



多言語プログラミング

→ プログラミングにおける骨格

電気技術研究会

NIT, NARA COLLAGE

FEBRUARY 19, 2023

概要

講義目的と進め方

講義目的

- 多言語にわたるプログラミングを独学する術を身に付ける
- 基本的な用語や技術を学び、活かせる力を身に付ける
- マイコンやウェブなど、広い範囲に適応できる人材育成

進め方

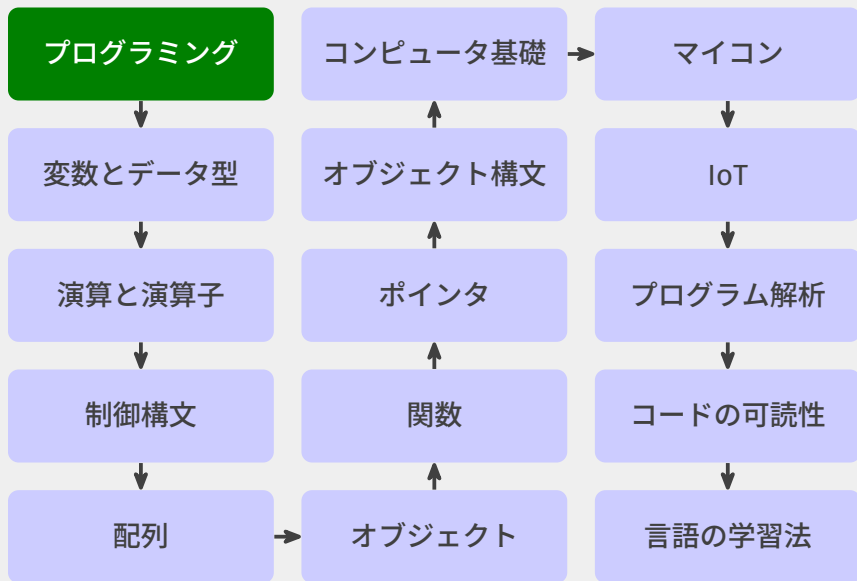
- 重要な章ごとにまとめて講義
- 1日（2～3時間）1～2章分を進めたい
- 各章ごとに演習がある

カリキュラム



プログラミングとは

進度



プログラミングについて

プログラミングどんなもの？

- プログラミングには様々な種類が存在
ウェブ、マイコン、ゲーム...etc
- 大まかには2種類
オブジェクト指向と構造化プログラミング（後述）
- 骨格は基本的に共通
関数や変数など、基本的な骨格がある

プログラミング言語

- 言語は多数ある
C, C++, C#, JavaScript, VisualBasic, Python, \LaTeX
- 本教材では主に JavaScript と Arduino による C++ を使用

基本的な骨格とは

- 冒頭で**ライブラリ**の宣言
- **メインループ**で処理
UI 等の場合では**イベント**を処理
- **サブルーチン (関数)**にて
繰り返し処理

C 言語の例

```
#include <stdio.h>

int main(void){
    printf('Hello World');
    return 0;
}
```

Arduino の例

```
void setup(){
    pinMode(2, OUTPUT);
}

void loop(){
    digitalWrite(2, HIGH);
    delay(200);
    digitalWrite(2, LOW);
    delay(200);
}
```


演習 1 : 問題

↪ プログラミングとは

JavaScript 動作確認

1. VSCode の作業フォルダを作成
2. 'Hello World' を表示させる

HTML ファイルへの埋め込み

- html ファイルの<script>タグ内
- ディスプレイさせる関数
`document.writeln(' 文字列')`
- 文末には ; をつける
- 文字列は ' か " で囲う

html テンプレート

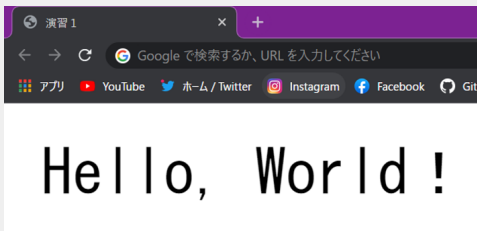
```
<html>
<head>
<meta
  http-equiv="Content-Type"
  content="text/html; charset=UTF-8">
<title>Hello, World! </title>
</head>
<body>
<pre>
<script type="text/javascript">
  // ここにスクリプトを記述
</script>
<noscript>
  JavaScriptが利用できません。
</noscript>
</pre>
</body>
</html>
```

演習 1 : 解答

↪ プログラミングとは

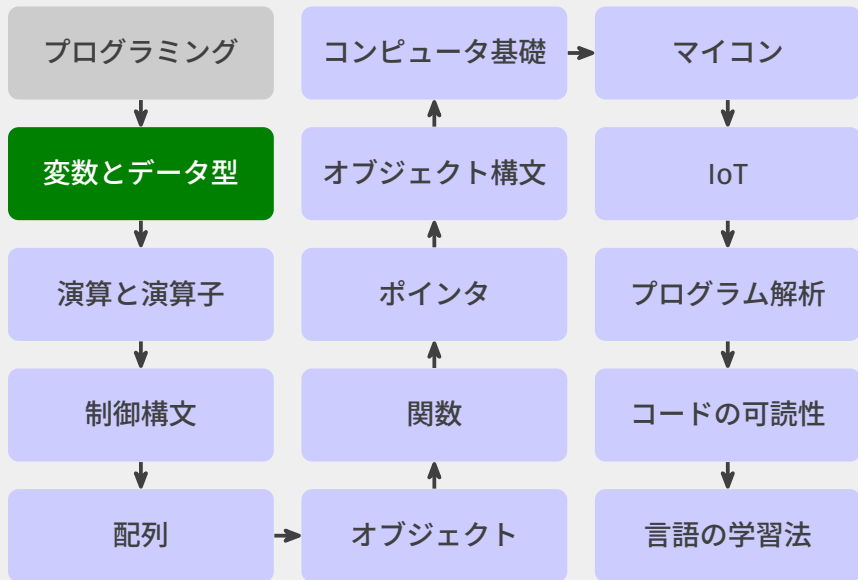
演習 1

```
document.writeln('Hello, World!');
```



変数とデータ型

進度



変数

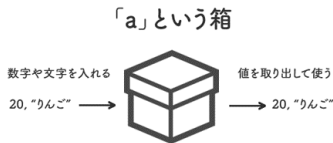
↪ 変数について

変数とは

- データを入れる箱
- 入れるデータによって種類がある
- 箱をひとまとめにしたもの ▷ 配列

変数の基本要素

- 宣言
変数を作成すること
- 初期化
変数の作成とともに数値を割り当てること
- 代入
変数の値を上書きすること



変数

↪ 宣言

- どのような言語でも**基本おなじ** (python では宣言が暗黙)
- (修飾子) + 型 + 変数名で宣言

```
let x;  
int x;
```

- 変数名は**予約語**以外なら原則何でも可

変数

↪ 初期化

- 宣言とともに値を割り当てる
(python では宣言と初期化がセット)
- 型 + 変数名 = 初期値で宣言 + 初期化をセットで

```
let x = 100;
```

```
int x = 100;
```

- 静的型付けでは初期値の型が一致している必要がある。

```
int x = 'a';
```

初期化をしないと...

ランダムに適当な値 (不定値) が割り当てられる

javascript では"undefined"と出力される

▶ 予測してない結果になる可能性がある



基本的に宣言と初期化はセットです

変数

↪ 代入

- 変数を上書きして内部の値を更新する

- 宣言後、任意の場所で 変数名 = 値

```
let x = 100;  
x = 200;
```

- =は実は演算子

演習 2 - 1 : 問題

↪ 変数とデータ型

`let` 型の変数を用いて `Hello World` を表示させる

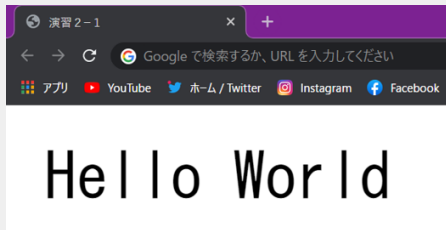
1. 宣言は 型 (今回は `let`) + 変数名 (適当)
2. 初期化は 宣言 = 初期値
3. 文字列は ' か " で囲う
4. `document.writeln(変数名)` を使用して表示させよう

演習 2 - 1 : 解答

↪ 変数とデータ型

演習 2 - 1

```
let str='Hello World';  
  
document.writeln(str);
```



データ型

↪ データ型について

データ型

↪ 動的型付けについて

データ型

↪ データ型の使い分け

演習 2 - 2 : 問題

↪ 変数とデータ型

演習 2 - 2 : 解答

↪ 変数とデータ型

演算と演算子

進度



制御構文

進度



配列

進度



オブジェクト

進度



関数

進度



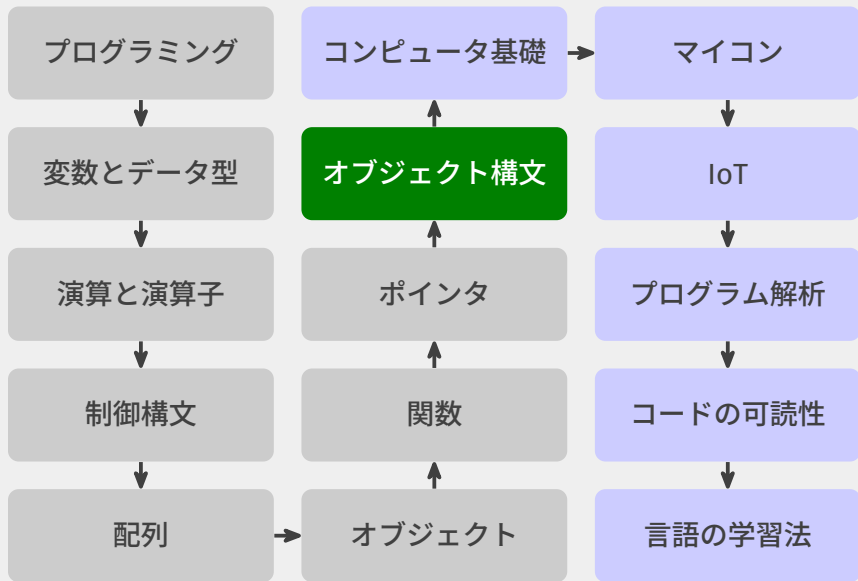
ポインタ

進度



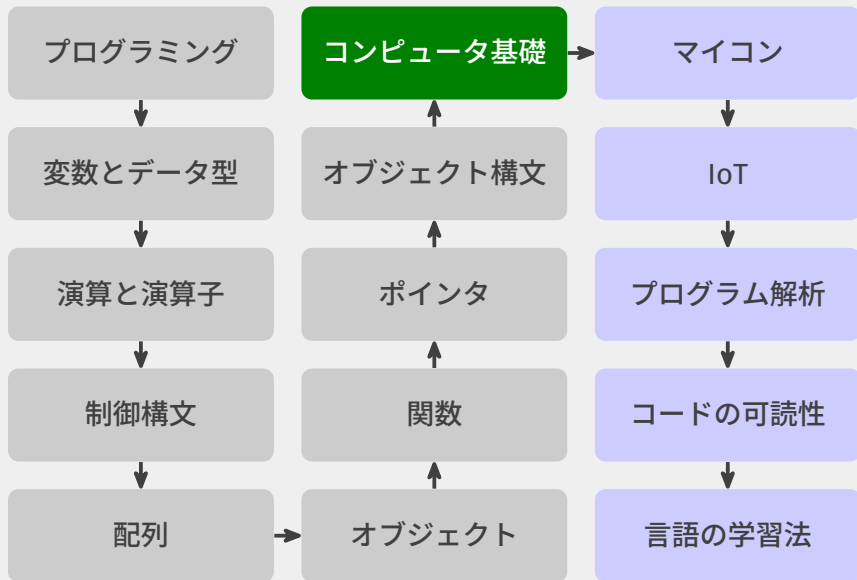
オブジェクト指向構文

進度



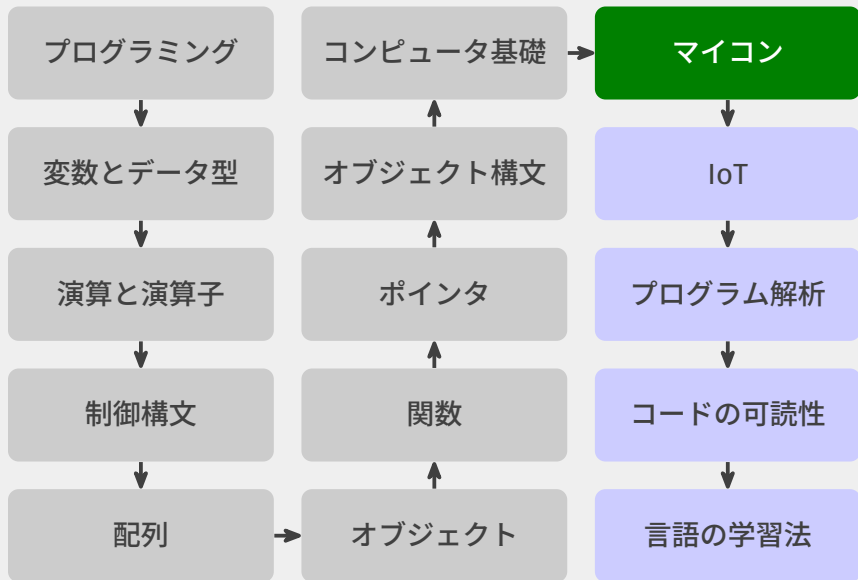
コンピュータ基礎

進度



マイコン

進度



IoT

進度



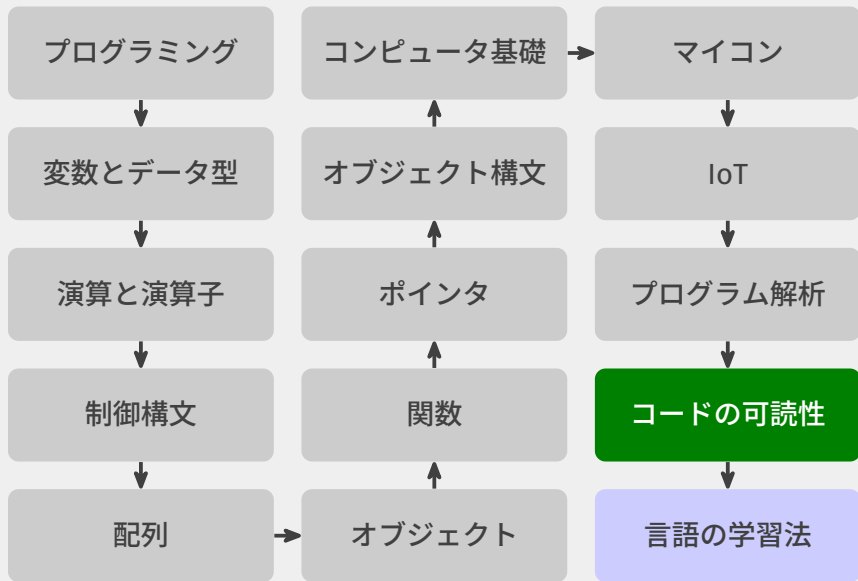
プログラム解析

進度



コードの可読性

進度



言語の学習法

進度



問題解決能力