

# C言語講習会

#3 変数

# 1. 前回のおさらい

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

`#include <stdio.h>` の意味

stdio.h という名前のヘッダファイルをプログラムの中に含んでね！！

```
int main(void){ }
```

main関数の意味

準備ができたなら、{ }で囲まれた部分からプログラムを実行してね！！

# 1. 前回のおさらい

```
printf( "Hello World!!\n" );
```

printf関数の意味

引数に渡した文字列を画面に表示してね！！

```
return 0;
```

return 0; の意味

関数を終了して、外側に戻り値(0)を返してね！！

## 2. 変数

### 変数とは

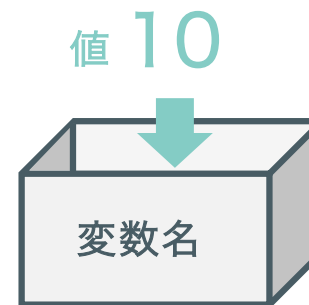
値を入れておくための箱のようなものを **変数(variable)** といいます。

プログラムの中で変数を使うには、あらかじめ変数を用意しておかなければなりません。これを 変数を **宣言** するといひ、以下のように記述します。

### 変数の宣言

**データの型** 変数名 ;

変数名 : 変数を識別するための名前



例 : 変数のイメージ

## 2. 変数

### 変数名の付け方

変数名の付け方には、以下のような細かなルールがあります。

- 半角の英数字と `_` (アンダースコア) を使う
- 英字の大文字と小文字は区別される
- 変数名を数字で始めることはできない
- C言語の他の命令で既に使われている用語 (予約語) は使えない

次のページで詳しく見ていきましょう



## 2. 変数

- 半角の英数字と \_ (アンダースコア) を使う

```
apple  
orange01  
fruits_basket
```

- 英字の大文字と小文字は区別される

```
apple  
Apple
```

← この2つは別の変数

## 2. 変数

- ・ 変数名を数字で始めることはできない

× 123apple

- ・ C言語の他の命令で既に使われている用語 (予約語) は使えない

× int

× return

○ int\_return

○ Return

予約語

void char short int long float double auto static  
const signed unsigned extern volatile register return  
goto if else switch case default break for while do  
continue typedef struct enum union sizeof

## 2. 変数

### 上手な変数名の付け方

// 変数に方向の値を入れる場合

- |             |              |
|-------------|--------------|
| ✘ banana    | 全く関係のない変数名   |
| ○ direction | 適切な英単語       |
| ○ dire      | 意味が通じる範囲での略語 |
| △ hokou     | ローマ字表記       |
- 
- |                 |                 |
|-----------------|-----------------|
| △ message       | 何のメッセージかわからない   |
| ○ error_message | メッセージの内容が想起しやすい |
| ○ err_msg       | 意味が通じる範囲での略語    |



## 2. 変数

変数を宣言する際に初期化

```
int apple;
```

変数に整数を代入する

```
apple = 10;
```

数学では **=** は等価の意味だが、  
C言語では**代入**の意味になる。

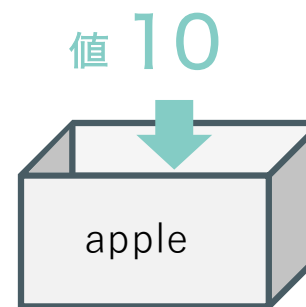


図: 変数appleのイメージ

## 2. 変数

変数を宣言する

```
int apple = 10;
```

変数に整数を代入する

✗ apple = 3.14;

✗ apple = 'e';

**int型**の変数 appleには、  
小数や文字を入れることができない。

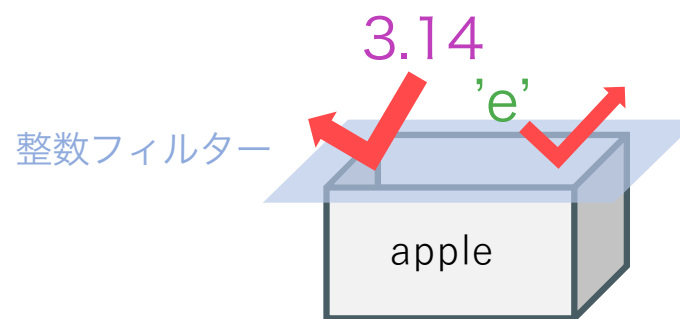


図: int型の変数apple

## 2. 変数

### ・変数を宣言する

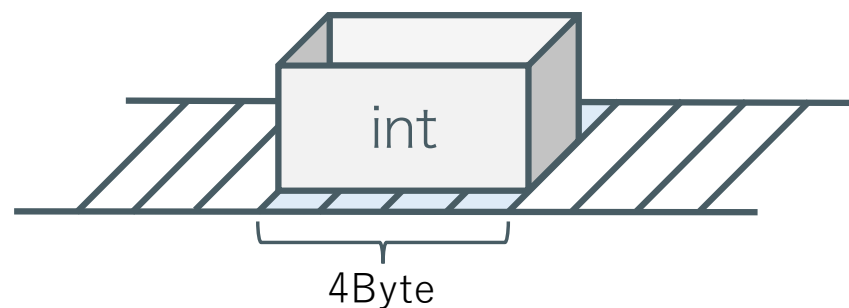
型の名称	読み方	型のサイズ	型の意味	保存
char	キャラ/チャー	1Byte	文字型	英数字1文字 または -128 ~ 127
int	イント	4Byte	整数型	-2,147,483,648 ~ 2,147,483,647
float	フロート	4Byte	単精度不動 小数点数型	$1.175494 \times 10^{-38} \sim$ $3.402823 \times 10^{38}$
double	ダブル	8Byte	倍精度浮動 小数点数型	$2.225074 \times 10^{-308} \sim$ $1.797693 \times 10^{308}$
void	ボイド	0Byte	空のデータ型	

## 2. 変数

### ・変数を宣言する

型の名称	読み方	型のサイズ	型の意味	保存
char	キャラ/チャー	1Byte	文字型	英数字1文字 または -128 ~ 127
int	イント	4Byte	整数型	-2,147,483,648 ~ 2,147,483,647
float	フロート	4Byte	単精度不動	$1.175494 \times 10^{-38} \sim$

**int型**の変数は、メモリ上に  
4バイトの領域を確保する。



## 2. 変数

代入する値に応じて変数の型を変える

- float pi = 3.141592;
- double napier = 2.71828;
- char exp = 'e';

変数の型の先頭につける**修飾子**というものもあります。

型の名称	読み方	型のサイズ	型の意味	保存
unsigned int	アンサインド イント	4Byte	符号なし整数型	0 ~ 4,294,967,295
short int	ショート イント	2Byte	短整数型	-32,764 ~ 32,767
long int	ロング イント	4Byte	長整数型	-2,147,483,648 ~ 2,147,483,647

## 2. 変数

- ・ int型の変数 apple を表示する方法は？

× `printf( "apple ¥n" );`



apple

今までのようにappleを " " で囲う

apple と表示される

"apple" は文字列、apple は変数名

× `printf( apple );`



zsh: segmentation fault

では、変数appleをそのまま入れる

セグメンテーション違反が起きる  
(ルール違反してメモリにアクセスすること)

## 2. 変数

変数appleを表示する方法

```
printf( "%d ¥n", apple );
```



実行

10

フォーマット指定子

指定子	意味	対応する型
%d	整数を10進数で表示	int
%c	文字(1文字)を表示	char
%g	実数を最適な方で表示	float
%lf	倍精度実数を表示	double

・ 複数の変数の整数を表示する方法

```
printf( “ぶどう%dこ と みかん%dこ ¥n”, grape, orange );
```

↑  
grape

↑  
orange

順番通りに代入される

# 参考文献

- ・ 大川内隆郎, 大原竜男, *かんたん C言語* [改訂2版], 技術評論社, 2017.
- ・ 笥捷彦, 高田大二 他, *入門 C 言語*, 実教出版株式会社, 2019.