

第
5

回 C 言語講座

テーマ:ポインタ

ポインタとは？

- ポインタとは変数のメモリ上のアドレス情報を格納したもののこと。

じゃアドレスって何？

- アドレス＝住所、メモリー上での場所のこと。(マンションの部屋番号みたいなもの)

とりあえず Let's coding!

- void main(void)
- {
- int x=5;
- int *p;
- p=&x;
- printf("x=%d , *p=%d ¥n",x,*p);
- }

コードの説明

- `int x=5;`
- 整数型の変数`x`を宣言し、5で初期化(`x`に代入)。
- わからない人は、前の講座の資料参照||ググレ。

コードの説明

- `int *p;`
- **整数型のポインタpを宣言。**
- **変数名の前に"*"をつけるとポインタになる。**
- `int* p;`でもok。
- **例:**
- `char *pChar;`// **文字型ポインタを宣言**
- `long *pLong;`// **ロング型ポインタを宣言**
- `void *pVoid;`// **ボイドポインタを宣言**

コードの説明

- `p=&x;`
- `p`に`x`のアドレスを代入。
- 変数名の前に“&”をつけるとその変数のアドレス(メモリ上の場所(住所))になる。

コードの説明

- `printf("x=%d , *p=%d ¥n",x,*p);`
- `x`と`*p`を表示。
- "`*p`"としているところがポイント。
- そのまま`p`とすると、アドレスが表示されるので、中身(ポインタの示しているものこの場合は`x`)を表示するためには、"`*`"をつけて"`*p`"とする。

- ここまでわかりましたか？
- わからなくても大丈夫。ファミ通の(ry・・・ではなくて、
- ポインタはC言語の要であり、一番の山場といっても過言ではない！（ポインタだけの解説書とかある。）
- 今日理解できなくても、それが普通です。
- ってか、こんな解説で完璧に理解できるんならC言語講座受けなくても(ry

ポインタを利用した関数への参照渡し。

値渡しの場合

- `int hogehoge(int input)`
- `{`
- `input ■■■`
- `return ■■■■`
- `}`
- `void main(void)`
- `{`
- `int x=123;`
- `hogehoge(x);`
- `}`

値渡し

- これは値渡しでの呼び出しという。(call by value)
- つまり、inputという変数(実引数という)に値を代入して、
- 呼び出しているようなもの。
- イメージ
- `input=x`
- `hogehoge(int input)`

- だからinputの内容を変更してもxには反映されない。
- だから、反映されるためにポインタを使う。
- まあ、他にもいろいろ便利なんですけど。

とりあえず Let's coding!

- `void swap(int* x, int* y)`
- `{`
- `int tmp;`
- `tmp=*x;`
- `*x=*y;`
- `*y=tmp;`
- `}`

- `void main(void)`
- `{`
- `int a=3,b=9;`
- `printf("a=%d , b=%d\n",a,b);`
- `swap(&a,&b);`
- `printf("a=%d , b=%d\n",a,b);`
- `}`

コードの解説

- `swap(&a,&b);`
- ここで、`swap(int*,int*)`を呼んでいる。

引数の型がアドレスなので、それに合わせて経変数 `a,b` の前に `&` をつけて `&a`、`&b` にすることで、アドレスを渡している。

コードの解説

```
Void swap(int *x,int *y)
```

```
tmp=*x;
```

tmpに*xの値を代入。

*xの値って???

→ swap(&a,&b);で呼ばれているので、aの値
(x=&a)。

この場合は、*x=a=3なので、tmp=3となる。

コードの解説

```
Void swap(int *x,int *y)
```

- `*x=*y;`
- `*x`へ`*y`を代入する。
- この場合、`*y=b=9`なので、`*x=9`となる。
- さらに、`x`は`main()`中の`a`のポインタなので、
- `a=*x=9`。

コードの解説

```
Void swap(int *x,int *y)
```

- `*y=tmp;`
- `*y`へ`tmp`の値を代入している。
- `tmp`は2つ前のステップで3が入っているので、
- `b=*y=tmp=3`

おわり

- おつかれさまでした。