コンピュータサイエンス第２
コンピュータで数（特に実数）を扱うことについて

南出 靖彦

第５回

準備

- Documents フォルダの下の CS2 フォルダに移動
  - `cd Documents/CS2`
- 講義のウェブページから `day5.zip` をダウンロードする.
- Terminal で day5 を CS2 に移動
  - `mv ~/Downloads/day5 ./`
  - `cd day5`

数の表現

(準備 1) 整数の二進数表記
（例） 十進数の 19 は二進数では 10011 となる。

(準備 2) 実数の二進数表記
（例） 十進数の 2.375 は二進数では 10.011 となる。十進数の 0.1 は二進数では循環小数になる。

$$2.375 = 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$0.1\ \text{（十進数）} = 0.00011001100 \cdots \text{（二進数）}$$

数の表現

(準備 2) 実数の二進数表記
（例） 十進数の 2.375 は二進数では 10.011 となる。十進数の 0.1 は二進数では循環小数になる。

$$0.1\ (\text{十進数}) = 0.00011001100\cdots(\text{二進数})$$

1 を 1010(2 進数の 10) で割ると

```
      0.000110011....
    --------------------
1010) 1  0000
         1010
         -----
          1100
          1010
          -------
            10000
             1010
             -------
              1100
```

(準備 3) 正規化された浮動小数点表記 (十進数)

（例）

$$663100000 = 6.631 \times 10^{8}$$

$$0.0003571 = 3.571 \times 10^{-4}$$

- 6.631 と 3.571 を仮数（かすう）と呼ぶ
- 8 と -4（あるいは 10^{8} と 10^{-4}）を指数と呼ぶ
- 正規化：仮数の整数部分を 1 桁にすること

(準備 4) 正規化された浮動小数点表記 (二進数)

（例）

$$11001 = 1.1001 \times 2^4$$

$$0.0101 = 1.01 \times 2^{-2}$$

- 1.1001 と 1.01 を仮数，
- 4 と -2（あるいは 2^4 と 2^{-2}）を指数と呼ぶ。
- ここでは指数は十進数表記する。

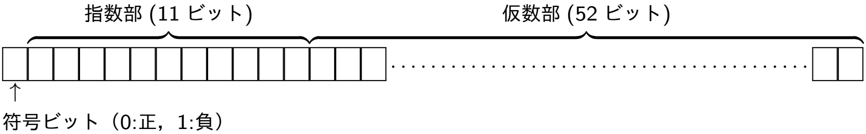
コンピュータ／プログラムでの数値の扱い

整数と実数は異なる扱いになる。

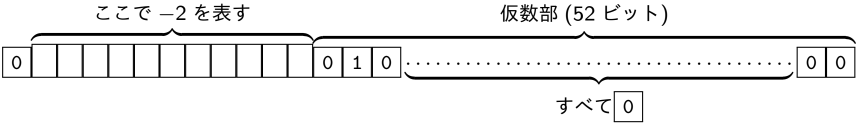
- Python の場合は整数は内部で扱える限りいくらでも大きな値を正しく計算できる
- C 言語などの多くの他の言語ではそうではない
- プログラム中
  - 「10」と書けば整数
  - 「10.0」と書けば実数
- コンピュータ内部での実数の実体は，固定された長さ（多くは 64 ビット？）の 2 進列である。

64 ビットで実数を表す標準的な方法

- 符号部 **1** ビット
- 指数部 **11** ビット
  - 指数部は -1022 から 1023 までの整数を 11 ビットで表現する（詳細は省略）。
- 仮数部 **52** ビット
  - （0 は別扱いにして）0 以外のどんな値の浮動小数点表記も仮数の先頭は常に 1
  - この 1 は覚える必要がないのでそれを取り除いて，正規化された仮数の小数部分 **52** 桁までを仮数部に持つ。



（例）二進数 0.0101（十進数 0.3125）は次のようになる。



丸め

小数部分 52 桁の仮数に収まらない値は，53 桁目以降を端数処理して 52 桁にする（「丸める」と言う）。

丸めの方法はいくつかある．

- 演習室の Python では次の「最近接偶数への丸め」が行なわれている，と推測される（小数点以下を丸めて自然数にする場合に「最も近い偶数」になるのでこう呼ばれる）。

【最近接偶数への丸め】

- 原則は 0 捨 1 入。
- ただし 53 桁目以降が $1000\cdots$（先頭だけ 1 で残りは全部 0）のとき
  - 52 桁目（仮数部の一番下）が 0 なら切り捨て，1 なら切り上げ。

【課題 4 (a)】

プログラム 1 は，次の動作を i の値を増やしながら行うものである。

「実数としての *2*」(「整数としての *2*」ではない）を i 乗した値から，

- *1* を引いたり，
- *1* を足したり，
- *1.5* を足したり，
- *2* を足したりした値を，

表示させる。

これを実行すると数学的な正しい値とは異なる値が表示される。その理由を説明せよ。

【課題４（b)】

Python には実数の定数 `math.e` があり，ネイピア数（自然対数の底）を表す。

- これが数学におけるネイピア数 e の真の値と同じか，異なるか述べよ。
- 異なる場合はどんな理由でどのように異なるかを説明せよ。

なおプログラム２を実行すると `math.e` を小数点以下 15 桁まで表示する。

プログラム２：

```
print(f"ネイピア数 math.e を小数点以下 15 桁まで表示すると\n {math.e:.15f} \n です。")
```

- `15f` の 15 を大きい値に変えてみてください．

【課題４ (c)】

ネイピア数 e は

$$e = 1 + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \cdots + \frac{1}{n!} + \cdots$$

を満たす。
そこで

$$\begin{aligned}
e_0 &= 1 \\
e_1 &= 1 + \tfrac{1}{1} \\
e_2 &= (1 + \tfrac{1}{1}) + \tfrac{1}{1\cdot 2} \\
e_3 &= ((1 + \tfrac{1}{1}) + \tfrac{1}{1\cdot 2}) + \tfrac{1}{1\cdot 2\cdot 3} \\
e_4 &= (((1 + \tfrac{1}{1}) + \tfrac{1}{1\cdot 2}) + \tfrac{1}{1\cdot 2\cdot 3}) + \tfrac{1}{1\cdot 2\cdot 3\cdot 4} \\
&\vdots
\end{aligned}$$

を計算するプログラムを作成・実行して，n を増やしても e_n の値が変わらなくなる n とそのときの e_n の値を求めよ。

さらに足し算の順番を変えて，

$$
\begin{aligned}
e'_0 &= 1 \\
e'_1 &= \tfrac{1}{1} + 1 \\
e'_2 &= (\tfrac{1}{1\cdot 2} + \tfrac{1}{1}) + 1 \\
e'_3 &= ((\tfrac{1}{1\cdot 2\cdot 3} + \tfrac{1}{1\cdot 2}) + \tfrac{1}{1}) + 1 \\
e'_4 &= (((\tfrac{1}{1\cdot 2\cdot 3\cdot 4} + \tfrac{1}{1\cdot 2\cdot 3}) + \tfrac{1}{1\cdot 2}) + \tfrac{1}{1}) + 1 \\
&\vdots
\end{aligned}
$$

を

計算して，n を増やしても e'_n の値が変わらなくなる n とそのときの e'_n の値を求めよ。そしてこれら 2 つの方法で得られた値を比較して論ぜよ。

なお，プログラム 3 は

$$g(n) = (\cdots(((1 + \frac{1}{2}) + \frac{1}{4}) + \frac{1}{8}) + \cdots + \frac{1}{2^n})$$

を計算するものである（この値は $n = 53$ で $g(n) = 2$ に収束することがわかる)。プログラム 3 を少しだけ変更すればこの課題用のプログラムが出来る。

提出方法等：課題４ (a)〜(c)

- 提出期限：1 月 15 日 午前 10:40
  - 午後 9:00 までの提出は採点しますが，大きく減点します．
- 提出するもの
  - レポート (PDF)
  - プログラム (課題 (c))
    - `e1.py`: e_n の計算するプログラム
    - `e2.py`: e'_n の計算するプログラム