

C言語とは

- 1972年にデニス・リッチーが開発
- 当時は高水準言語だったが今では低水準言語に分類される。

バイブル

<https://www.amazon.co.jp/プログラミング言語C-第2版-ANSI規格準拠-B-W-カーニハン/dp/4320026926>

他の言語との歴史の比較

- C 1973年
- COBOL 1959年
- Java 1994年
- PHP 1959年
- Python 1991年
- VB 1991年
- Rust 2009年

参考:

<https://ja.wikipedia.org/wiki/プログラミング言語の比較>

本題 ポインタの仕組み

参考: <https://ja.wikibooks.org/wiki/C言語/ポインタ>

メモリ(主記憶)にはアドレスと言う番号が振られている。
それぞれのアドレスの領域は1バイトです。
ポインタはこのアドレスを格納している変数のことです。

データ型とサイズ

char: 1バイト

short: 2バイト

int: 4バイト

float: 4バイト

double: 8バイト

ポインタの参照

```
#include <stdio.h> //おまじないとよく言われるやつ。標準入出力のためのライブラリ

int main(void) {
    int i = 55;
    int *pi = &i; //piにはiのアドレスが格納されます。
    *pi = 1234;
    printf("%d\n", i)
}
```

実行結果

1234

アドレスの演算

```
#include <stdio.h>
#include <stddef.h> // for ptrdiff_t

int main(void) {
    const int ary[] = {123, 55, 818, 1192, 777};

    const int *a = &ary[0]; // 要素0のアドレスを格納
    printf("&ary[0] = %p\n", a);

    const int *b = &ary[1]; // 要素1のアドレスを格納
    printf("&ary[1] = %p\n", b);

    ptrdiff_t delta = b - a; // 配列の要素間の距離（間にある要素の数）
    printf("&ary[1] - &ary[0] = delta = %td\n", delta);

    const int *v = ary + delta;
    printf("*vは%d\n", *v);
}
```

実行結果

```
&ary[0] = 0x7ffe0052c910  
&ary[1] = 0x7ffe0052c914  
&ary[1] - &ary[0] = delta = 1  
*vは55
```


関数ポインタ

ポインタを指すポインタを宣言できる

```
#include <stdio.h>

int main(void) {
    int i,
        *pi = &i,
        **ppi = &pi; // 理論上何重にもネストできる

    **ppi = 1234;
    printf("%d", i); // 1234が出力される
}
```

セグフォ(セグメンテーションフォールト)

以下のような場合に発生してプログラムが終了します。(デバッグしにくい)

- アクセス禁止のメモリ領域にアクセスしたり、読み込み専用のメモリに書き込もうとしたとき
- 再帰が深すぎる時