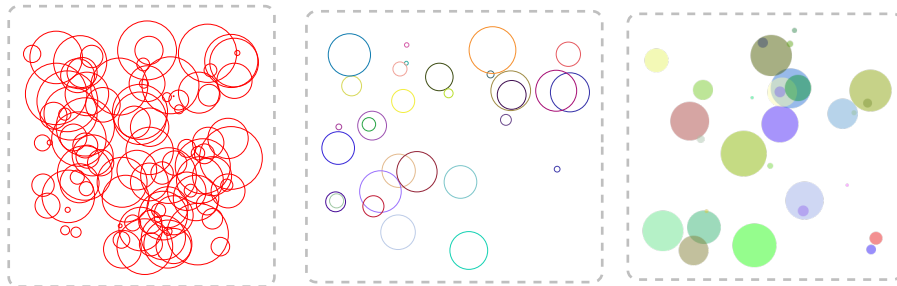


### 3. Wolfram言語でアートプログラミングに挑戦しよう！

ランダム関数を使うと、同じプログラムコードでも、プログラムを実行するたびに違った結果が出てきます。この演習では、円を描く関数とランダム関数を使って、大小さまざまな円を組み合わせたアート作品を、Wolfram言語のプログラミングで作っていきましょう。

手順は以下の通りです。

1. Circle関数で円を描こう
2. 中心座標と半径にランダム関数を指定して円を描こう
3. 複数のランダムな円をまとめて描こう
4. 円に色をつけて完成！
5. 応用：円以外のグラフィックスや、乱数を使わないプログラムの例



図P3-1 アートプログラミングの作品例

#### (1)Circle関数で円を描こう

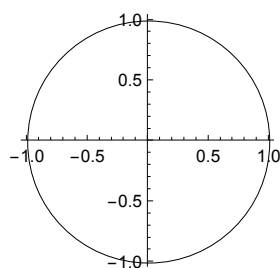
まず、Circle関数を使って円を描いてみましょう。

`Graphics[Circle[{0,0},1],Axes→True]` は、中心座標{0,0}で半径1の円を座標軸と共に描画する、というプログラムです。

(→矢印は、マイナス記号と>記号を続けて入力すると自動的に→に変わります。キーボードの矢印キーではありませんの注意してください。)

In[1]:= `Graphics[Circle[{0, 0}, 1], Axes → True]`  
[グラフ...] [円] [軸...] [真]

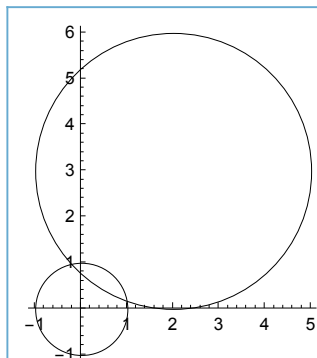
Out[1]=



複数の円を同時に描きたいときは、次のようにCircleのプログラムを{}で括って、円のリストを作ります。リストとは、{}で複数の要素をひとつにまとめたものです。  
この例では、中心(0,0)で半径1の円と、中心(2,3)で半径3の円を描いてみました。

```
In[2]:= Graphics[{Circle[{0, 0}, 1], Circle[{2, 3}, 3]}, Axes → True]
```

Out[2]=



練習問題：中心座標や半径の大きさを変えて、プログラムを実行（**SHIFT+ENTER**）してみましょう。

## (2) 中心座標と半径にランダム関数を指定して円を描こう

次に、RandomReal関数を組み合わせて、ランダムに選ばれた中心座標と半径で円を描いていきます。（Realは実数という意味です）

まずは、RandomReal[10]として、0~10の範囲の実数の乱数を生成してみましょう。プログラムを実行するたびに数値が変わります。何度も実行して数値が変わるか試してみましょう。

```
In[3]:= RandomReal[10]
```

Out[3]= 4.85823

RandomReal[10,2]とすると、0~10の範囲の数値の中から、2つの乱数がリスト{数値1, 数値2}という形で出力されます。これも、実行するたびに数値が変わります。試してみましょう。

```
In[4]:= RandomReal[10, 2]
```

Out[4]= {6.4244, 0.345992}

さあ、これらをCircle関数の引数として使用すれば、座標と半径をランダムに決めるプログラムになります。実際に、どんな値になっているかを、以下のプログラムを実行して確認してみましょう。

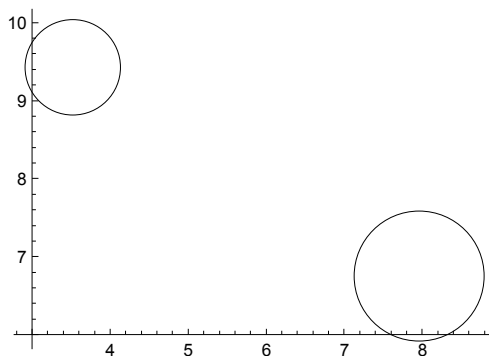
```
In[5]:= Circle[RandomReal[10, 2], RandomReal[1]]
          |円      |実数乱数      |実数乱数
```

```
Out[5]= Circle[{4.07612, 3.10583}, 0.679111]
```

それでは、上記(1)の最後のプログラムを少し変更して、ランダム関数を使ったプログラムにしてみます。以下は、中心座標が{-10,-10}から{10,10}の範囲、半径が0~1の範囲の円を2つ描くプログラムです。プログラムを何度か実行し、実行するたびに描画される円が変わることを確認しましょう。

```
In[6]:= Graphics[
          |グラフィックス
          {Circle[RandomReal[10, 2], RandomReal[1]],
            |円      |実数乱数      |実数乱数
            Circle[RandomReal[10, 2], RandomReal[1]]},
          |円      |実数乱数      |実数乱数
          Axes -> True]
          |軸...   |真
```

```
Out[6]=
```



練習問題：上のプログラムは、中心座標が{-10,-10}から{10,10}の範囲、半径が0~1の範囲です。この範囲を変えるにはどの数字を変えたらいいか考えて試してみましょう。

### (3) 複数のランダムな円をまとめて描こう

指示した内容を繰り返し何回でも実行することは、プログラムの得意技です。  
まずは手始めに、次のプログラムを実行してみましょう。abcが5個のリストが生成されます。

```
In[7]:= Table[abc, {5}]
      リストを作成
```

```
Out[7]= {abc, abc, abc, abc, abc}
```

次に、abcをRandomReal[10]に変えて実行してみましょう。0~10の範囲でランダムに選ばれた5個の数値のリストが生成されます。

```
In[8]:= Table[RandomReal[10], {5}]
      リ... 実数乱数
```

```
Out[8]= {6.30551, 8.57317, 2.38843, 7.10729, 6.03105}
```

このTableのプログラムを、円を複数作るプログラムに変更していきます。  
ちょっと長くなりますが、次のプログラムでは、(2)でやったランダムな中心座標とランダムな半径で、5個の円を作ります。

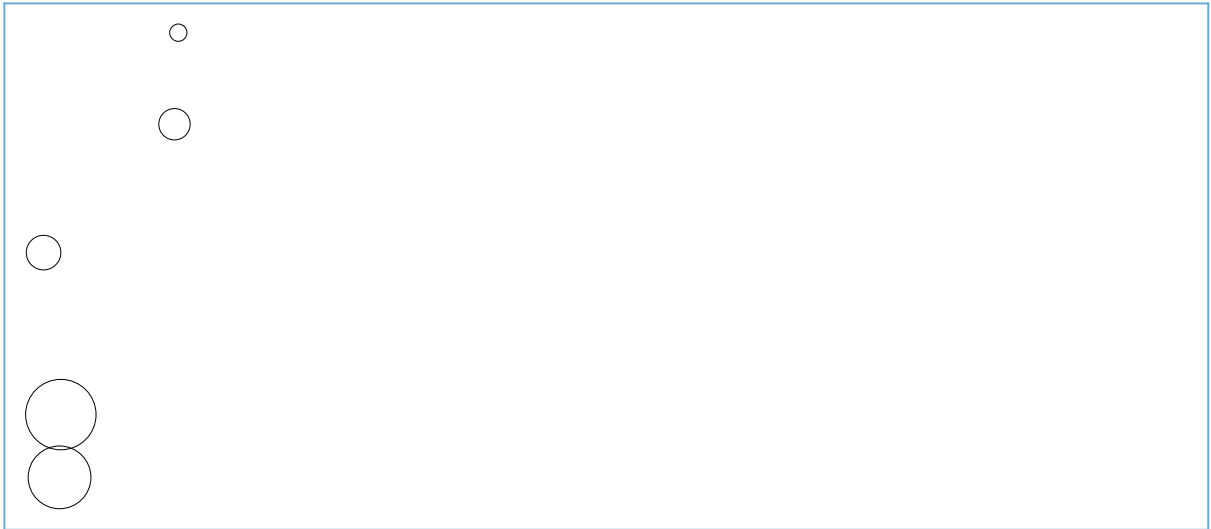
```
In[9]:= Table[Circle[RandomReal[10, 2], RandomReal[1]], {5}]
      リ... 円 実数乱数 実数乱数
```

```
Out[9]= {Circle[{7.12451, 5.8954}, 0.26668],
        Circle[{3.09116, 2.05591}, 0.320598], Circle[{6.73172, 1.98165}, 0.335822],
        Circle[{5.77738, 7.36664}, 0.925251], Circle[{5.58767, 0.491387}, 0.142424]}
```

これを描画すると、ランダムな位置と大きさと異なる5つの円が描けそうですね。  
上のプログラムをGraphics[]の中にいれて、グラフィックスとして描画しましょう。

```
In[10]:= Graphics[Table[Circle[RandomReal[10, 2], RandomReal[1]], {5}]]
```

Out[10]=



円の増やして、100個描いてみます。よりアートっぽくなりますね！

```
In[11]:= Graphics[Table[Circle[RandomReal[10, 2], RandomReal[1]], {100}]]
```

Out[11]=



練習問題：円の個数を変えたり、中心座標の範囲や半径の大きさの範囲を変えて、プログラムを実行してみましょう。

#### (4) 円に色をつけて完成！

さあ、いよいよ、きれいな色をつけて、素敵なグラフィックスにしていきましょう。

Circleの前に色の関数Redを追加し、Red, Circle[RandomReal[10,2],RandomReal[1]]を、{}で括ると、赤色で、半径が0~1の値をランダムに取る円が5個、描画されます。

```
In[12]:= Graphics[Table[{Red, Circle[RandomReal[10, 2], RandomReal[1]]}, {5}]]
```

グラフィックス リスト 赤 円 実数乱数 実数乱数

Out[12]=



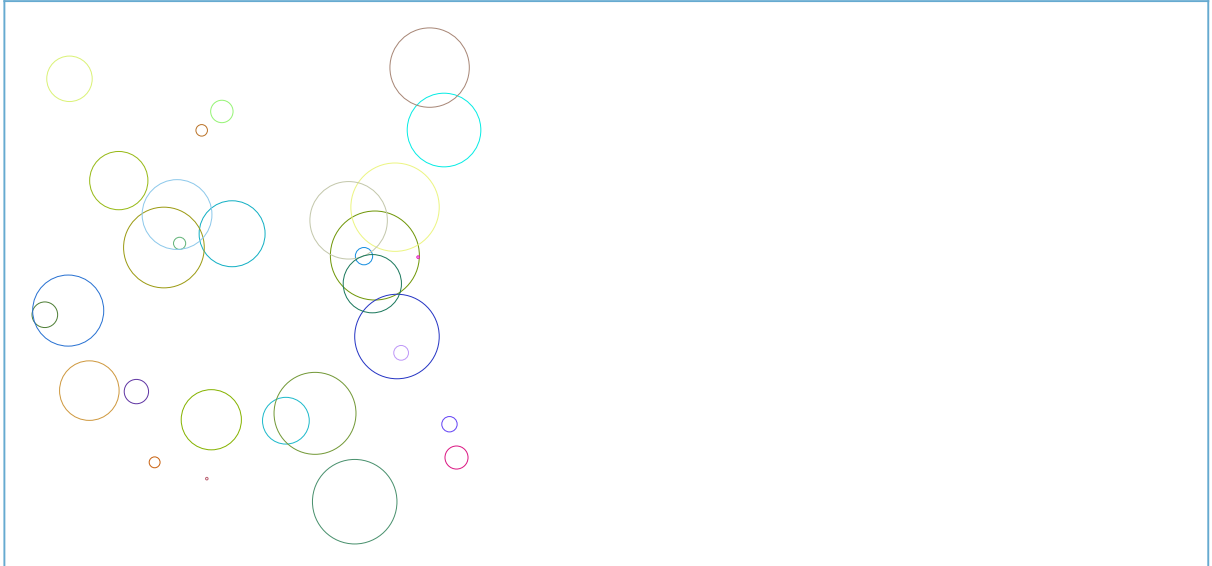
RandomColor関数を使えば、色もランダムにつけられます。

RedをRandomColor[]に変更すると、以下のようなカラフルな作品になります。

```
In[13]:= Graphics[
  Table[{RandomColor[], Circle[RandomReal[10, 2], RandomReal[1]]}, {30}]]
```

グラフィックス リスト ランダムな色 円 実数乱数 実数乱数

Out[13]=

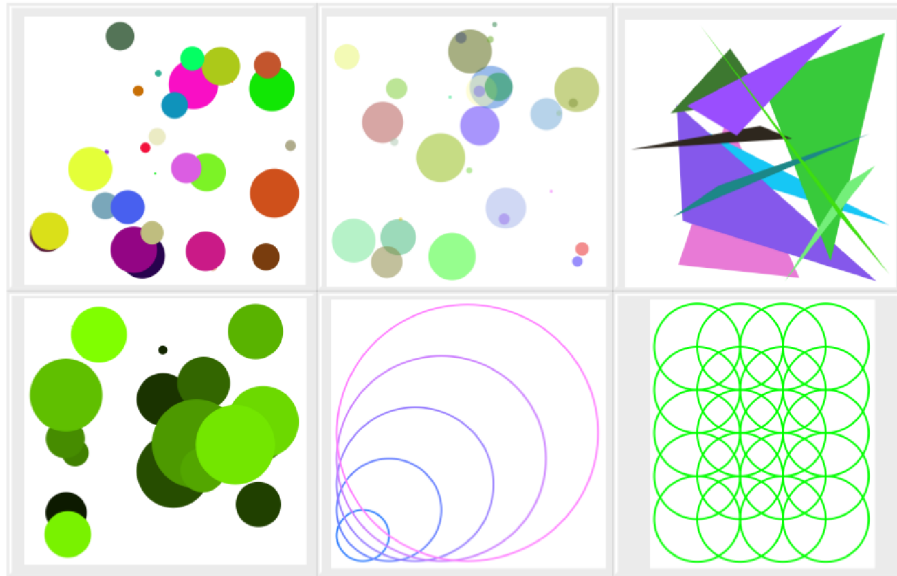


練習問題：円の個数や色などを自由に変えてみて、お気に入りの作品を作りましょう。第3章で学んだ RGBColor関数や、実践編2で紹介した色関数を使うと色を指定できます。ランダムな関数を使うと、同じプログラムでも実行するたびに違った結果になります。

## 応用：他の形のアート作品にチャレンジしてみよう

円を塗りつぶしたり、円ではなく三角形にしたり、ランダムではなくパラメータを規則的に動したりして、さまざまなグラフィックスが表現できます。以下にいろいろな例を載せたので、参考にしてチャレンジしてみましょう。

Wolfram言語の関数は、オンラインドキュメントで調べることができます。新しい関数は自分でドキュメントを調べて、オリジナルのアート作品にチャレンジしてみましょう。(\*1)



図P3-2 アートプログラミングの応用作品の例(\*2)

(\*1) オンラインドキュメントはコラムB(Pxx)で紹介しています。

(\*2) 応用例のプログラムは、本書サポートページで紹介しています。