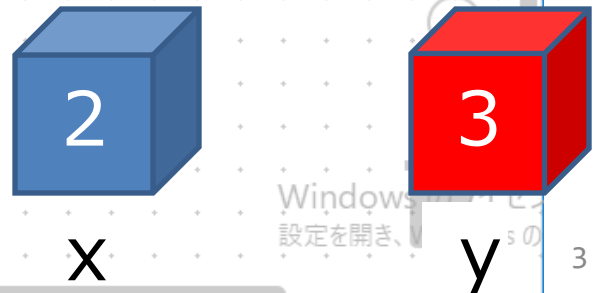
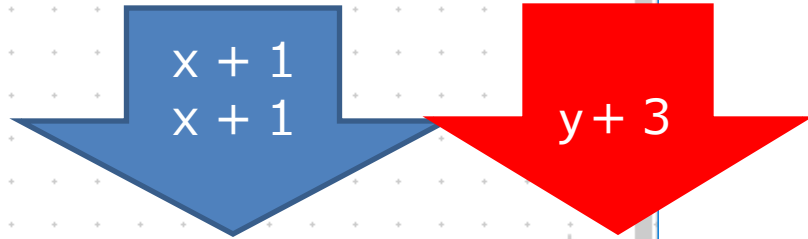
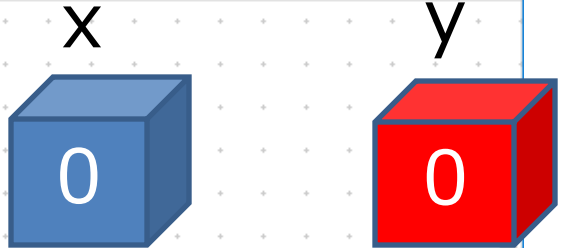
使う変数をあらかじめ作っておく
(Create variable...)

Set 変数 to 式 (数値)

setで
へ代入

print
で表示

```
set x to 0
set y to 0
print create text with x, y
set x to x + 1
set x to x + 1
set y to y + 3
print create text with x, y
```



ちなみにPythonだと



The screenshot shows a web browser window titled "Blockly Demo: Code" with the URL "blockly-demo.appspot.com/static/demos/code/index.html". The page displays a navigation menu with tabs for "Blocks", "JavaScript", "Python", "PHP", "Lua", "Dart", "XML", and "JSON". The "Python" tab is selected and circled in red. Below the tabs, the Python code is displayed in a text area:

```
x = None
y = None

x = 0
y = 0
print("join([str(x2) for x2 in [x, ' ', y]]))
x = x + 1
x = x + 1
y = y + 3
print("join([str(x3) for x3 in [x, ' ', y]]))
```

The code is color-coded, and a red box highlights the final three lines: `x = x + 1`, `x = x + 1`, `y = y + 3`, and `print("join([str(x3) for x3 in [x, ' ', y]]))`. The browser's address bar and various icons are visible at the top.

条件分岐とは



条件分岐

- プログラムは“前もって”書いておくので使われる時どのような状況かわかりません
 - 条件によって動作を変えると様々な状況に対応できます
 - 月曜日なら...
 - 変数aが正の数なら...
 - 左に曲がれるなら...
- のように“〇〇なら”の条件により〇〇する、と書くのが条件分岐
- 例：雨が降っているなら傘を持っていく

「Blockly Games: 迷路」を クリック



プログラミング概論 - MATSUMOTO x +

保護されていない通信 | daisy.fwex.tohoku-gakuin.ac.jp/proginthro.html

Gmail YouTube マップ プログラミング概論 -...

東北学院大学 教養学部 情報科学科 松本章代 研究室

MATSUMOTO Akiyo Laboratory

Tohoku Gakuin University

ホーム 研究紹介 講義 研究室メンバー 業績リスト 備忘録 リンク集

プログラミング概論

授業に必要なソフトのインストール・設定

- [NotePad++](#)
- [BlueStacks 5](#)
- [Google Chrome](#)
- [Google アカウント](#)

プログラミングサイト

- [Blockly Games: 迷路](#)
- [Blockly Games: Pond Tutor](#)
- [Blockly Code](#)
- [ピクトプログラミング](#)
- [MIT App Inventor](#)

迷路の問題 6

Blockly Games : 迷路

6

10



▶ プログラムを実行

まっすぐ進む

左を向く ◻ ▾

右を向く ◻ ▾

「まで繰り返す」 ◻

実行

もし左に進めるなら ◻ ▾

実行

「まで繰り返す」 ◻

実行

- まっすぐ進む
- もし左に進めるなら ◻ ▾
- 実行
- 左を向く ◻ ▾



プログラムの基本：条件（分岐）と繰り返し

繰り返し（「～まで繰り返す」）

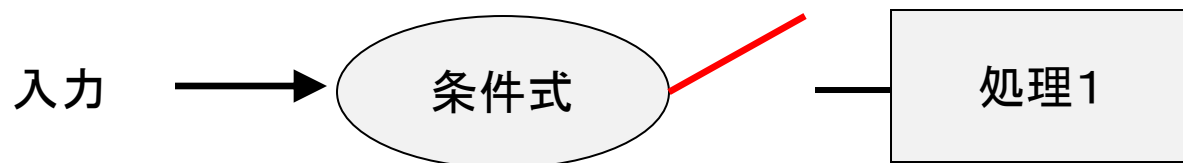


条件(分岐)（「もし～なら・・・する」）

条件分岐

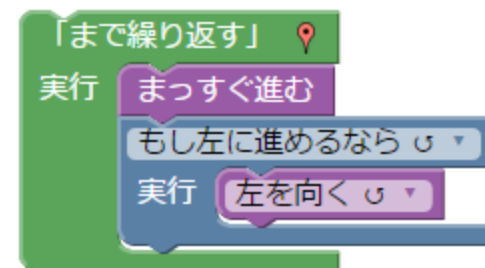
もし・・・であれば処理1を行う

・・・には条件(式)がはいる



例：迷路問題6

もし左に進めるならば左を向く



Blockly Codeで条件分岐は？



ブラウザのタブ: ブロックリーゲーム: 迷路, プログラミング概論 - MATSUMOTO

アドレスバー: 保護されていない通信 | daisy.fwex.tohoku-gakuin.ac.jp/progintro.html

プログラミング概論 -...

東北学院大学 教養学部 情報科学科 松本章代 研究室

MATSUMOTO Akiyo Laboratory

Tohoku Gakuin University

ホーム 研究紹介 講義 研究室メンバー 業績リスト 備忘録 リンク集

プログラミング概論

授業に必要なソフトのインストール・設定

- [NotePad++](#)
- [BlueStacks 5](#)
- [Google Chrome](#)
- [Google アカウント](#)

プログラミングサイト

- [Blockly Games: 迷路](#)
- [Blockly Games: Pond Tutor](#)
- [Blockly Code](#)
- [ピクトプログラミング](#)
- [MIT App Inventor](#)

配付資料

- [第1回講義資料 \(09/20\)](#)
- [第2回講義資料 \(09/27\)](#)

Blockly CodeではLogicを使う

条件分岐の基本 if文：もし (if) 条件式を満たすなら実行(do)

if 条件

do 条件を満たせば実行したいこと

else そうでなければ実行したいこと

The image shows a screenshot of the Blockly Code editor interface. On the left, there is a 'Blocks' palette with various categories: Logic, Loops, Math, Text, Lists, Colour, Variables, and Functions. The 'Logic' category is highlighted with a green arrow. Inside the 'Logic' category, the 'if do' block is circled in red. A red arrow points from this block to a larger, magnified view of the 'if do' block in the workspace. The magnified view shows the 'if do' block with a gear icon and a play button. Below it, there are three options: 'else if', 'else', and 'if'. The 'if' option is selected, and a callout box highlights the 'else if' and 'else' options.

条件式を入れるには

The image shows the Scratch block palette on the left, with a green arrow pointing to the 'Logic' category. The 'Logic' category is expanded, showing 'if do' blocks. A red arrow points from the 'if do' block in the palette to an 'if do' block in the workspace. Another red arrow points from the '=' block in the palette to the condition slot of the 'if do' block in the workspace. A third red arrow points from the condition slot to a dropdown menu showing various comparison operators: '=', '≠', '<', '≤', '>', and '≥'. The '=' operator is selected and highlighted in yellow.

Logic
Loops
Math
Text
Lists
Colour
Variables
Functions

if do

set x to 2

if do

and

if do

✓ =
≠
<
≤
>
≥

中央の部分をクリックすると
プルダウンメニューで等号、
不等号などが選択できる

条件分岐の例

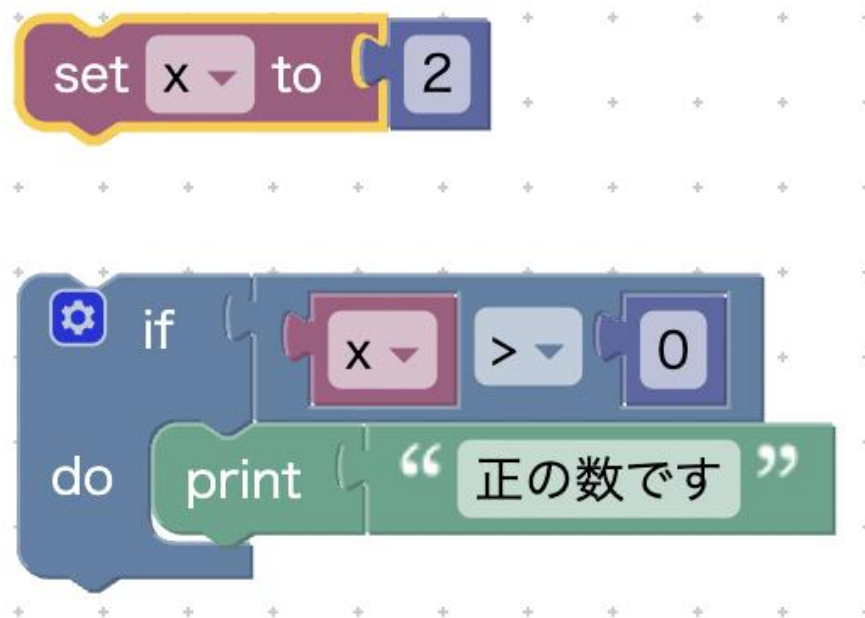
- ①変数 x に 2 を代入
- ②変数 x が正の値なら → “正の数です” と表示

プログラムの流れ

- x の値が 2 とします (例)



- x の値が正の値なら“正の数です”と表示する



演習 ブロックを組んで実行させて下さい

①変数 x に 2 を代入

②変数 x が正の値なら → “正の数です” と表示

Hint: ブロックの種類について（画面左側のメニュー）

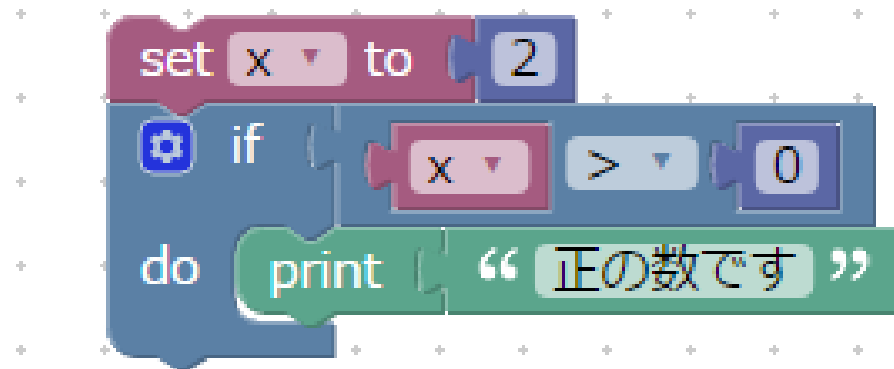
- ・ 変数のブロックを使うときはVariables
- ・ 数字や計算のブロックを使うときはmath
- ・ 条件分岐をさせるブロックを使うときはLogic
- ・ 表示させるブロックを使うときはtext

プログラミング時の注意

- 数字、アルファベットは“半角”
 - プログラミングでは半角が基本
 - printで表示する部分は全角でもよい
 - 半角でないと数字として入力できない
 - 全角とまぜると条件分岐で間違えたりする

完成

- ①変数 x に 2 を代入
- ②変数 x が正の値なら → “正の数です” と表示



数字をキーボードから入力するには

- ・変数 x の値をキーボードから入力する



プログラムで書いてみよう

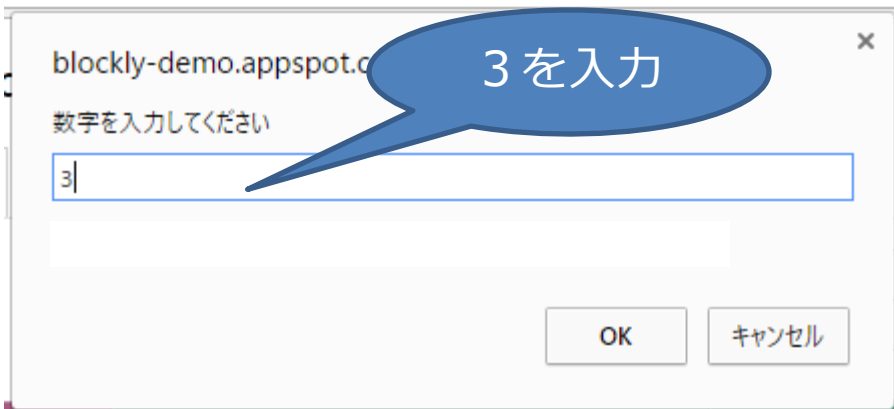
1. 前のプログラムで x に2を代入する部分の数値の部分を削除
2. set x toがもし x でなければ x に変更
3. Textから「prompt for text ▼ with message “ ”」を選択
4. 「set x ▼ to □」の□につなげる
5. textをnumberに変更
6. textから「 “ ” 」ブロックを選んでこのブロックにつなげ “ ” に「数字を入力してください」

set x ▼ to prompt for number ▼ with message “ 数字を入力してください ”

完成

- ①変数 x にキーボードから数値を代入
- ②変数 x が正の値なら → “正の数です” と表示

```
set x to prompt for number with message “数字を入力してください”
if x > 0
do print “正の数です”
```



迷路の問題 9

Blockly Games : 迷路

9

10



▶ プログラムを実行

まっすぐ進む

左を向く ↶

右を向く ↷

「まで繰り返す」

実行

もしまっすぐ進めるなら

実行

もしまっすぐ進めるなら

実行

それ以外

「まで繰り返す」

実行

もしまっすぐ進めるなら

実行

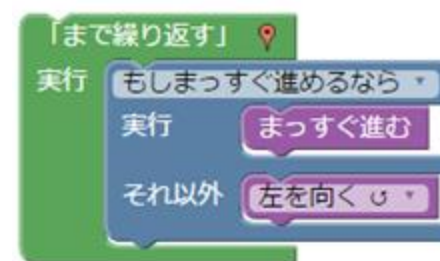
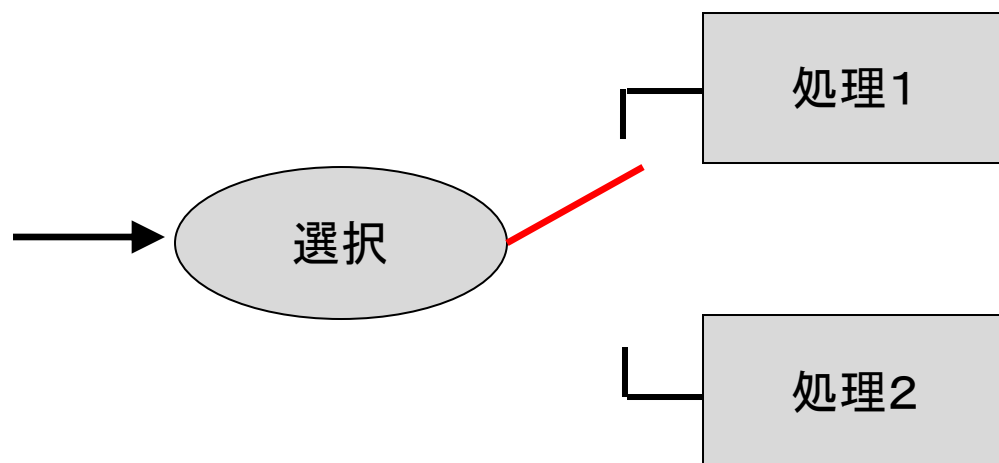
まっすぐ進む

それ以外

左を向く ↶

条件分岐 2

もし・・・であれば処理1を、そうでなければ処理2を行う



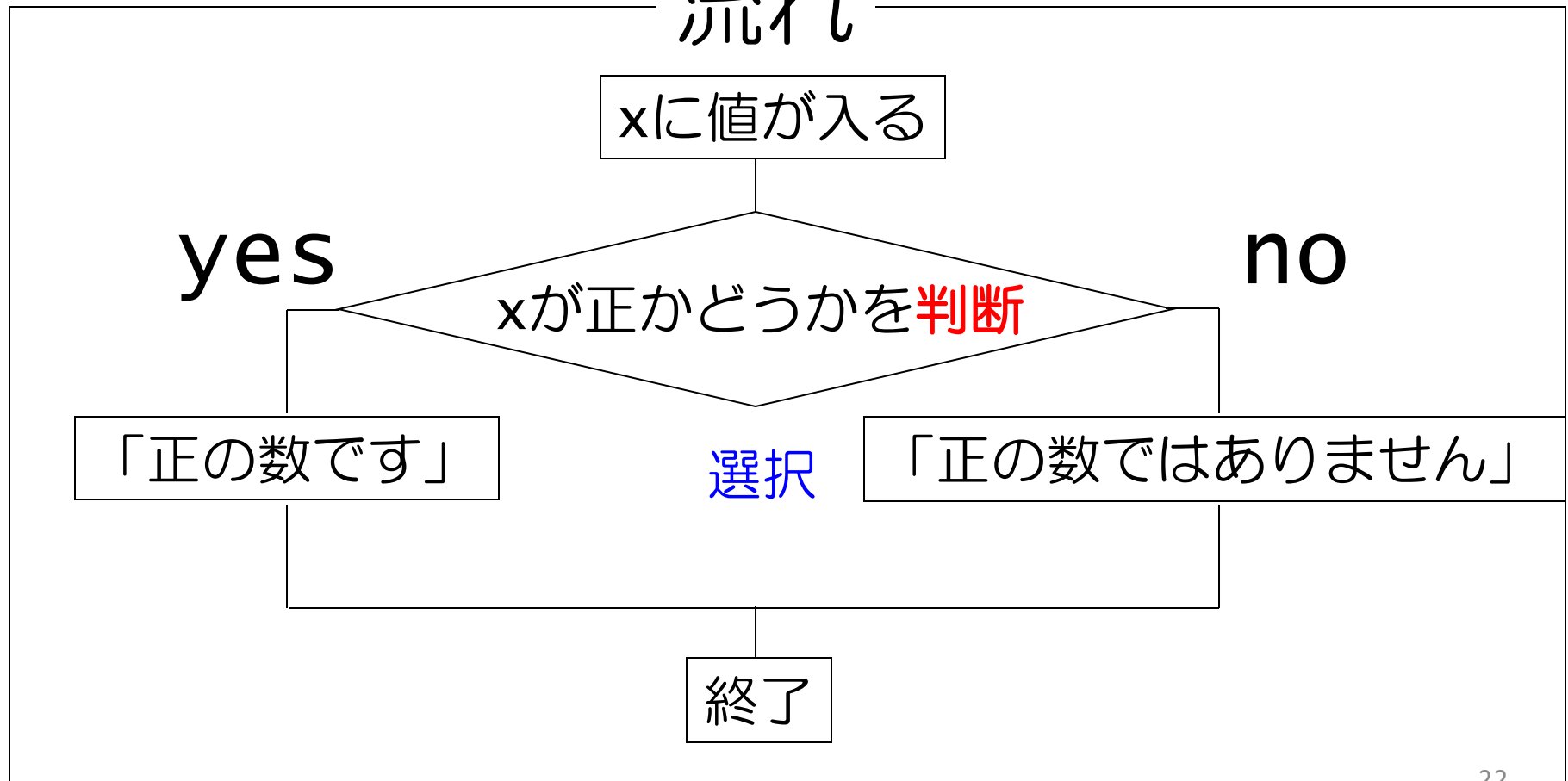
例：迷路問題 9

もしまっすぐ進めるならばまっすぐ進む、
そうでなければ(それ以外では) 左を向く

条件分岐 もし…の場合…そうでない場合…

- xが正なら → 「正の数です」と表示
- そうでないときは → 「正の数ではありません」と表示

流れ

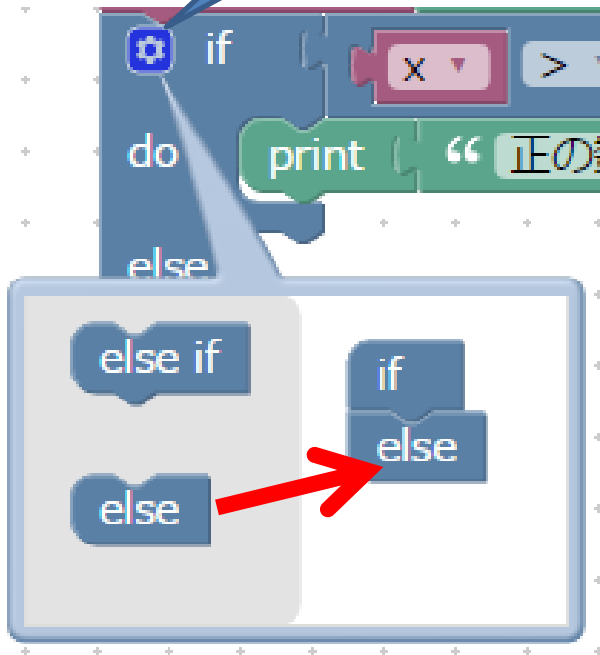


プログラムで書くと「そうでないときは」 else

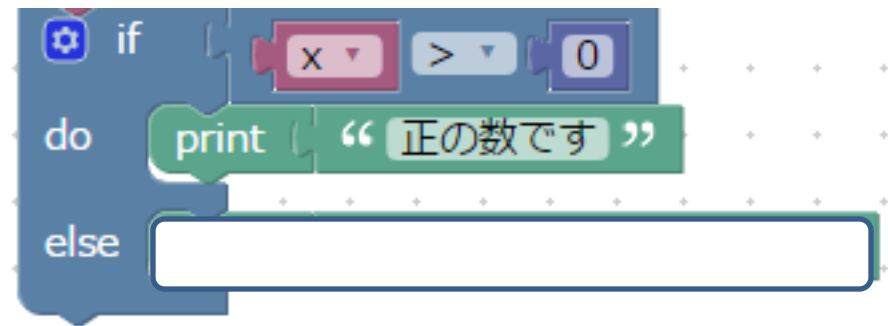
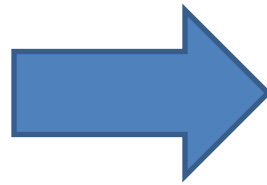
- xが正なら → 「正の数です」と表示
- そうでないときは → 「正の数ではありません」と表示

プログラム

設定ボタン
をクリック



elseをifの下につなげる



演習：先ほどのプログラムを書き換えて下さい

プログラムの完成例と実行画面

```
set x to prompt for number with message "数字を入力してください"  
if x > 0  
do print "正の数です"  
else print "正の数ではありません"
```

便利な機能：同じようなブロックを作る場合
コピーしたいブロックの上で右クリック
⇒メニューからDuplicate（複写）を選択
またはCTRL+CとCTRL+Vで同じ操作ができる

blockly-demo.appspot.com の内容:

数字を入力してください

0

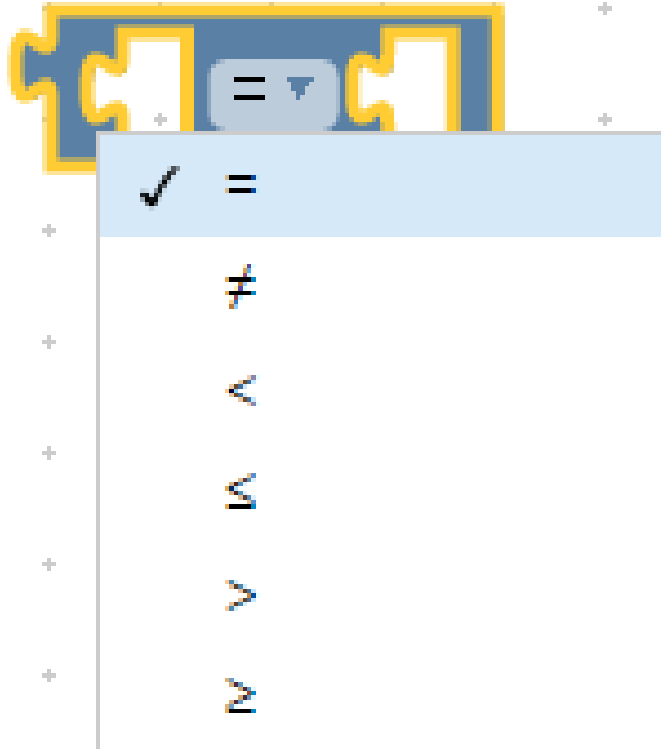
0を入力

blockly-demo.appspot.com の内容:

正の数ではありません

「正の数ではありません」の表示

=以外の記号も覚えよう



記号	説明
=	等しい
≠	等しくない
<	小さい
≤	小さいか等しい
>	大きい
≥	大きいか等しい

プログラムを作ってみよう

問題1

• 「割られる数を入力してください」、「割る数を入力してください」の表示とともに、数字をふたつキーボードから入力して、それぞれ変数a, bに代入する。aがbで割り切れれば、「aはbで割り切れます」と表示し、割り切れなければ「aはbで割り切れません」と表示するプログラムを作成してください。

(注意：a, bの部分は実際には入力した数を表示させること)

ヒント

- 変数はa, b, cの3つをcreate variable...から作ってください
- 割ったときのあまりの計算は「Math」のなかから remainder of を選択して計算させよう
- 割ったときのあまりを変数cにいれ、その値が0かそうでないかで条件分岐させよう
- 表示にはcreate text withとa, bを用いて表示させよう

結果例

入力 1

blockly-demo.appspot.com の内容

割られる数を入力してください

OK

キャンセル

入力 2

blockly-demo.appspot.com の内容

割る数を入力してください

OK

キャンセル

出力

blockly-demo.appspot.com の内容

6は3で割り切れます

OK

迷路問題 9 再び (別解答?)

Bloky Games : 迷路

9 10



▶ プログラムを実行

まっすぐ進む

左を向く

右を向く

「まで繰り返す」

実行

もしまっすぐ進めるなら

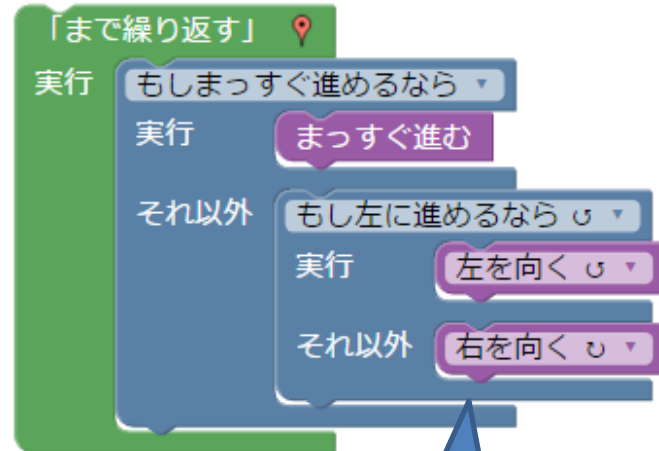
実行

もしまっすぐ進めるなら

実行

それ以外

条件分岐の中にさらに条件分岐が入っている！ = 多方向分岐



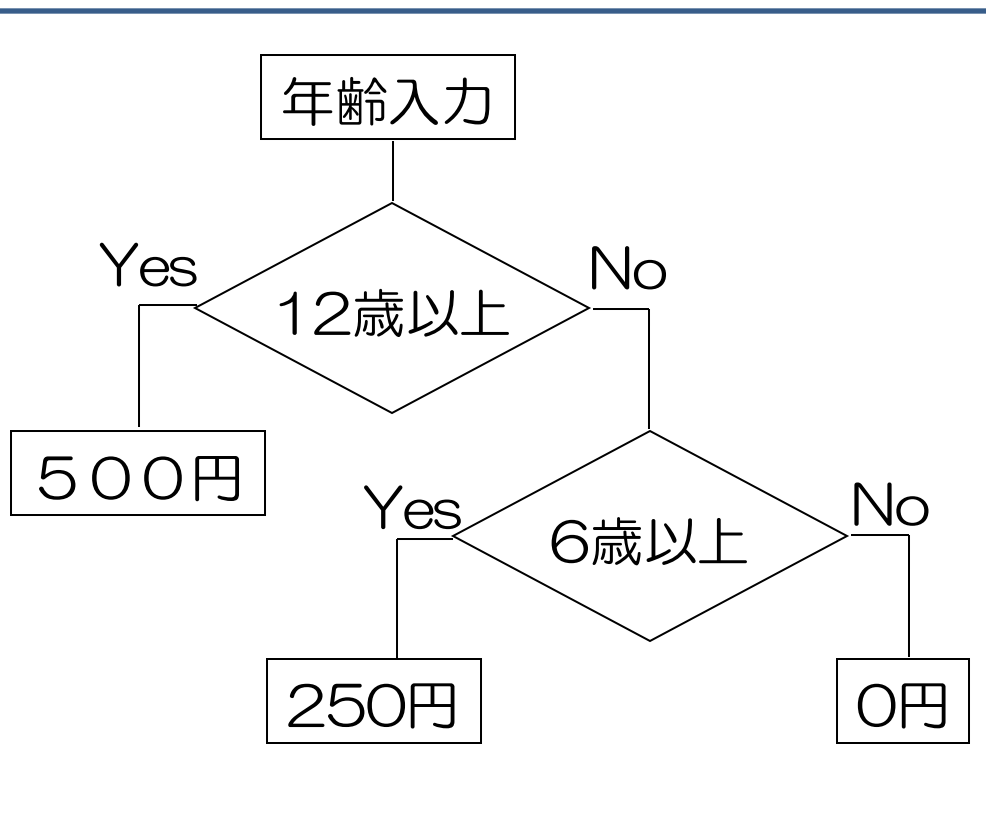
「まっすぐ進める場合」
以外でさらに
「左に進める場合」
に左を向く

else if で多方向分岐する

年齢（変数名 age）を入力させる

- ◆ 年齢が12歳以上：「500円」
- ◆ 年齢が6歳以上12歳未満：「250円」
- ◆ 年齢が6歳未満：「0円」

12歳以上
でなくさらに
もし6歳以上なら
250円



elseif...
そうではな
くさらに
もし...なら

if

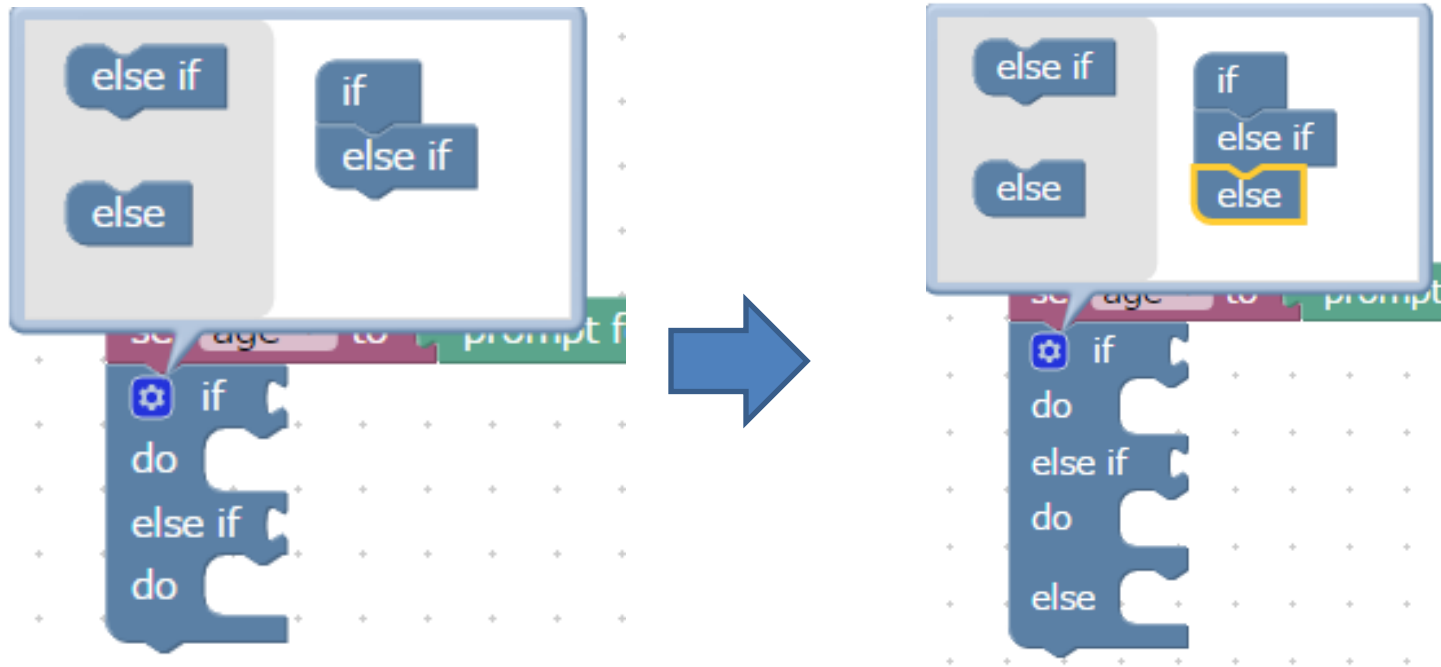
else
if

プログラムでは

0. 年齢の変数ageをcreate variable...から作成する
1. 年齢を入力させ、変数ageに代入する

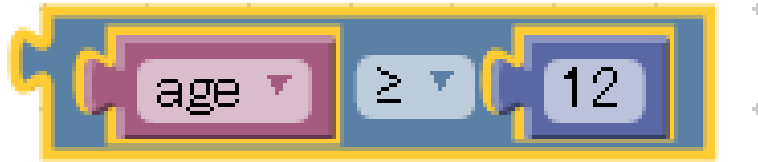
set age to prompt for number with message “年齢を入力してください”

2. Logicからifを選択
3. ifに「else ifを追加」し、その下に「else」も追加



プログラムでは

4. ifのところに条件（12歳以上なら）を追加



5. doのところで変数xに500を代入



6. else ifのところに条件（6歳以上なら）を追加

7. doのところで変数xに250を代入

8. elseのところで変数xに0を代入

9. printでxの値を出力

プログラムを作ってみよう

大人：18歳以上 / 中人（中学・高校生）：12～17歳 / 小人（幼児・小学生）：4～11歳 / シニア：65歳以上 ※3歳以下は無料

	パークチケット名称	大人	中人	小人	シニア	オンライン購入
	 1デーパスポート	¥7,400	¥6,400	¥4,800	※シニアパスポートをご利用ください。	 オンラインで購入

※ ちょっと古いですが...

問題2

- 年齢(変数age)を入力して、チケットの金額(変数y)を表示するプログラムを作成してください。
- 設定 シニア(65～)：6700円、大人(18～64)：7400円、中人(12～17)：6400円、小人(4～11)：4800円、3歳以下は無料

結果例

入力例 1



出力

ページ <https://blockly-demo.appspot.com> の記述: ✕

年齢は?

OK キャンセル

blockly-demo.appspot.com の内容: ✕

4800

入力例 2

出力

ページ <https://blockly-demo.appspot.com> の記述: ✕

年齢は?

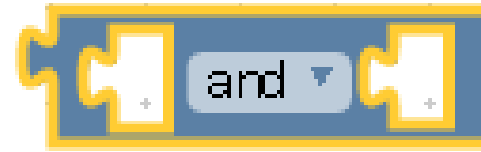
OK キャンセル

blockly-demo.appspot.com の内容: ✕

6700

論理演算子 1 : 「～かつ～の場合・・・」

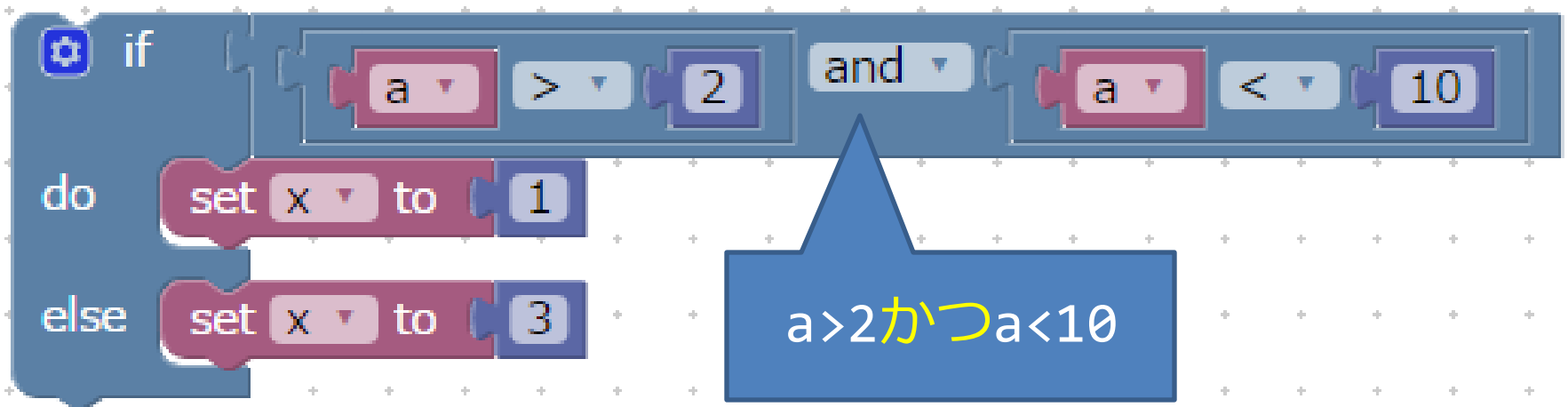
2つ以上の条件を満たす場合の処理



(例1)

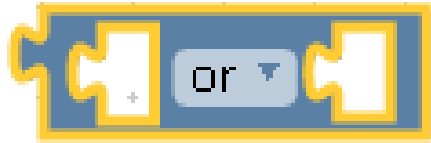
- aが2より大きくかつ10より小さいなら → xに1を代入
- そうでないときは → xに3を代入

プログラム例



論理演算子 2 : 「～または～の場合・・・」

複数条件のうちのひとつ以上を満たす場合の処理

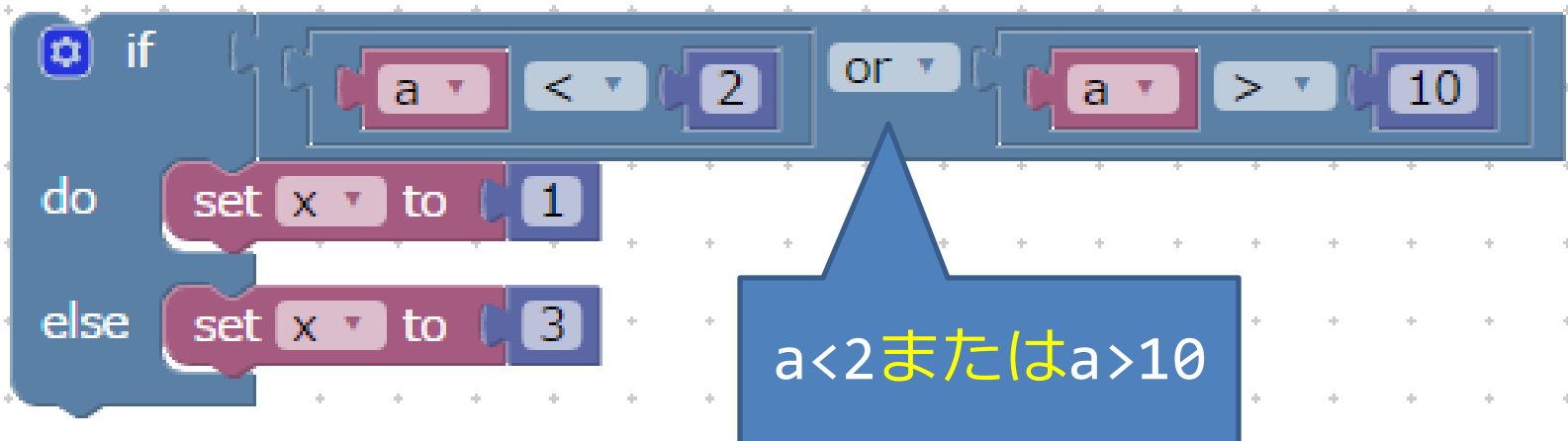


「and」のところを「or」に変更

(例2)

- aが2より小さい**または**10より大きいなら → xに1を代入
- そうでないときは → xに3を代入

プログラム



プログラムを作ってみよう

問題3

・「国語の点数」、「数学の点数」の表示とともに国語と数学の点数を入力させ、それを変数suとkoに代入する。数学の点数と国語の点数が両方とも60点以上なら「合格」と表示し、それ以外は「不合格」と表示するプログラムを作成してください。

Hint

- ・変数は数学の点数su, 国語の変数koの2つをcreate variable...から作ってください
- ・条件分岐にはif elseを用います
- ・国語が60点以上かつ数学が60点以上なら合格ですからif文の条件式にはその条件をあらわすように論理演算子を使ってください

結果例

入力 1

blockly-demo.appspot.com の内容

国語の点数

OK キャンセル

入力 2

blockly-demo.appspot.com の内容

数学の点数

OK キャンセル

出力

blockly-demo.appspot.com の内容

不合格

OK

結果例

入力 1

blockly-demo.appspot.com の内容

国語の点数

OK キャンセル

入力 2

blockly-demo.appspot.com の内容

数学の点数

OK キャンセル

出力

blockly-demo.appspot.com の内容

合格

OK

時間が余った人は

- 迷路問題をやってみよう
- 迷路問題が終わったら、Pondに挑戦してみよう
(途中から条件分岐を使います)

東北学院大学 教養学部 情報科学科 松本章代 研究室

MATSUMOTO Akiyo Laboratory

Tohoku Gakuin University

ホーム 研究紹介 講義 研究室メンバー 業績リスト 備忘録 リンク集

コンピュータと論理A

授業に必要なソフトのインストール・設定

- [NotePad++](#)
- [NoxPlayer](#)
- [Google Chrome](#)
- [Google アカウント](#)
- [BlueStacks](#)

プログラミングサイト

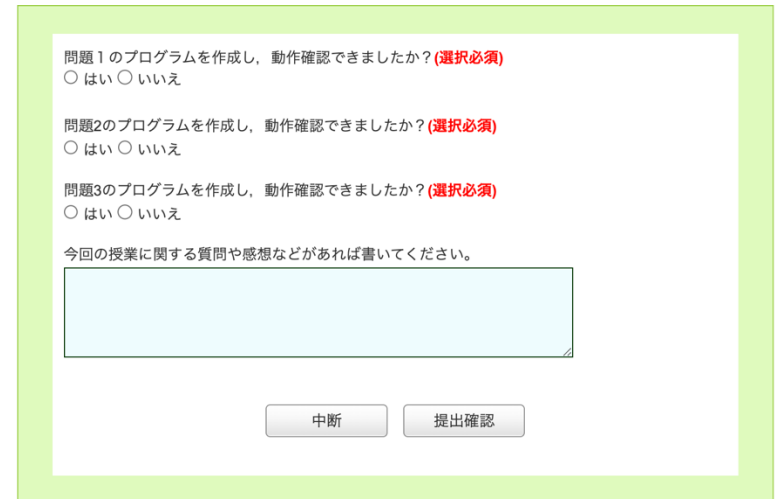
- [Blockly Games: 迷路](#)
- [Blockly Games: Pond Tutor](#)
- [Blockly_Code](#)

← Pondに挑戦するときはここをクリック

作業終了後（終了しなくても16:45まで）

manabaで振り返りを提出

- manabaにログイン
- プログラミング概論の小テストから第4回振り返り課題を選択し，スライドのプログラムを作成し動作確認できたかを回答する
- 感想や分からなかった点などもあれば記入



The screenshot shows a quiz submission interface with the following elements:

- Question 1: "問題1のプログラムを作成し，動作確認できましたか？(選択必須)" with radio buttons for "はい" and "いいえ".
- Question 2: "問題2のプログラムを作成し，動作確認できましたか？(選択必須)" with radio buttons for "はい" and "いいえ".
- Question 3: "問題3のプログラムを作成し，動作確認できましたか？(選択必須)" with radio buttons for "はい" and "いいえ".
- A text input field for "今回の授業に関する質問や感想などがあれば書いてください。".
- Two buttons at the bottom: "中断" (Cancel) and "提出確認" (Submit Confirmation).

終了：今日のまとめと次回予告

■今日の授業

- if文で条件を満たした場合の処理が指定できる
- elseで条件を満たさなかった場合の処理を指定できる
- prompt for number ▼ with messageでキーボードからメッセージ付きで数字が入力できる
- else ifで多方向に分岐できる
- andで「かつ」 orで「または」という条件式を作ることができる

■次回は繰り返しについて