

この時間のねらい: ① 処理の種類を知ろう。
② 反復処理・ループ処理について学ぼう。

前回話した通り、プログラミングには3種類の処理があります。これを 3大処理 と言います。今回はそれらを見ていきましょう。
この処理に分類されているものは以下の3つです。

1. 順次処理 又は 逐次処理 2. 反復処理 又は ループ処理 3. 条件分岐

これらを、細かめに話していきましょう。

1番の処理では、前回行った変数の宣言が分かりやすいでしょう。右のプログラムの場合では、最初に整数型変数の"a"を宣言し、次に変数"a"へ代入、最後にそれをコンソール表示という順番だと分かるはずですよ。

```
1 int a;  
2  
3 void setup() {  
4   a = 1;  
5 }  
6  
7 void draw() {  
8   print(a);  
9 }
```

まあ、このプログラムを実行すると、ただ延々とコンソールへ1を描画させるという呪いをパソコンにかけるだけです。なんと、生産性のないプログラムなんだ。

なお、この順番を入れ替えると当然エラーが起きます。気を付けましょう。

※エラーが起きた際は赤字部分の文章を検索することで、内容が大まかにわかることが多いです。

2番の処理は、同じ処理を何度も行うことができる処理です。例えば、まったく同じ円を50個描こうとしたとき、手作業となると非常に面倒ですね。これを、勝手にやらせるということが、プログラミングでできてほしいことです。もちろん使えるようになってほしい処理です。(前回の変数同様、知らないだけで、「あっ(察し)」のような扱いになるほどの大問題です...)

ちなみに、今回の時間で、使い方をマスターしてもらいます。(圧)

3番の処理は、条件を満たした際に、行う処理を変更する処理です。例えば、リズムゲームというのは条件分岐の宝庫となっております。右の画像はnamcoで開発されている、「太鼓の達人」のゲーム画面です。皆さんもわかっているかと思いますが、このゲームでは、太鼓の面と縁を譜面通りのタイミングで叩くと点数が加算されていくというものです。



非常にわかりやすいのに、なかなか面白いゲームですよ。

また、このプレイ画面では普通と難しいをプレイしています。これらは全て、プレイヤーが選んだ、叩いたら、それぞれに対応した処理を行う。

これが処理の内容になります。これは次回マスターしてください。

今回から先の内容は、プログラミングの中でなかなか難しい内容になりますので、とにかく、理解できなければ聞く、難しくとも試してみるなど、どんどん使い方に慣れていってできるようにしてください。

ちなみに、余談ですが、私も「太鼓の達人」で遊びます。(これ以外のゲームを遊ぶと頭の中でプログラムを構築してしまう病に侵されています。)

皆さんは()に書いてあるような病にかからない程度にプログラミングを学びましょう。

＜練習問題＞

(1) 円を等間隔に3つ並べて表示させなさい。

(2) 四角を等間隔に3つ並べて表示させなさい。

(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。
(1) 改良版を書き写しなさい。	(2) 改良版を書き写しなさい。

さて、今までの知識からプログラムを作成すると、同じ関数を何回も使用しなければいけません。ただ、この中にも「いや、俺は コピペ を使うから大丈夫です!」と言う人もいるんじゃないでしょうか。私はその人に対して、「それじゃ、同じ図形を3000個出してみよ。」と言いますね。どうですか? 嫌ですよ。なので、右の関数を使います。もしくは、その下の関数を使うと良いでしょう。

この2つの関数が局所的に 反復処理 を行うための関数になります。どちらも表し方が違うだけで、やっていることは同じです。

また、反復処理を行う場合は、i, j という変数を宣言することが多いです。これらのうち、どちらか一方の表し方を完璧にできるようにしてください。

たくさんの図形を描く場合、人によっては、ランダムな値を指定したい場合があります。それを実現させるためには、中心位置、座標を指定する引数を以下の関数を利用して書きましょう。

`random(最小値, 最大値);`

日本語でも馴染み深い(だろう?)ランダムな値を出すための関数です。一応、色の指定にも利用することができますが、工夫しないと、昔のポケモンであった事件と同じことになってしまうので、注意しましょう。(ポリゴン ショックのことですね。)

<練習問題>

- (1) 縦に5個、横に5個並べて合計25個の円を表示させなさい。
- (2) 画面内に50個の円を表示させなさい。(表示位置はランダムにする。)

(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。
------------------------	------------------------