

この時間のねらい: ① 変数を使ってみよう(数値編)

② 変数を使ってみよう(文字編)

### ① 数値を管理する変数

数値の変数宣言は以下の 2 つになります。※前回のプリントにもあります。

int 型	整数を格納するための変数を宣言
float 型	浮動小数点数を格納するための変数を宣言

・浮動小数点数とは

float 型にて出てきた言葉ですが、あまり馴染みがないと思います。簡単に言えば、**小数**のことです。もしも、整数を格納するための int 型で宣言した変数に小数を入れてみるとどうなるでしょうか。正解は「**エラーが起き、メッセージが表示される**」です。

・int 型と float 型のメリット・デメリット

#### ◆int 型

##### ●メリット

- ・**計算**が楽なため、コンピュータへの負担が少なく済む。
- ・どのような計算が行われているかを(**人間が**)イメージしやすい。

##### ●デメリット

- ・**刻み幅**が荒いため、細かい処理を行うことが難しい場合がある。
- ・値によっては**バグ**が起きてしまうことが多くある。

#### ◆float 型

##### ●メリット

- ・**刻み幅**が細かいため、細かく処理することができる。
- ・**バグ**が起きるのを減らすことができる。(完全にはなくならない)

##### ●デメリット

- ・**計算**が複雑になるため、コンピュータへの負担が多くなる。
- ・どのような計算が行われているかを(**人間が**)イメージするのが難しい。

使い方は以下ようになります。(int に関しては前回のプリントを参照)

```
float x = 1.1; //浮動小数点数を格納する変数 x を宣言し、x に 1 を代入
```

もしくは、

```
float x;      //浮動小数点数を格納する変数 x を宣言
```

```
    x = 1.1;  //x に 1 を代入
```

## ① 文字を管理する変数

文字の変数宣言は以下の2つになります。※前回のプリントにもあります。

char 型	文字を格納するための変数を宣言
String 型	文字列を格納するための変数を宣言

この2つの宣言は非常に間違えやすいものになります。私が最初に学んだ際にもよく間違えていました。

・ char 型、String 型の違い

### ◆char 型

一文字を格納することができます。例えば、「a」や「あ」などになります。使い方は以下のようにになります。

```
char moji = 'a'; //文字を格納する変数 moji を宣言し、moji に a を代入
```

もしくは、

```
char moji; //文字を格納する変数 moji を宣言
```

```
moji = 'a'; // moji に a を代入
```

### ◆String 型

文字列を格納することができます。例えば、「apple」や「あまいみかん」などになります。使い方は以下のようにになります。

```
String mojiretsu = "apple"; //文字を格納する変数 mojiretsu を宣言、mojiretsu に apple を代入
```

もしくは、

```
String mojiretsu; //文字を格納する変数 mojiretsu を宣言
```

```
mojiretsu = "apple"; // mojiretsu に apple を代入
```

以上のように記述すると、宣言、代入ができる。

条件に沿って変数に自分の名前を格納し、それを使ってコンソールに次のように表示させましょう。

条件：変数名は name にすること。

表示させるもの：私の名前は（名前）です。（名前）の好きなものは（好きなもの）です。

変数を活用すると、もう一度利用したいときに活躍させられるので、積極的に使っていくとよいでしょう。