

# メモリ, メモリアドレス

(2進数, 16進数, メモリ, 論理演算, 画像と画素)

URL: <https://www.kkaneko.jp/cc/math/index.html>

金子邦彦



# メモリとは



- **メモリ**は、データの記憶を行うチップ
- データを覚えさせたり（**書き込み**）、取り出したり（**読み出し**）の機能がある

# メモリとアドレス



- **メモリはバイト (8ビット) 単位に区切られている**
- **各バイトには0から始まる通し番号**が付けられている。これを**アドレス**という (番地ともいう)

メモリ内のデータは

01	00	00	00	02	00	00	00	03	00
----	----	----	----	----	----	----	----	----	----

0 1 2 3 4 5 6 7 8 9

アドレス

# メモリアドレス

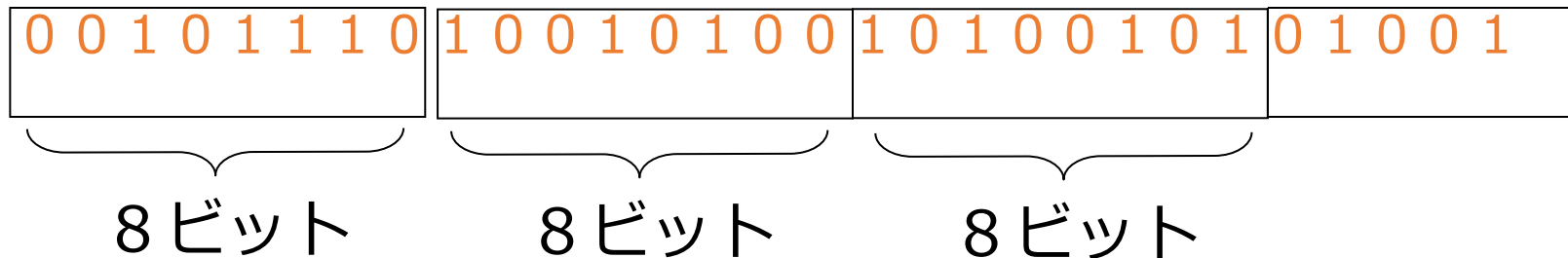


- メモリアドレスは、読み書きすべきデータの「場所」を示す

(例) 0065FDF0 (16進)

→ メモリ先頭から0065FDF0 (16進数) 番目  
という意味

- メモリ内のデジタルデータは、8ビットずつ区切られて、メモリアドレスが付けられている



# メモリアドレスは、ふつう16進数表記する



```
C [dropdown] Enter a title here
Main.c [x] [+]
1 #include <stdio.h>
2 int main(void){
3     double teihen = 3;
4     double takasa = 4;
5     double menseki;
6     menseki = teihen * takasa / 2.0;
7     printf("&teihen %x\n", &teihen);
8     printf("&takasa %x\n", &takasa);
9     printf("&menseki %x\n", &menseki);
10 }
11
```

実行 (Ctrl-Enter)

出力 コンパイルエラー 入力 コメント 0

```
&teihen cf323990
&takasa cf323988
&menseki cf323980
```

16進数



実行結果の例

# なぜ 16 進数なのか



- メモリアドレスそのものもデジタル（「0」, 「1」の列）
- といって, メモリアドレスを, 「0」, 「1」の並びで書くのは, 長すぎて人間にとって分かりづらい

(例)

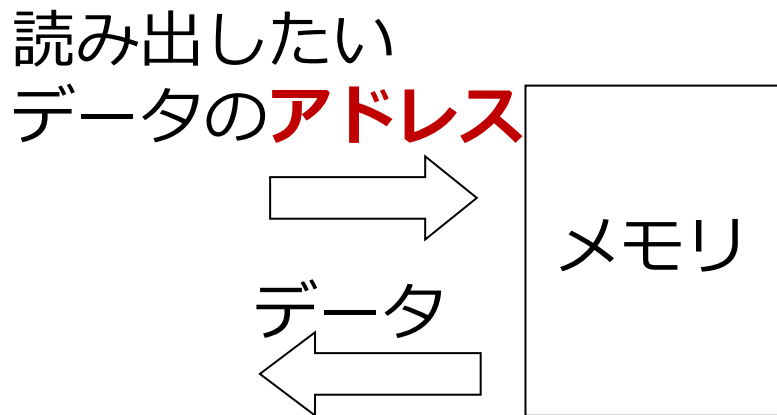
0 0 1 0 1 1 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 0 1

そこで, 「16 進数」を使う

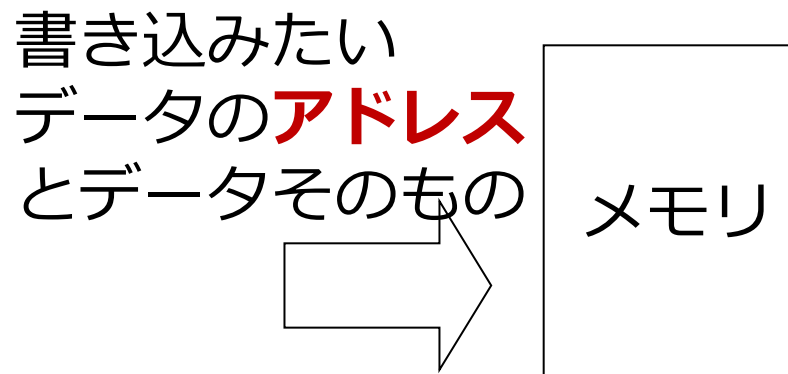
# メモリへの操作



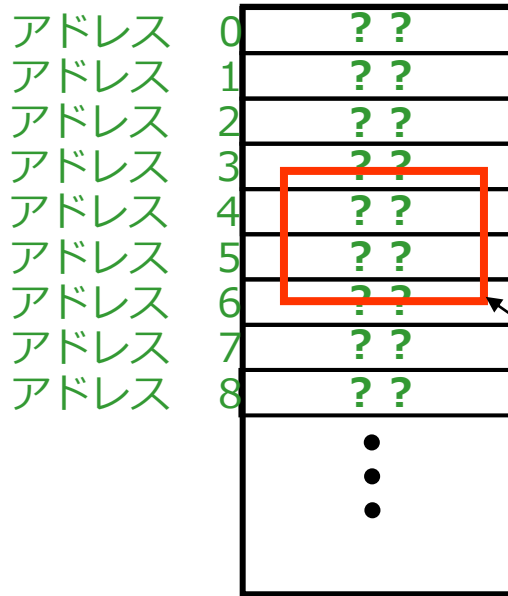
## • 読み出し



## • 書き込み



# 読み出し



アドレス4番地, 5番地  
から2バイト分  
読み出すとき

メモリの値は変化  
しない

メモリの各区画は1バイト  
(16進数で2桁)

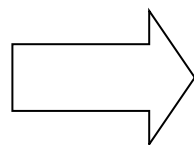


# 書き込み



アドレス 6 番地,  
7 番地に  
「0400」を  
書き込むと

アドレス	0	??
アドレス	1	??
アドレス	2	??
アドレス	3	??
アドレス	4	??
アドレス	5	??
アドレス	6	??
アドレス	7	??
アドレス	8	??
		⋮



	??
	??
	??
	??
	??
	??
	??
	0 4
	0 0
	??
	⋮

メモリの各区画は1バイト  
(16進数で2桁)

前の値は消える