

プログラミング基礎

<http://bit.ly/prog2d>

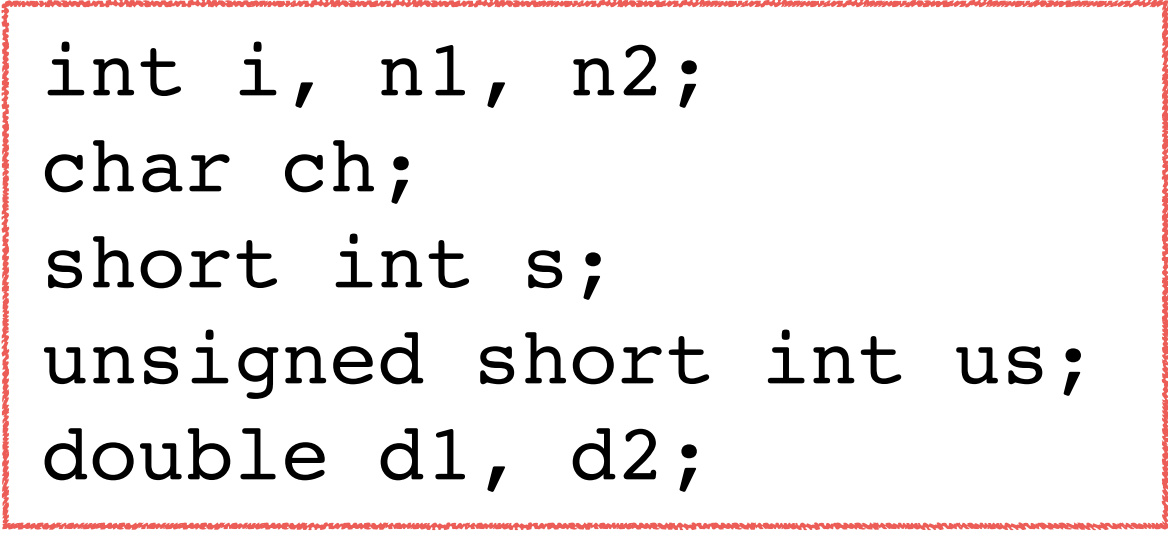
数値表現 (1)

前期 第11週

2017/7/3

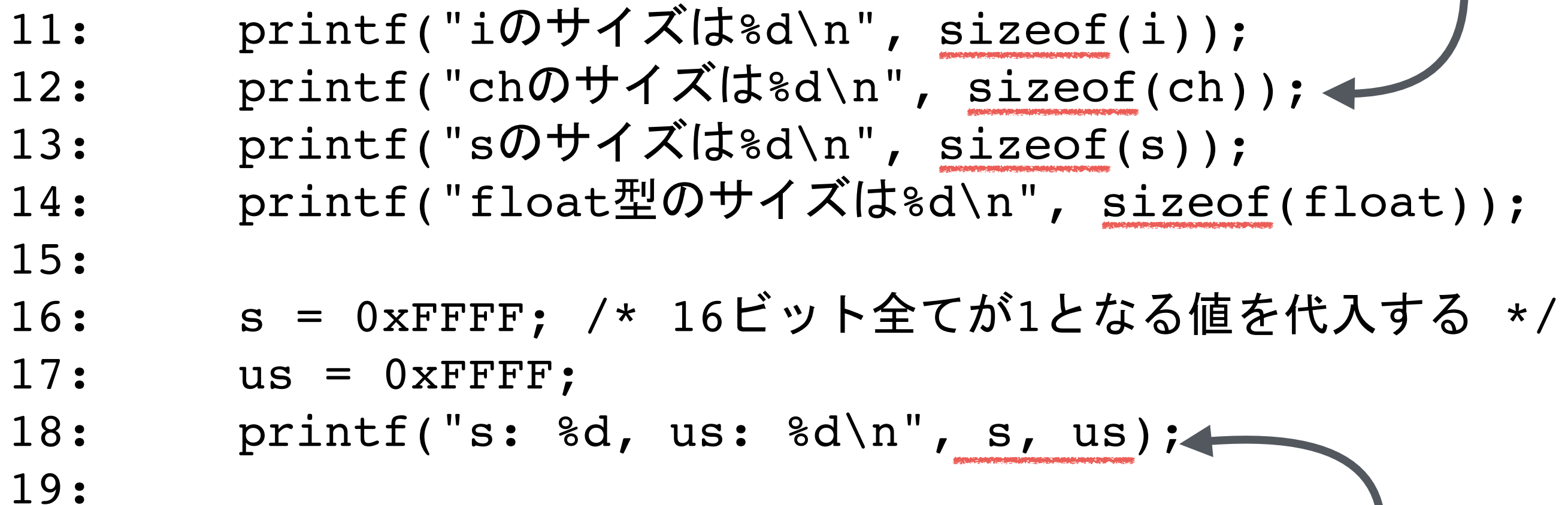
【Point 1】Cには「**型**」に応じて格納できる情報（値）の種類が決まる。格納できる値の**サイズも型によって決まる**。
(p.49 表3-1)

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     int i, n1, n2;
6:     char ch;
7:     short int s;
8:     unsigned short int us;
9:     double d1, d2;
10:
```



【Point 2】 **sizeof**演算子を使うと、型のサイズ（**バイト**単位）を得ることができる。sizeof演算子の後ろには、変数名, 型名, 式を指定できる。 (p.92～93)

```
11:    printf("iのサイズは%d\n", sizeof(i));
12:    printf("chのサイズは%d\n", sizeof(ch));
13:    printf("sのサイズは%d\n", sizeof(s));
14:    printf("float型のサイズは%d\n", sizeof(float));
15:
16:    s = 0xFFFF; /* 16ビット全てが1となる値を代入する */
17:    us = 0xFFFF;
18:    printf("s: %d, us: %d\n", s, us);
19:
```



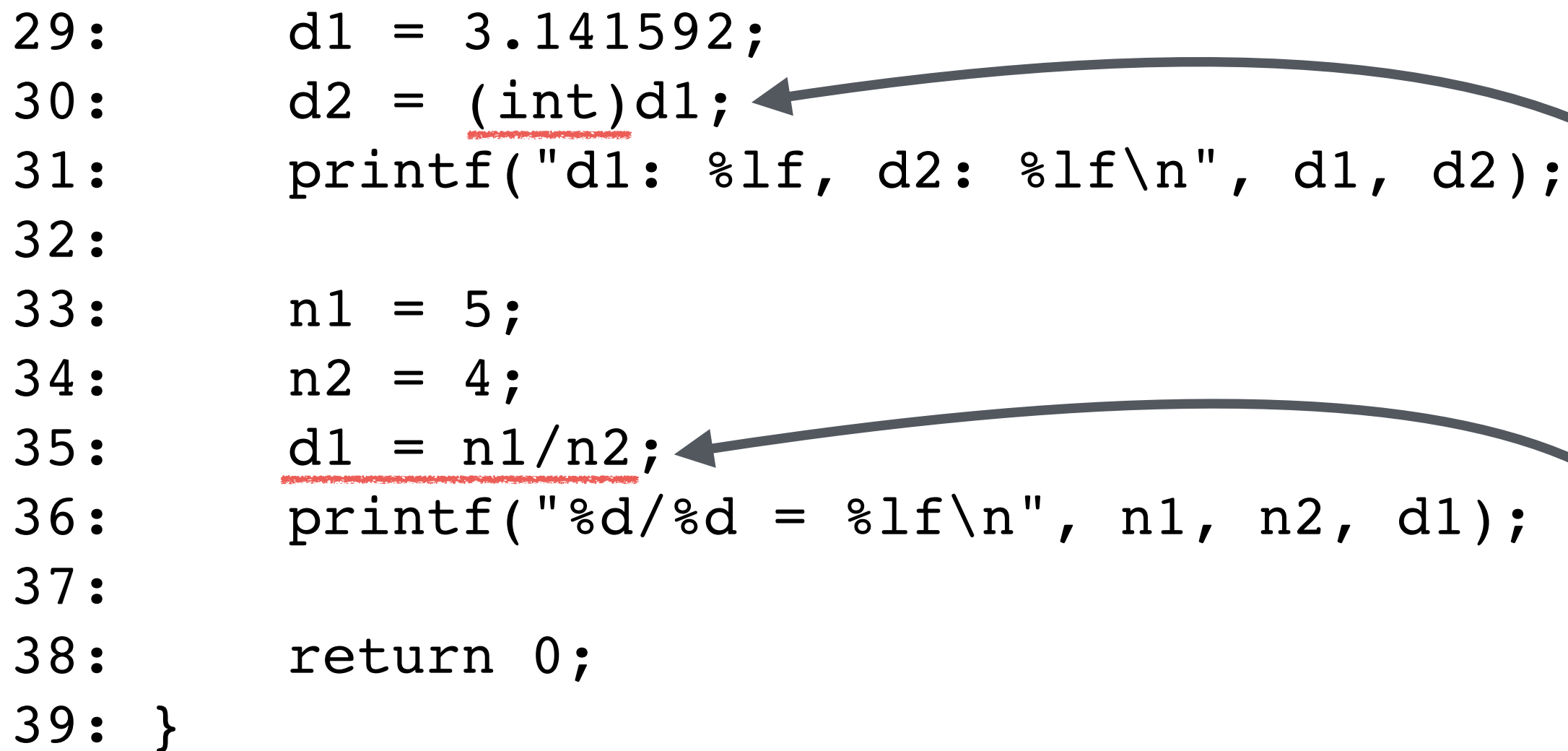
【Point 3】 **unsigned**が付く型は、「**符号なし**」として数値を扱う。「**符号あり**」の場合と比べて表現できる**値の範囲が異なる**。
(p.49 表3-1) 0xFFFFは、符号ありの場合、2の補数で -1 を表現して、符号なしの場合、 $2^{16} = 65535$ を表している。

【Point 4】 数値を書く時、**8進数**で表す場合は値の先頭に「**0**」を付け、**16進数**の場合は「**0x**」を付ける。 (p.38)

```
20:      i = 16; i = 0xf; i = 020;
21:      printf("%d\n", i);
22:
23:      i = 32;
24:      printf("%d %x %o\n", i, i, i);
25:
26:      printf("16進数を入力 > "); scanf("%x", &i);
27:      printf("%d\n", i);
28:
```

【Point 5】 printf(), scanf()の変換仕様で、**8進数**は「**%o**」、**16進数**は「**%x**」を使う。
(p.390 表12-1, 表12-2, p.395)


```
29:    d1 = 3.141592;
30:    d2 = (int)d1;
31:    printf("d1: %1f, d2: %1f\n", d1, d2);
32:
33:    n1 = 5;
34:    n2 = 4;
35:    d1 = n1/n2;
36:    printf("%d/%d = %1f\n", n1, n2, d1);
37:
38:    return 0;
39: }
```



- 【Point 6】 **キャスト演算子**を使うと、明示的に型変換ができる。
(p.104～105)
- ・ 式の計算結果となる値を表現する型は、**式に使う値の型**で決まる。
(「n1/n2」はint型として扱われる)
 - ・ 異なる型への代入は、**自動的に型変換される**場合がある。
(d1への代入はdouble型に変換されている)
- (p.101～109)

【練習11-1】

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。

【練習11-2】

**p.107 Sample11.cのプログラムの説明を参考に、
1ページ目のプログラムの36行目の出力が意図した
通りの結果となるように変更してみましょう。**

【課題1 1-1】

double型, long int型, long long int型のサイズを
出力するプログラムを作ってください。

【課題11-2】

整数の配列array（要素数は5個程度）をグローバル変数として次のように宣言・初期化します。

[配列arrayをグローバル変数として宣言する]

```
int array[5] = {10, 17, 29, 27, 62};
```

arrayの全ての要素の「10進数, 16進数, 8進数」を出力する関数show_digit()を作成してください。

[この関数のプロトタイプ宣言]

```
void show_digit();
```

```
/* arrayの各要素をprintf()で出力する */
```

【課題11-2】

[mainでの処理 (arrayが {10, 17, 29, 27, 62} の場合)]

```
show_digit();
```

[実行結果]

```
0: 10, a, 12  
1: 17, 11, 21  
2: 29, 1d, 35  
3: 27, 1b, 33  
4: 62, 3e, 76
```

【課題11-3】

配列arrayの全ての要素に対して、引数nで割った結果を出力する関数show_division()を作成してください。

[この関数のプロトタイプ宣言]

```
void show_division(int n);
```

```
/* Sample11.cでのキャスト演算子の使い方を参考に作る */
```

【課題11-3】

[配列arrayをグローバル変数として宣言する]

```
/* 課題11-2で作った配列を宣言しておく */
```

[mainでの処理 (arrayが {10, 17, 29, 27, 62} の場合)]

```
show_division(7);
```

[実行結果]

```
1.428571  2.428571  4.142857  3.857143  8.857143
```

【課題11-4】

配列arrayに対して、scanf()によって16進数で入力された値を各要素へ代入する関数input_array16()を作成してください。

[この関数のプロトタイプ宣言]

```
void input_array16();
```

```
/* scanfを使って16進数で入力された値を配列の要素に格納する */
```

【課題11-4】

[配列arrayをグローバル変数として宣言する]

```
/* 課題11-2で作った配列を宣言しておく */
```

[mainでの処理の場合]

```
input_array16();  
show_digit();
```

[実行結果 (16進数と10進数の対応が確認できる)]

```
array[0] > 10      ( ↓ input_array16() 内で16進数で入力 )
```

```
array[1] > 20
```

```
array[2] > 30
```

```
array[3] > 40
```

```
array[4] > 50
```

```
0: 16, 10, 20      ( ↓ show_digit() で出力 )
```

```
1: 32, 20, 40
```

```
2: 48, 30, 60
```

```
3: 64, 40, 100
```

```
4: 80, 50, 120
```

まだ余裕のある人は…【課題11-5】

long long int型とunsigned long long int型が表現できる値の範囲を出力するプログラムを作成してください。以下の要領で作ることができます。

- ▶ unsigned long long int型は、全てのビットが0の時に最小値、全てのビットが1の時に最大値となる。
- ▶ long long int型は、最上位の桁（一番先頭のビット）が0で、それ以外が1の時に最大値となる。最小値は、最大値の2の補数となる。
- ▶ 以上の点に注意して、16進数で値を代入すると、最大値と最小値が格納される。
- ▶ 格納された値をprintf()で10進数で出力すると、値の範囲が確認できる。なお、unsigned long long int型の変換仕様は「%11u（エル・エル・ユー）」、long long int型は「%11d（エル・エル・ディー）」である。

【課題11-5】

[実行結果]

```
unsigned long long int: 0 ~ 18446744073709551615
```

```
long long int: -9223372036854775807 ~ 9223372036854775807
```

小テストについて

小テストの注意点

- 他人の力は借りずに、自分だけでプログラムを作成する。（つまり定期試験と同様）
- 小テスト中は、演習室外へのネットワークアクセスは遮断される。

小テストについて

小テスト中に参照できるもの

- 教科書, 配付資料
- 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
- 小テストでは紙媒体のものは参照可能
- 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