

```
// 整数の値の乱数が欲しいとき、int 関数を使って、整数の値にする
// text pp46
int dice;
dice = int(random(1,7)); //from 1 to 6
println(dice);

// 乱数を使った楕円の描画
void setup(){
  size(400,400);
}

void draw(){
  ellipse(random(width),random(height),
          random(10,20),random(10,20));
}

// 乱数を使った円の描画
float r;

void setup(){
  size(400,400);
}

void draw(){
  r = random(10,20);
  ellipse(random(width),random(height),
          2*r,2*r);
}

// 乱数を使った円の描画
int r;
```

```
void setup(){
  size(400,400);
}
```

```
void draw(){
  r = int(random(10,20));
  println(r);
  ellipse(random(width),random(height),
          2*r,2*r);
}
```

//乱数を発生させる random 関数の利用例

```
float r;
r = random(10);
println(r);
r = random(-10,10);
println(r);
```

//沢山の円をでたらめな位置に描画

```
float x,y;
size(400,200);
background(150);
fill(255);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
```



```
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
x = random(width);
y = random(height);
ellipse(x,y,20,20);
```

```
// 繰り返し処理を使った、でたらめな位置への円の描画
float x,y;
size(400,200);
background(150);
fill(255);
for(int loop=0;loop<10;loop++){
  println(loop);
  x = random(width);
  y = random(height);
  ellipse(x,y,20,20);
}
```

```
// 繰り返し処理を使った、長方形の描画
```

```
float x,y;  
size(400,200);  
background(150);  
noFill();  
for(int loop=0;loop<5;loop++){  
    println(loop);  
    x = random(width);  
    y = random(height);  
    rect(0,0,x,y);  
}
```

```
// 縦線の描画
```

```
size(300, 200);  
background(255);  
stroke(0);
```

```
line(25, 20, 25, 180);  
line(50, 20, 50, 180);  
line(75, 20, 75, 180);  
line(100, 20, 100, 180);  
line(125, 20, 125, 180);  
line(150, 20, 150, 180);  
line(175, 20, 175, 180);  
line(200, 20, 200, 180);  
line(225, 20, 225, 180);  
line(250, 20, 250, 180);
```

```
// カウンタ変数の値を利用した繰り返し処理
```

```
size(300, 200);  
background(255);  
stroke(0);
```

```

for(int loop = 0;loop < 11;loop++){
    println(loop,loop*25 + 25);
    line(loop*25 + 25, 20, loop*25 + 25, 180);
}

// カウンタ変数の値を利用した繰り返し処理
size(200, 405); //200X405 のウィンドウを表示
background(255); // 背景を白色で塗りつぶす
fill(170); // 背景を白色で塗りつぶす

for (int j=0; j<10; j++) { // カウンタ変数 j の値を 0 ~9 まで変えながら
    // 点(30,10+40*j) を頂点とする横 140、縦 20 の矩形を表示する
    rect(30, 10+40*j, 140, 20);
}

// カウンタ変数の値を利用した繰り返し処理
size(400,400);
noFill();
for(int loop = 0;loop < 5;loop++){
    println(loop,75*loop+10);
    ellipse(width/2,height/2,75*loop+10,75*loop+10);
}

// カウンタ変数の値を利用した繰り返し処理
size(400,400);
noFill();
for(int loop = 0;loop < 5;loop++){
    println(loop,75*loop+10,60*loop);
    stroke(60*loop,10,10);
    ellipse(width/2,height/2,75*loop+10,75*loop+10);
}

```

```
// 複数の繰り返し処理
size(400, 200);
background(0);
smooth();
fill(255);
for (int x=0; x<11; x++) { // この繰り返しでは、変数 x がカウンタ変数
    ellipse(40*x, 0, 40, 40);
}
for (int y=0; y<6; y++) { // この繰り返しでは、変数 y がカウンタ変数
    ellipse(0, 40*y, 40, 40);
}
```

```
// 繰り返し処理の入れ子
size(400, 200);
background(0);
smooth();
fill(255);
for (int x=0; x<11; x++) {
// 一番外側の繰り返し処理のカウンタ変数は x
    for (int y=0; y<6; y++) { // この繰り返し処理のカウンタ変数 y
        println(x,y);
        ellipse(40*x, 40*y, 40, 40);
    }
}
```

```
// 繰り返し処理の入れ子、インデント無しバージョン
size(400,200);
background(0);
fill(255);
stroke(100);
for(int j = 0;j < 17;j++){
int y = 20 + 10*j;
```

```
for(int i = 0;i < 37;i++){  
  int x = 20+10*i;  
  ellipse(x,y,4,4);  
  line(x,y,width/2,height/2);  
}  
}
```

```
// 繰り返し処理の入れ子、インデント無しバージョン  
size(400,200);  
background(0);  
fill(255);  
stroke(100);  
for (int j = 0; j < 17; j++) {  
  int y = 20 + 10*j;  
  for (int i = 0; i < 37; i++) {  
    int x = 20+10*i;  
    ellipse(x, y, 4, 4);  
    line(x, y, width/2, height/2);  
  }  
}
```

```
//rectMode(CORNER) 【デフォルト】 で四角形描画  
void setup(){  
  size(400,400);  
}  
  
void draw(){  
  background(255);  
  
  rect(mouseX,mouseY,40,40);  
}
```



```
//rectMode(CORNER) 【デフォルト】 で四角形描画
```

```
void setup(){  
  size(400,400);  
}
```

```
void draw(){  
  background(255);  
  
  rect(mouseX-20,mouseY-20,40,40);  
}
```

```
//rectMode(CENTER)で四角形描画
```

```
void setup(){  
  size(400,400);  
}
```

```
void draw(){  
  background(255);  
  rectMode(CENTER);  
  rect(mouseX,mouseY,40,40);  
}
```

```
//rectMode(CENTER)で四角形描画、マウスカーソルの中心に長方形をおきた  
い
```

```
void setup() {  
  size(400, 400);  
  rectMode(CENTER);  
}
```

```
void draw() {  
  background(255);  
  fill(175);
```

```
rect(mouseX, mouseY, 100, 60);
fill(0);
rect(mouseX-30, mouseY-35, 20, 10);
rect(mouseX+30, mouseY-35, 20, 10);
rect(mouseX-30, mouseY+35, 20, 10);
rect(mouseX+30, mouseY+35, 20, 10);
}
```

```
// rectMode(CORNERS)による
```

```
void setup() {
  size(400, 300);
  rectMode(CORNERS);
  noStroke();
  fill(175);
}
void draw() {
  background(255);
  rect(mouseX, mouseY, width-mouseX, height-mouseY);
}
```

```
// rectMode(CORNERS)を使わずに上と同じ動作をさせる
```

```
void setup() {
  size(400, 300);
  // rectMode(CORNERS);
  noStroke();
  fill(175);
}
void draw() {
  background(255);
  rect(mouseX, mouseY, width-2*mouseX, height-2*mouseY);
}
```