

# 1 ポインタによる配列表現

学習のポイント

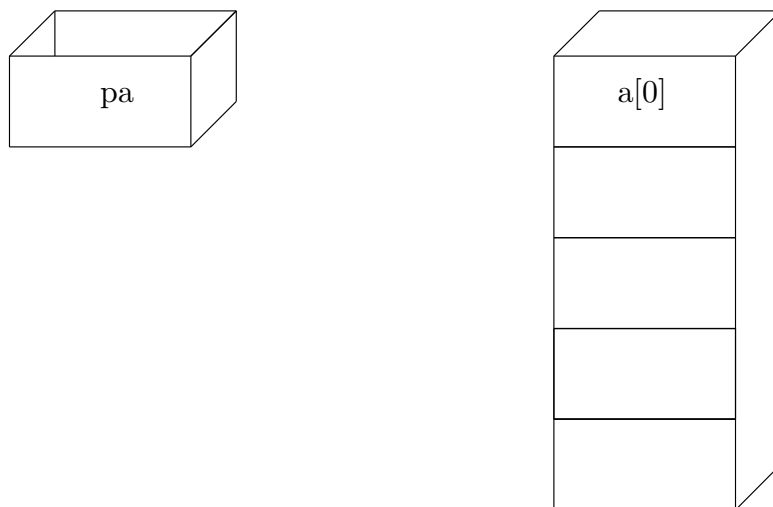
ポインタを用いた配列表現方法と、そのようにして表現された配列と実体のある配列との違いを学びます。

先の例では、ポインタ `pa` の値を次々に進めて配列 `a[ ]` の各データを参照しましたが、`pa` を固定しておいて、そこから何番目のデータというように指定することもできます。たとえばポインタ `pa` を用いて配列 `a[ ]` の `i` 番目の要素を参照するには、次の 2 通りの表現ができます。

`*(pa+i)`  
`pa[i]`

2 つとも `pa` が持つアドレスに `i` を加えたアドレスの内容を取り出しています。C の配列表現を行う `[ ]` はアドレスを計算する演算子と考えられます。つまり、`pa[i]` を論理的にみると配列 `pa[ ]` の `i` 番目の要素ですが、物理的にみると、`pa+1` というアドレス計算をし、そのアドレスが示す内容を取り出しているのです。

ここで注意することは、`pa[0]` から `pa[9]` という領域が新たに確保されているのではなく、実在するのはポインタ変数 `pa` の 2 バイトの領域だけであることです。`pa[ ]` はポインタ変数 `pa` のアドレスに `i` を加えたアドレスの内容を取り出す式 `*(pa+i)` と同じものなのです。「C 言語」(河西朝雄著 ナツメ社)88 頁



## 1.1 例題 17

ポインタにより指し示されている文字列を後の文字から表示しなさい。

```
/*
 例題 17 C 言語 89 頁
 ポインタにより指し示されている文字列を後の文字から表示しなさい。
  reidai17.c
*/

#include <stdio.h>

int main()
{
    char *str = "Hello";
    int i;

    for(i = 4; i >= 0; i--){
        putchar(str[i]);
    }

    return 0;
}
```

## 1.2 練習問題 17

ポインタにより指し示されている文字列を逆に並べて配列 `rev[ ]` に入れるプログラムを作りなさい。

```
/*
  練習 17 C 言語 89 頁
  ポインタにより指し示されている文字列を逆に並べて配列 rev[] に入れるプログラムを
  作りなさい。
  rensyu17.c
*/

#include <stdio.h>

int main()
{
    char *str = "Hello";
    char rev[10];
    int i;

    for(i = 0; i <= 4; i++){
        rev[4 - i] = str[i];
        rev[5] = '\0';
    }

    printf("%s", rev);

    return 0;
}
```