

この時間のねらい: ① 条件文を使ってパソコンに指示する方法を学ぼう。

### ① 条件文の使い方(2 通りあります)

#### ・if 文の使い方

例えば、敵を全部倒したら GAME CLEAR 画面に移動、倒された場合 GAME OVER 画面に移動など一定の条件を達成した場合に、どのような処理をするかを個別に決める場合、条件文を活用する必要があります。使用 방법은 以下のように記述をします。

```

1  if( /*条件1*/ ) {
2      //条件1の場合の処理
3  } else if( /*条件2*/ ) {
4      //条件2の場合の処理
5  } else {
6      //それ以外の場合の処理
7  }
```

条件を決めるためには前回まで勉強した変数を利用する必要があるため、しっかりと理解できていない人は戻って復習をしておきましょう。

また、条件を **重ねる** 場合は else if とし、引数に条件を追加します。

**それ以外** の場合は else のみで処理を記述します。

#### 条件の記述方法

変数 <= 数値 or 変数	数値 or 変数以下
変数 >= 数値 or 変数	数値 or 変数以上
変数 == 数値 or 変数	等しい
変数 != 数値 or 変数	数値 or 変数以外

変数 < 数値 or 変数	数値 or 変数より小さい
変数 > 数値 or 変数	数値 or 変数より大きい
条件 1    条件 2	条件 1 または条件 2
条件 1 && 条件 2	条件 1 かつ条件 2

上の等号・不等号を **比較演算子** と呼びます。

#### 処理内に記述する一例

変数++;	<b>変数の値</b> を 1 増やす
変数 = 数値 or 変数;	新しい数値 or 変数を代入

以上を活用することで、以前のブロック崩し(ブロック無し)のプログラムを作成しています。もう一度プログラムを見ていきましょう。私のサイトにある第 3 回参考プログラムを Processing にコピペしてください。

どのような処理をしているのか、条件の変更を行うとどのように動きが変わるのかを周りの人たちと話してみましょう。

開発を行う際には、使わないわけにはいかないものであるため、自分で作成できるように積極的に活用して使い方を覚えましょう。

## ・ switch 文の使い方

先ほどの if 文では数値を管理し、その数値の変化により処理方法の変更を行うという手法でした。次に紹介する switch 文は if 文よりもコンパクトに条件分岐をさせることができるものになります。ただ、if 文より若干理解しづらいものになるかもしれません。使い方は以下のようになります。

```
1 int num = /*数値*/;
2
3 switch(num) {
4     case /*数値*/:
5         /*処理内容*/
6         break;
7     case /*数値*/:
8         /*処理内容*/
9         break;
10 }
```

switch の引数 num の値によって処理内容を決める条件文です。例えば、0 の場合は「タイトル画面」、1 の場合は「ゲーム画面」、2 の場合は「リザルト画面」のように、画面の遷移を考えるとときなどに使いやすいです。case の後には **コロン** が付きます。

また、break; の意味としては、**それ以上先に進まない**、というものになります。

switch 文の特徴として case によって分けられているものでも、すべて実行するということが挙げられます。そのため、break; を入れない場合以降の処理がすべて行われてしまうのです。

この文が非常に利用しやすいのは分岐が**多い**ときになります。基本的に switch の引数の値を変更することで処理内容の管理を行うことができるので、処理内容の管理に関しては、こちらのほうが効率良く行えるはずです。

それでは、今回学んだ条件文を活用して、以下の演習問題に取り組みましょう。

演習問題：以下の条件に沿ってプログラムを作成してください。

条件：①文字列型の変数 unsei を宣言する。

②unsei に格納された文字列に“大吉”“吉”“小吉”“凶”が入っているとき、それぞれ以下のように表示されるようにしてください。

unsei に入っている文字列	コンソールに表示するもの
大吉	すごいですね
吉	いいですね
小吉	まあまあいいですね
凶	ちょっとヤバいですね