

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

September 3, 2020

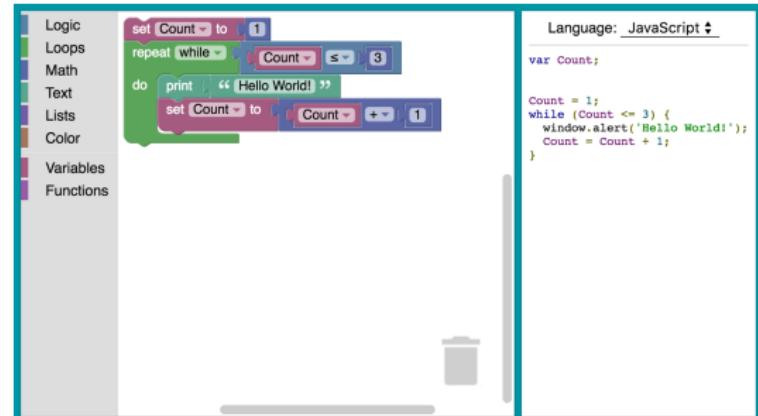
# はじめに

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。



The screenshot shows a Scratch-like programming interface. On the left, there's a sidebar with categories: Logic, Loops, Math, Text, Lists, Color, Variables, and Functions. In the center, a script is displayed:

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

On the right, the language is set to JavaScript, and the generated code is:

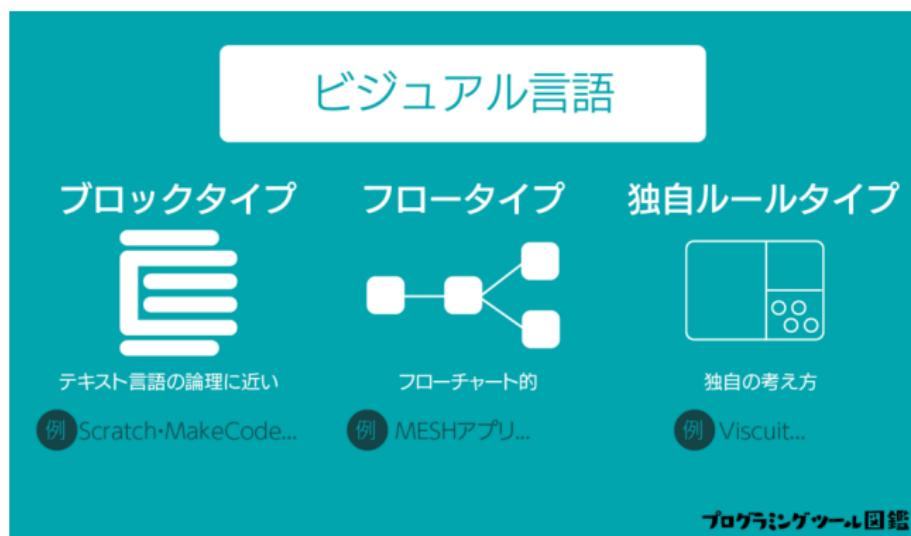
```
var Count;

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

# システム完成図

6/10

- 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。

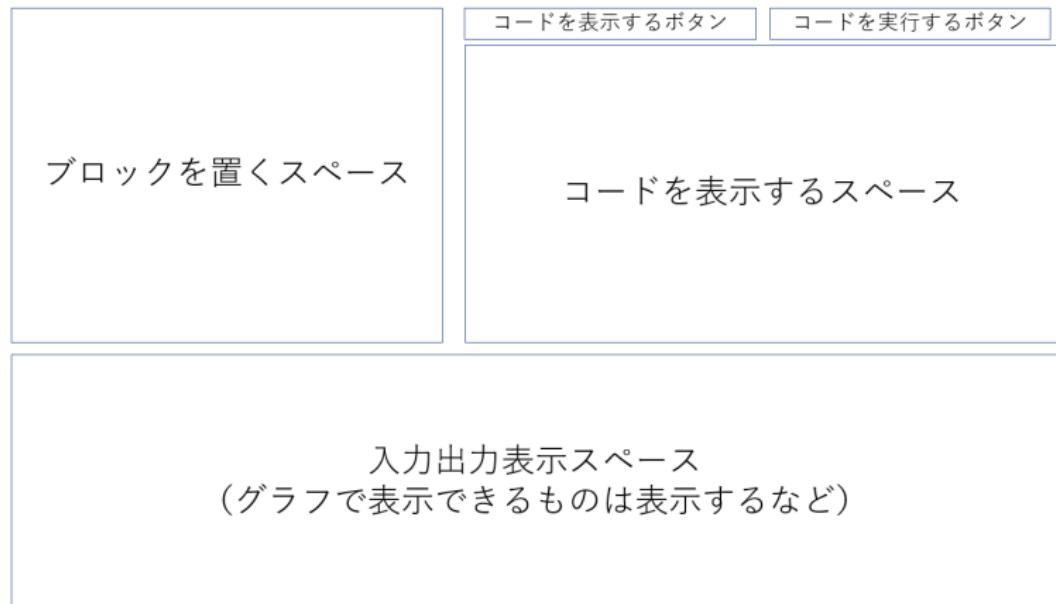


図1 システムの概略図

# 今回研究でやったこと

7/10

1. はじめに
2. ビジュアルプログラミング
3. おわりに

- 1 とりあえず、Javascript のみを使った一通りの動きの完成
- 2 web 上に公開
- 3 python と js の通信
- 4 研究室サーバーの設定の確認

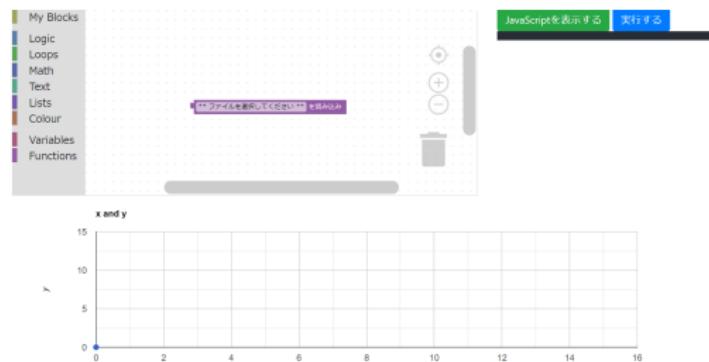


図 2 blockly

# 最低限のシステムで作成したもの

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

- 1 ファイル入力ブロック
- 2 最小二乗法による回帰分析とその結果ブロック
- 3 ファイル出力ブロック
- 4 入力図表示システム



図 3 ブロック

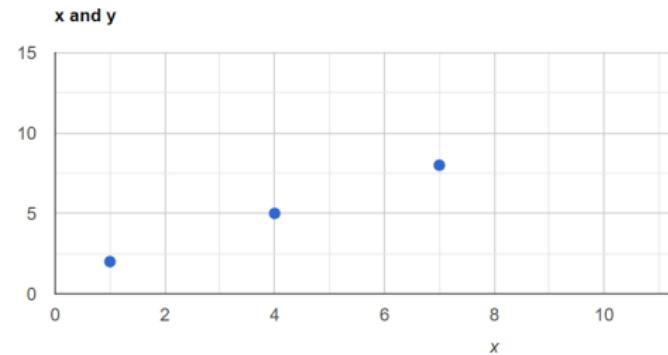


図 4 図表示システム

# いろいろとデモ

9/10

1. はじめに
2. ビジュアルプログラミング
3. おわりに

## 研究室サーバー一覧

1. はじめに
2. ビジュアルプログラミング
3. おわりに

- ① **dic515mas.pu-toyama.ac.jp(133.55.115.238)**
- ② **dic515s1.pu-toyama.ac.jp(133.55.115.239)**
- ③ **dic515s2.pu-toyama.ac.jp(133.55.115.240)**

今のところ、ほとんど機能は無いので機能拡張していきたい。欲しい機能などあれば。

## 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
- ② **python** と **javascript** の通信の完成
- ③ サーバーの管理及び運営

## 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** のデータ処理のサーベイ
- ③ **blockly** のデータ処理の部分を **python** にメッセージを送るよう  
に変更
- ④ **python** サーバーの立ち上げ
- ⑤ サーバーの管理及び運営方法のサーベイ