

Arduino 入門

いらいざ (@Eliza_0x)

2016/7/21

今回勉強するもの

C言語

C 言語勉強会

Arduino は出てきません

なんでC言語なん？

- Arduino 言語は C 言語を便利にしたもの
- C 言語が理解できれば分かる

プログラムとは何か

- 機械への指示をリストしたもの
- なおかつ人間が読めるもの

人間が読めるもの

- この事は非常に大切
- 人間がよめなきゃ意味がない!

人間が読めるもの

にんげんはねえ プログラムかくため 生
まれてきたんじゃないんだよにんげんが
さき プログラムはあと みつを

人間が読めるもの

C言語を理解して人間が分かるコードを書こう

プログラムを書こう

その前に

関数ってなんだったか思い出そう

\sin, \cos, \tan や二次関数をおもいだせますか？

関数ってなんだったか思い出そう

関数のイメージですが、値をあたえたら値を返してくれるものという認識で良いと思います

- $\sin(0)=0$
- $\sin(30)=0.5$
- $\tan(45)=1$

それではプログラミングを始めましょう

Let's try!

Hello, World

```
#include<stdio.h>
```

```
int main(void){  
    printf("Hello,World");  
}
```

プログラムを理解しよう

`"Hello,World"`

文字列です

プログラムを理解しよう

```
printf("Hello,World");
```

出力の関数です

プログラムを理解しよう

```
#include<stdio.h>
```

入出力に関連した関数を準備しています

プログラムを理解しよう

int は整数という意味です

```
# int main(void)
```

main 自体も実は関数ですこれは main は何も受け取らず、数値を返すという意味です

プログラムを理解しよう

sin は数字を受け取って数字を返す関数ですね？

■ $\sin(30)=0.5$

プログラムを理解しよう

- main はプログラムのいちばん始めに動作する関数
- なにも受け取れない
- プログラムが終了後がコンピュータにきちんと動作したかの数字を返している

プログラムを理解しよう

```
int main(void)
```

変数を理解しよう

- 数学で x を5とするのような表記を思い出してほしい。
- `int x = 5` \div x を5とする
- `int` は整数という意味でしたね

変数を理解しよう

```
int x, y, z;
```

```
x = 5;
```

```
y = 6;
```

```
z = x + y;
```

この場合zの中身は何でしょう？

変数を理解しよう

```
int x, y, z;
```

```
x = 5;
```

```
y = 6;
```

- 整数があてはまる x,y,z が準備される
- x を 5
- y を 5 とする

変数を理解しよう

もう $x + y$ は分かるよね

変数を理解しよう

int みたいな値の情報をあらわしているものを型と
いって

- string “Hello”
- float 1.23
- int 123

などがある

条件分岐しなきゃね

- プログラムの世界の「もし～ならば」です

```
int main(void){  
    int x, y;  
    x = 5;  
    if(x == 5){  
        printf("x = 5");  
    } else {  
        printf("x not = 5");  
    }  
}
```

条件分岐しなきゃね

```
if(x == 5){  
    printf("x = 5");  
}
```

- == は比較演算子といいます
- 両端の数字を受け取る関数で、同じなら True, 違うなら False を返します。
- ifは次に書いてある () の中で True が帰ってきたなら、その次に書いてあることを実行します
- 比較演算子は == 以外にも >=, <=, != などがあります

条件分岐しなきゃね

ところで、選択肢が増えると書くのは面倒ではありませんか？

```
if(x == 5){  
    printf("x = 5");  
} else {  
    printf("x not = 5");  
}
```

- ifは「もし～」ならば
- elseは「もし～でなかったならば」

単純作業は嫌いだ

- 50回5をかけてください
- たぶん面倒なプログラムになる

5 * 5 * 5 * 5 * 5 * 5*

単純作業は嫌いだ

while はカッコの中が True の限り中に書いてある事を繰り返しつつづけます

```
int x, y;  
y = 1;  
while(50 >= x){  
    y = y * 5;  
    x = x + 1;  
}
```

関数

- 関数ごとに機能をまとめると使い勝手がよくなる
- なんどもおなじような事を書いている場合は関数にまとめられないか考えよう
- たとえば $+$ は関数

関数

- 足し算する関数 add を作ろう

```
int add(int x,int y){  
    return x + y;  
}
```


ここでちからつきました

(じつは無睡眠)

Any Question?