

1. はじめに
2. ビジュアルプログラミング
3. おわりに

ビジュアルプログラミング

情報基盤工学講座 横井 稜

1. はじめに
2. ビジュアルプログラミング
3. おわりに

August 26, 2020

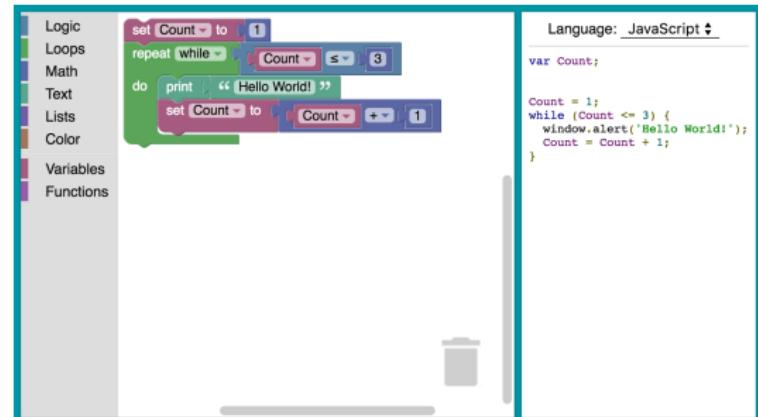
はじめに

2/10

本研究

1. はじめに
2. ビジュアルプログラミング
3. おわりに

近年、企業などでは世間に溢れる様々な情報を収集し、ビッグデータと呼ばれる非常に巨大で複雑なデータの集合として扱うことが増えてきている。しかし、そのデータを機械により処理する知識を得るには莫大な時間が必要である。本研究の目的は、データ処理の未経験者でもビッグデータを解析できるようにすることである。



Logic Loops Math Text Lists Color Variables Functions

set Count to 1
repeat (while Count ≤ 3)
do
print "Hello World!"
set Count to Count + 1

Language: JavaScript

```
var Count;  
  
Count = 1;  
while (Count <= 3) {  
    window.alert('Hello World!');  
    Count = Count + 1;  
}
```

VPL

- はじめに
- ビジュアルプログラミング
- おわりに

プログラムをテキストで記述するのではなく、視覚的なオブジェクトで記述するプログラミング言語のこと。視覚的でわかりやすいものが多いため、プログラムの組み立て方を学ぶのに有効であると注目されている。



メリット

1. はじめに
2. ビジュアルプログラミング
3. おわりに

- ① 直感的に操作できる
- ② 学習が比較的簡単（記憶する必要がほぼない）
- ③ 文字列を経験豊富なプログラマーが認識するように表示してくれる
- ④ 間違いにくい

デメリット

- ① プログラムの修正に時間がかかる
- ② コードを検索しにくい
- ③ テキストとブロックの対訳を作るのが難しい
- ④ 本格的なプログラミング学べない

ブロックタイプ

- ① ブロックを使う
- ② ブロックを並べる順序や論理構造の作り方は、テキスト言語に似ている
- ③ オープンソースのものが多い

フロータイプ

- ① フローチャートを使う
- ② ブロックタイプと見た目の違いがあるだけで基本的なプログラムの作り方は同じ

独自ルールタイプ

- ① 文字や数字で指示や論理構造を作ったり考えたりするのではなく、より直感的な独自の手法でプログラムを作る
- ② テキストプログラミングに移行するのは難しい

- はじめに
- ビジュアルプログラミング
- おわりに

目標として、MAGELLAN BLOCKS ができる数値回帰をできるようにする。

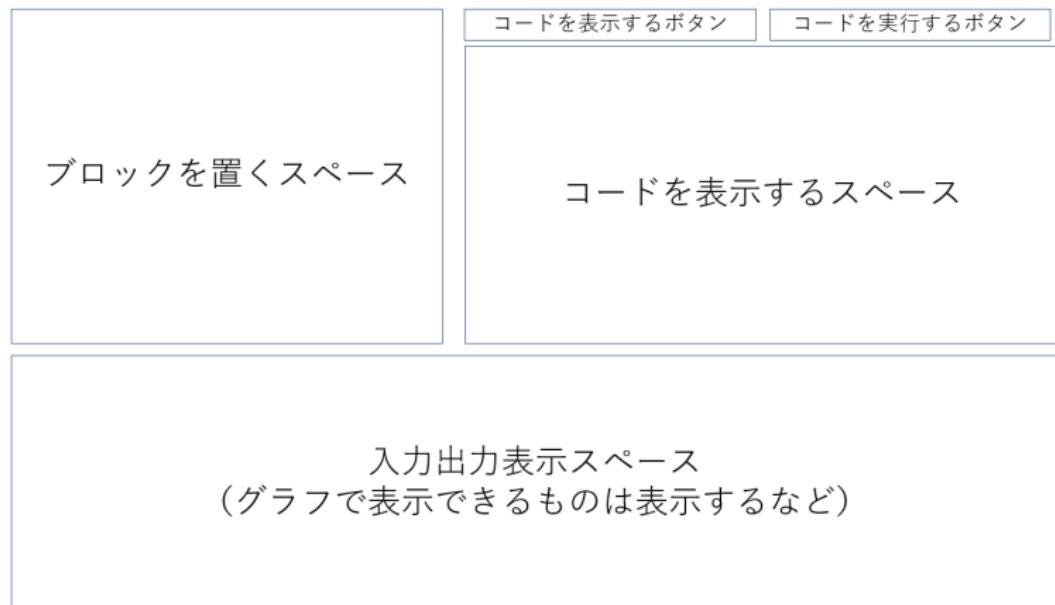


図1 システムの概略図

制限

1. はじめに
2. ビジュアルプログラミング
3. おわりに

- ① 1行目の3列で入力の数、出力の数、データの総数を指定
- ② 入力の列が出力の列より左側になるように作成する

	A	B	C	D	E	F
1	5	1	7			
2	1	2	1	2	2	3
3	4	5	4	5	5	6
4	7	8	7	8	3	2
5	3	4	3	4	5	4
6	2	5	2	5	6	1
7	1	4	2	5	5	9
8	2	3	4	5	6	1

図2 データ例

最初に作るシステム

8/10

1. はじめに
2. ビジュアルプログラミング
3. おわりに

最小二乗法による線形回帰ができるシステムを作成する。入力としてヘッダーと x と y 、出力として傾きと y 切片と相関係数を出す。入力方法としては、エクセルファイルを使う。出力方法としては、横軸 x 、縦軸 y の図とエクセルファイルを用いる。

未作成の必要なもの

- 1 エクセルファイル入出力ブロック
- 2 最小二乗法による線形回帰ブロック
- 3 入出力図表示システム

1. はじめに
2. ビジュアルプログラミング
3. おわりに

python と **javascript** は **python** 側をサーバーとして起動することで連携できる。なので、回帰分析などのデータ処理のブロックの中に **python** にメッセージを送るプログラムを記述し、**python** にてプログラムを実行して、その終了時に **javascript** へメッセージを送ってプログラムを再開させる。

- ① **python** → **javascript** 未完成
- ② **javascript** → **python** 完成

メリット

- ① **python** にはライブラリが沢山あり自分でデータ処理プログラムを記述する時間を省ける
- ② 研究室には **python** に関する本がたくさんある

まとめ

1. はじめに
2. ビジュアルプログラミング
3. おわりに

- 1 カスタムブロック及びシステムの作成

今後の課題

- 1 カスタムブロック及びシステムの作成
- 2 python と javascript

-
- 1 どんなデータ処理のブロックを作ってほしいか