

この時間のねらい:

- ① 変数の種類を知ろう。
- ② 変数の有用性を知ろう。

数学において、変数と呼ばれるものがあります。その名の通り、変化する数を表します。数なのに変化するのはこれいかにという川柳が読めるほどに嫌われている内容だと感じます。(数学教員としては...)

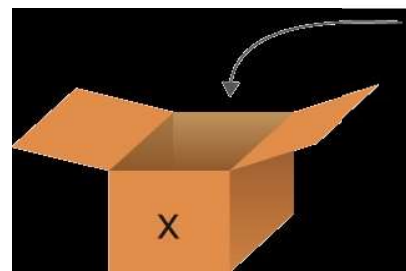
ただ、この内容が使えない場合、「プログラミングを勉強したのに分からないのか?」とか「PG教育の敗北www」などと言われてしまいます。屈辱ですね。

ということで、この内容は非常に重要です。しっかりと理解できるように練習、頭の中で整理をしましょう。まずは、プログラム内での文字の役割を考えましょう。

例えば、数学で嫌というほどであった文字「x」や「y」は文字ですが、最後にその中身は数字であることが分かります。

$2x=4$ を解いて $x=$ \_\_\_\_と出ますよね。

このことから、ここで使用されているxという変数は\_\_\_\_を格納しておく箱(閉じた状態)だとわかります。



2

それでは1回目のコンソール表示を行った時のことを思い出してください。3 と "3" にはある違いがありました。

それは何でしたっけ? どちらも見た感じは数字となっています。データとしての見え方は違いますね。

3は\_\_\_\_データです。"3"は\_\_\_\_データです。基本的に皆さんの良く知る、Excel 等でもこれらはすべて区別しています。また、これら以外にもデータの区別があるため、覚えなくてもこんな感じのものがあつたかなと思い出せるくらいになりましょう。

それではProcessingで利用できる変数宣言です。以下のものが利用できます。

1	int seisuu;	_____
2	float shousuu;	_____
3	boolean ronri;	_____
4	char moji;	_____
5	String mojiretsu;	_____
6	color iro;	_____
7	PImage gazo;	_____
8	PFon font;	_____
9	PShape SVG;	_____

以上の宣言がよく使われるのではと思います。本当はこの他にもありますが、ここでの紹介はこのくらいにしましょう。

これらについての詳細はこれから少しずつ行います。少しずつでも着実にできるようにしましょう。

### ＜練習問題＞

- (1) 整数型変数の ichinen を作成して365を代入し、それを利用してコンソールに「1年は365日です」と表示させよ。
- (2) 小数型変数の enshuritsu を作成して3.14を代入し、それを利用してコンソールに「円周率は3.14です」と表示させよ。

(1) プログラムをそのまま書き写しなさい。	(2) プログラムをそのまま書き写しなさい。
------------------------	------------------------

ここで、プログラミングを行う上で、非常に重要な関数を学びます。右に並んでいる関数です。この関数は Processing で開発を行う場合に必ず使うレベルのものです。  
 setup() はその名の通りこのプログラミングで必要な情報を作成したりするために使われます。例えば \_\_\_\_\_ や \_\_\_\_\_ ということが挙げられます。

```

1 void setup() {
2   _____
3 }
4
5 void draw() {
6   _____
7 }

```

また、draw() は実行する結果をプログラムします。ゲームであれば、ステージの表示やボタン操作についての記述を行う場所になります。  
 とりあえずそうなんだと、思いましょう。これから先に進むうちになんとなく分かれば占めたものです。

＜文字系の型の変数宣言＞

文字データを格納するのは char(\_\_\_\_\_) 型と String (\_\_\_\_\_) 型があります。これらの使い方には圧倒的な違いがあります。それは代入する文字を挟む記号です。

char: \_\_\_\_\_                      String: \_\_\_\_\_

それぞれ、記号が違います。気を付けましょう。

＜練習問題＞

- (1) 1文字型変数 kigou を作成して、その中に 'ア' を入れ、それを利用してコンソールに「(ア)」と表示せよ。
- (2) 文字列型変数 kansha を作成して、その中に "ありがとう" を入れ、それを利用してコンソールに「ありがとう」と表示せよ。

(1) プログラムをそのまま書き写しなさい。	(2) プログラムをそのまま書き写しなさい。
------------------------	------------------------