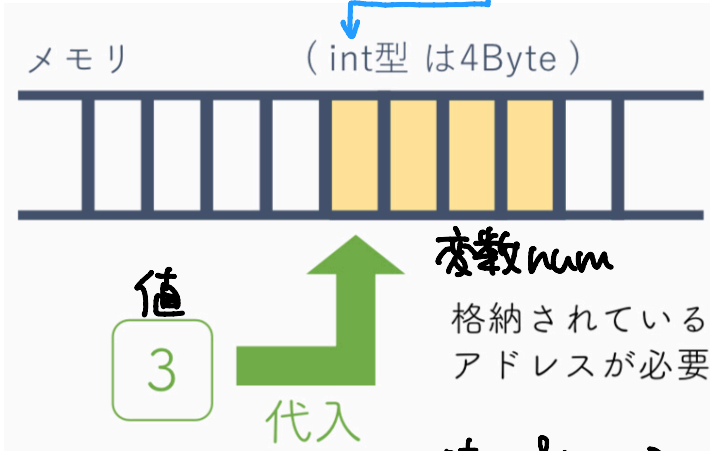


# ポインタ

✓ 複習

```
int num = 3;
```

<アドレス>  
変数numのアドレスは  
0012FF50



```
pt = &num;
```

✓ アドレスの取得

**&変数名**

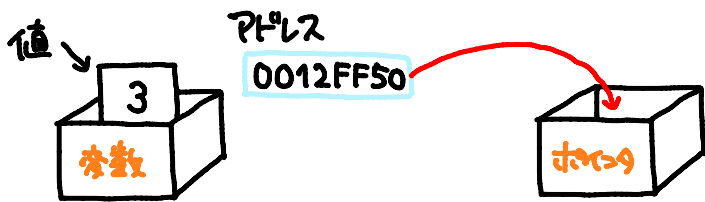
```
printf("%p\n", &a);
```

(実行結果)

1回目: 0x7ffec4681698  
2回目: 0x7ffecfe7698  
↳ 毎回アドレスが異なる.

✓ ポインタとは

→ アドレスの値を格納するための変数



pointer = pointer (指す人)

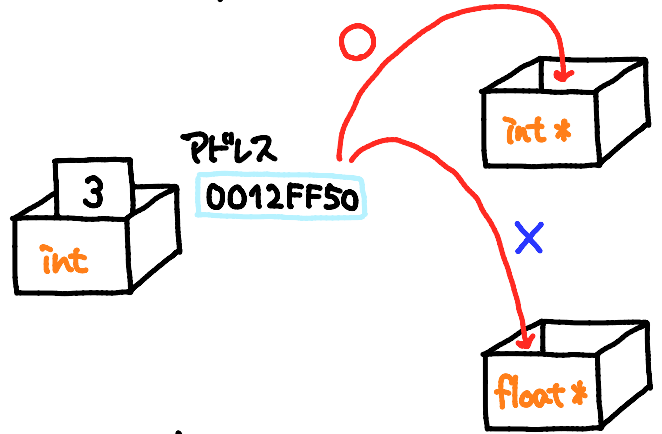
✓ ポインタ変数

**型名 \*変数名**

```
int a;  
int *ptA; ← ポインタ変数を宣言  
ptA = &a; ← ポインタ変数に  
          アドレスを格納
```

# 注意

```
int *ptA; ← 宣言には*が付く  
ptA = &a; ← ポインタ名には*が付かない
```



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

```
int a = 3;  
int *ptA;  
ptA = &a;
```

アドレスが表示される

```
printf("%p\n", &a);  
printf("%p\n", ptA);
```

同じ結果

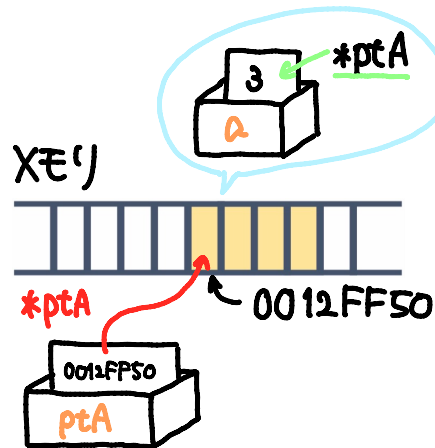
✓ アドレスから値を参照する.

**\*ポインタ変数名;**

値が表示される

```
printf("%d\n", a);  
printf("%d\n", *ptA);
```

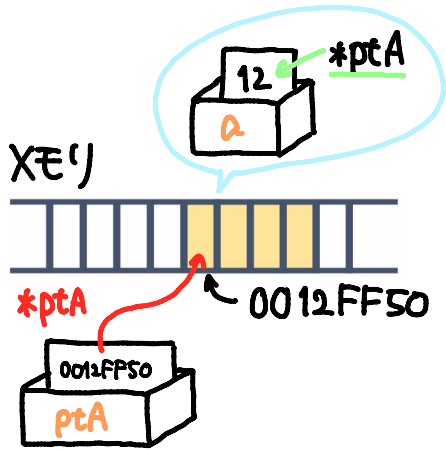
同じ



Q. \*ptA = 12; とすると、aの値はどのように変わるだろう?

```
*ptA = 12;
printf("%d\n", a);
```

➔ 12 と表示



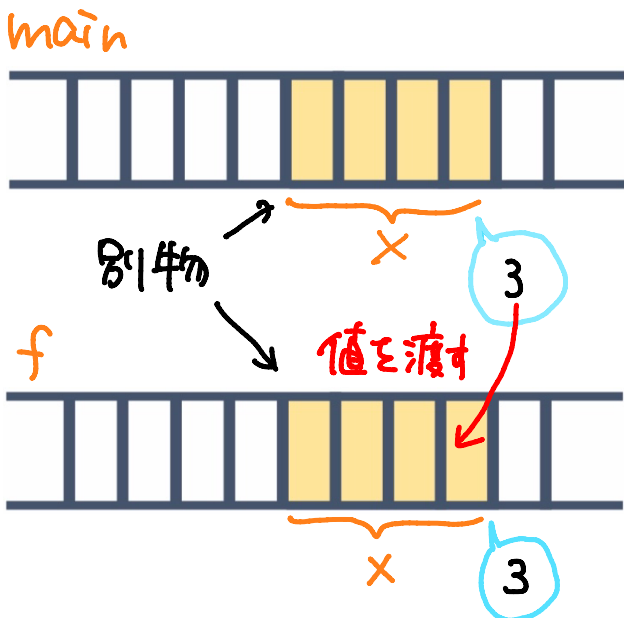
② 値渡しと関数.

✓ 値渡し

```
void f(int x) {
    x = x * 2;
    return;
}

int main(void) {
    int x = 3;
    f(x);
    ...
}
```

f(x) に a の値を渡す.

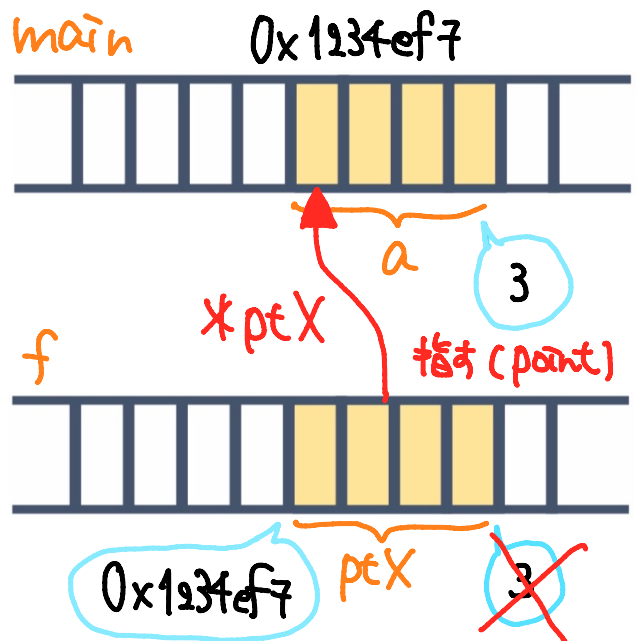


✓ 値渡し

```
void f(int *ptX) {
    *ptX = (*ptX) * 2;
    return;
}

int main(void) {
    int a = 3;
    f(&a);
    return 0;
}
```

f(x) に アドレスを渡す



ptX は 3 への値を 持つていい。