

比較演算子と論理演算子

論理演算子

演算子	==	<>	>	>=	<	<=
意味	等しい	等しくない	大きい	以上	小さい	以下

論理演算子（複数の比較演算子の結果を組み合わせて使用）

論理演算子	意味	if 文での使用例	例の意味
&&	かつ(and)	if(x >= 60 && x < 70)	x が 60 以上かつ 70 未満
	または(or)	if(x < 10 x > 90)	x が 10 未満または 90 より大きい
!	否定(not)	if(!(x == y))	x と y が等しくない

条件分岐のあるプログラムの書き方（if 文）

<pre>if (論理式) { 条件を満たすときに実行する処理; }</pre>	<pre>if(x > 200) { //もしも、x が 200 よりおおきければ x = 0; //x を 0 にする (左に戻す) }</pre>
--	--

if-else 文：条件以外の場合の処理（論理式を満たさない場合の処理）

<pre>if(論理式) { 条件を満たした場合の処理 } else { 条件を満たさない場合の処理 }</pre>	<pre>if(x <= 200) { //もしも、x 以下ならば ellipse(x, y, 10, 10); //丸を描く } else { //そうでなければ rect(x, y, 10, 10); //長方形を描く }</pre>
--	--

if-else if- else 文:複数条件の条件分岐

<pre>if(条件 1) { 条件 1 を満たした場合の処理; } else if(条件 2){ 条件 1 を満たさずに、条件 2 を満たした場合の処理; } else { どの条件も満たさない場合の処理; }</pre>	<pre>if(score < 50) { //もしも、スコアが 50 未満なら text("Rank : C",10,10); //ランクは C と表示 } else if(score < 80){ //スコアが 80 未満なら text("Rank : B",10,10); //ランクは B と表示 } else { //そうでなければ(スコアが 80 以上なら) text("Rank : A",10,10); //ランクは A と表示 }</pre>
--	--

変数に変数を足す

- 例) `x = x + vx;` //x に vx を足す
 - vx が 1 なら、x は右に 1 ずつ動く
 - vx が 2 なら、x は右に 2 ずつ動く
 - vx が -1 なら、x は左に 1 ずつ動く
- 足す変数の値によって、足される変数の変化量が変わる ⇒ 動きを制御することが出来る

左右に跳ね返る図形

```
int x = 0; //図形の x 座標を管理する変数
int vx = 1; //図形の x 座標方向の移動速度を管理する変数
void setup(){
  size(400, 400);
}
void draw(){
  background(255);
  ellipse(x, 200, 50, 50);
  x = x + vx; //x に vx(移動速度)を足す
  if(x > width){ //もしも、x が右端まで行ったら
    vx = -1; //x の移動速度を-1(左方向)に変える
  } else if(x < 0) {
    vx = 1; //x の移動速度を 1(右方向)に変える
  }
}
```

時間経過で動きを変える

- **frameCount** 変数 : Processing 起動後からのフレーム数を記憶している変数
 - システム変数の 1 つ (width や height のように最初から用意されている変数)
- **frameRate** メソッド : Processing の実行速度を指定することが出来るメソッド
 - `setup()`の中で、**framerate(30);** のようにして呼び出す。
 - ◇ `framerate(30);` は、1 秒間に 30 回の更新間隔で動作させる、の意味
 - ◇ 注意 : フレームレートを指定しても、正確に 1/30 秒で動作するわけではない (ある程度の誤差がある) 。また、高すぎる fps を指定しても PC の処理能力が追い付かないので、指定通りに動かないこともある。

時間経過で動きを変えるプログラム例（3秒経ったら色が変わる図形）

```
int maru_x = 0; //図形の x 座標を管理する変数
int maru_H = 0; //図形の色彩(Hue)を管理する変数
void setup(){
  size(400, 400);
  frameRate(30); //30fps で動作するように指定（1秒間で30回更新される）
  colorMode(HSB,360,100,100,100); //カラーモードをHSBモードに
}
void draw(){
  background(255);
  fill(maru_H, 80, 80, 100);
  ellipse(maru_x, 200, 50, 50);
  maru_x += 1;
  if(maru_x >= width){
    maru_x = 0;
  }
  if(frameCount >= 90){ //3秒経ったら 30fpsの場合、30回動けば1秒 3秒なら 30*3で90フレーム
    maru_H += 1; //図形の色彩を変化させる
  }
}
```