

# プログラミング基礎

<http://bit.ly/prog2d>

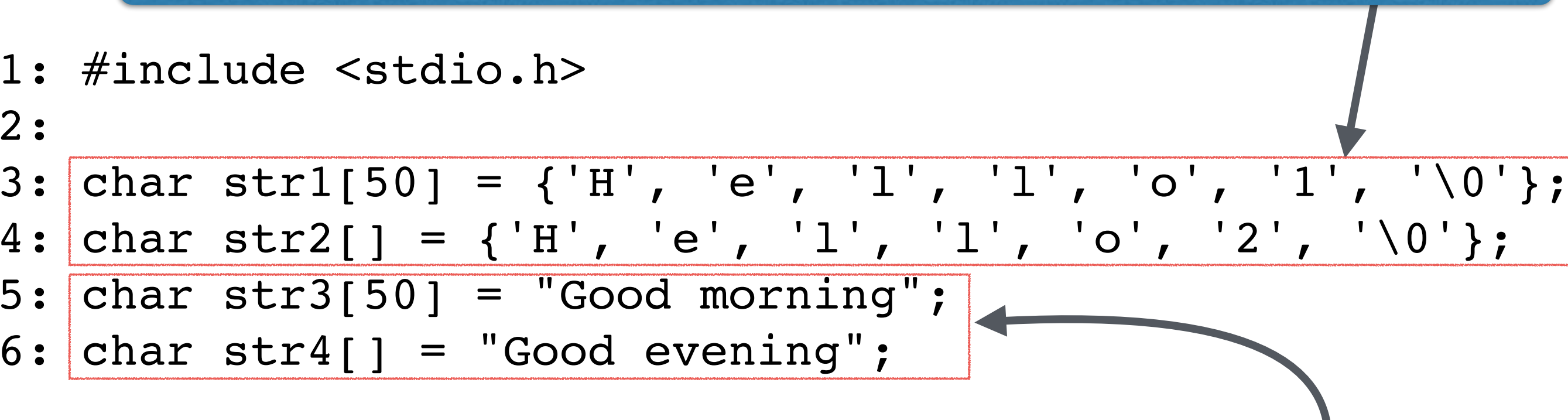
## 配列と文字列

前期 第6週

2017/5/22

## 【Point 1】 文字列（文字の配列）を初期化する (p. 204~205)

```
1: #include <stdio.h>
2:
3: char str1[50] = {'H', 'e', 'l', 'l', 'o', ' ', '\0'};
4: char str2[] = {'H', 'e', 'l', 'l', 'o', '2', '\0'};
5: char str3[50] = "Good morning";
6: char str4[] = "Good evening";
```



## 【Point 2】 初期化時のみ " " で囲った代入が可能 (p. 205)

【Point 3】 **char型の配列**を使うと文字列を表現することができる。文字列は、配列の要素に一文字ずつ格納され、最後に '¥0'（または'\0'）という**NULL文字（終端文字）**が格納される。そのため、文字列を配列に格納するためには、「**文字列の長さ+1**」の要素数が最低でも必要となる。例えば、文字列str1は下図のように格納される。(p. 204参照)

'H'	'e'	'l'	'l'	'o'	' '	'¥0'		.....	
str1[0]	str1[1]	str1[2]	str1[3]	str1[4]	str1[5]	str1[6]	str1[7]		str1[49]

【Point 4】 文字列に対する繰り返し処理は、  
「**'¥0'** ではない間繰り返す」という条件が  
使える (p. 209)

```
7:
8: int main( )
9: {
10:     int i, l;
11:
12:     for(i=0; str1[i]!='\0'; i++) {
13:         printf("str1[%d]: %c\n", i, str1[i]);
14:     }
15:
```

【Point 5】 文字列の配列の要素1つは、  
文字つまり**char型**として扱うことができる


【Point 6】 printf(), scanf()により文字列の入力と出力の変換仕様には **%s** を使う (p. 207～208)

```
16: printf("str2: %s\n", str2);  
17:  
18: printf("str3: %s\n", str3);  
19: printf("str3 > ");  
20: scanf("%s", str3);  
21: printf("str3: %s\n", str3);  
22:
```

【Point 7】 scanf()では、配列名の前に **&** を付けない  
(厳密には、&str3[0]を省略した書き方) (p. 207～208)

文字列の最初の文字（0番目の文字）から、終端文字になるまで繰り返す処理となり、**文字数**（終端文字を除く）をカウントしている

```
23:     l = 0;
24:     for(i=0; str4[i]!='\0'; i++) {
25:         l++;
26:     }
27:     printf("%sの文字数は%dです。\\n", str4, l);
28:
29:     return 0;
30: }
```



# 【練習6-1】

サンプルプログラムをコンパイルして、  
実行結果を確認しましょう。

# 【課題6-1】

配列str1に格納されている文字列の文字数を返す関数str\_length()を作成してください。

[この関数のプロトタイプ宣言]

```
int str_length();
```

```
/* サンプルプログラムの23～26行目にある繰り返し処理を使う */
```

```
/* 求めた文字数1を戻り値とする */
```

# 【課題6-1】

[ 配列str1をグローバル変数として宣言する ]

```
char str1[50];
```

[main()での処理]

```
int r;  
printf("str1 > ");  
scanf("%s", str1);  
r = str_length();  
printf("str1の文字数: %d\n", r);
```

[実行結果]

str1 > HelloWorld.      (← "str1 > " が出力され、続けて入力した)

str1の文字数: 11



# 【課題6-2】

配列str1に格納された文字列に対して、  
「'\_'（アンダーバー または アンダースコア）」を  
「' '（空白）」に置き換える関数replace\_space()  
を作成してください。

[この関数のプロトタイプ宣言]

```
void replace_space();
```

```
/* p.213 練習5 を参考に作ることができる */
```

```
/* この処理の結果は配列str1に反映される */
```

# 【課題6-2】

[ 配列str1をグローバル変数として宣言する ]

```
char str1[50];
```

[main()での処理]

```
printf("str1 > ");  
scanf("%s", str1);  
replace_space();  
printf("str1: %s\n", str1);
```

[実行結果]

```
str1 > This_is_a_pen.  
str1: This is a pen.
```

# 【課題6-3】

配列str1に格納され文字列に対して、仮引数の文字chを探す関数find\_char()を作成してください。

[この関数のプロトタイプ宣言]

```
int find_char(char ch);
```

```
/* 文字chが文字列str1にあるかどうかを探す */
```

```
/* 見つかった場合は、配列の何番目にあったのかを戻り値とする */
```

```
/* str1にchが複数含まれている場合は最初に見つかった方でよい */
```

```
/* 見つからなかった場合は戻り値として-1を返す */
```

# 【課題6-3】

[main()での処理]

```
/*   (以下は課題6-2の続きとなる、  
      つまり、str1には「This is a pen.」が格納されている)   */  
  
printf("find: %d\n", find_char('s'));  
printf("find: %d\n", find_char('z'));
```

[実行結果]

```
find: 3  
find: -1
```

# 【課題6-4】

配列str1に格納され文字列に対して、**逆順に並べた**文字列を配列str2に格納する関数reverse\_str()を作成してください。

[この関数のプロトタイプ宣言]

```
void reverse_str();
```

```
/* str1と同じサイズの配列str2をグローバル変数として用意する */
```

```
/* まず、str_length()を使ってstr1の文字列の長さを調べる */
```

```
/* str1の文字列の末尾の方から繰り返し、
```

```
    str2の先頭から文字をstr1の文字を1つずつ代入していく */
```

```
/* str2に格納した文字列の最後に'\0'を代入する */
```

```
/* この処理によってstr1の文字列は変更されない */
```

# 【課題6-4】

[ 配列str1とstr2をグローバル変数として宣言する ]

```
char str1[50], str2[50];
```

[main()での処理]

```
printf("str1 > ");  
scanf("%s", str1);  
reverse_str();  
printf("str1: %s\n", str1);  
printf("str2: %s\n", str2);
```

[実行結果]

```
str1 > GoogleGlass  
str1: GoogleGlass  
str2: ssalGelgooG
```

# まだ余裕のある人は…【課題6-5】

配列str1に格納された文字列に対して、x番目を境目に文字列を二分割し、配列str2, str3に格納する関数divide()を作成してください。

[この関数のプロトタイプ宣言]

```
void divide(int x);
```

```
/* 分割された文字列を格納する配列str2, str3を  
   グローバル変数で宣言しておく（要素数はstr1と同じでよい） */  
/* まず、0番目～x-1番目の文字をstr2へコピーし、  
   次に、x番目～文字列の最後までをstr3へコピーする */  
/* str2, str3へコピーした文字列の最後には、  
   '\0'をそれぞれ代入する必要があることに注意する */  
/* この処理によってstr1の文字列は変更されない */
```

# 【課題6-5】

[ 配列str1, str2, str3をグローバル変数として宣言する ]

```
char str1[50], str2[50], str3[50];
```

[main()での処理]

```
printf("str1 > ");  
scanf("%s", str1);  
divide(5);  
printf("str2: %s\n", str2);  
printf("str3: %s\n", str3);
```

[実行結果]

```
str1 > AppleSony  
str1: AppleSony  
str2: Apple  
str3: Sony
```



# 小テストの追試について

**6月2日（金） 13:00**に、小テストの追試験を実施します。未受験の小テストがある人は、この時に必ずまとめて受けて下さい。  
（やむを得ず欠席する際は事前に連絡を下さい。）

未受験分の小テストは0点として評価します。

# 定期試験の実施について

## 試験中に使用できるもの

- 筆記用具

(メモ用紙が必要な人には試験中に配布する)

- 演習室のコンピューター台

(一つの机に一人の配置で、座る場所はどこでもよい)

# 定期試験の実施について

## 試験中に参照できるもの

- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル  
（定期試験では紙媒体のものは参照不可）
- 上記以外の情報を参照することは不正行為とする  
例：USBで接続された機器に保存されているファイルの参照  
ネットワークを介した情報の参照、など
- 試験中は、演習室外へのネットワークアクセスは遮断される