

# Roteiro de atividade 1

Esse roteiro de atividade se baseia no livro **The sparkfun guide to processing** e foi escrito por Camila Oliveira e revisado pelo professor Carlos Shigue.

## 👉 Termos úteis

---

- **IDE**(integrated development environment): já como o nome diz, é um ambiente integrado, onde permite que você escreva o seu código(instruções que vão fazer o computador executar uma tarefa), e também execute essas instruções como um programa de computador;
- **Sketch**: em português, esboço, é como chamaremos o arquivo gerado pelo processing com o seu código
- **Função**: função é um conjunto de instruções que realizam determinada tarefa;
  - Função estrutural: é uma função que divide o código em alguns blocos principais, e cada bloco contém uma série de comandos parecidos entre si com relação a finalidade desses comandos.

Exemplo: Uma receita é separada em duas partes principais: os ingredientes e o modo de preparo, a função setup() e a draw() são como essa divisão de dois grandes grupos.

setup(): é uma função como a parte dos ingredientes numa receita. Ela apenas executa seu interior uma vez no início do código quando ele é rodado.

draw(): é a parte da ação da receita. É uma função loop, que repete seu interior 60 vezes por segundo até que você mande-a parar.

Essas são funções essenciais no seu código, e sem elas você não consegue fazer uma receita. Você pode não colocar nada dentro do draw() e colocar tudo na função setup(), mas isso significa que o seu conteúdo apenas rodará uma vez.

- **Sintaxe**: é o nome que se dá as regras de escrita que se aplicam aos códigos de programação. Todos os códigos seguem regras para que o computador consiga processar tudo que você escreveu de forma correta.
- **Código base**: a seguir, um exemplo de código base que vocês utilização muito:

```
void setup()
{
  //código setup é colocado aqui e ele roda apenas uma vez
}
void draw()
{
  //código é colocado aqui e roda várias vezes em um looping
}
```

# 👉 Atividade 1-Arte com Pixel

Nessa atividade você passara um desenho seu de baixa resolução para o computador via código processing.

- 1º passo: Desenhe um quadrado composto de 20x20 quadradinhos menores, ou se você já tiver alguma facilidade faça algum outro desenho utilizando quadradinhos menores. Veja o exemplo abaixo:

Fonte: (RUNBERG,2015)

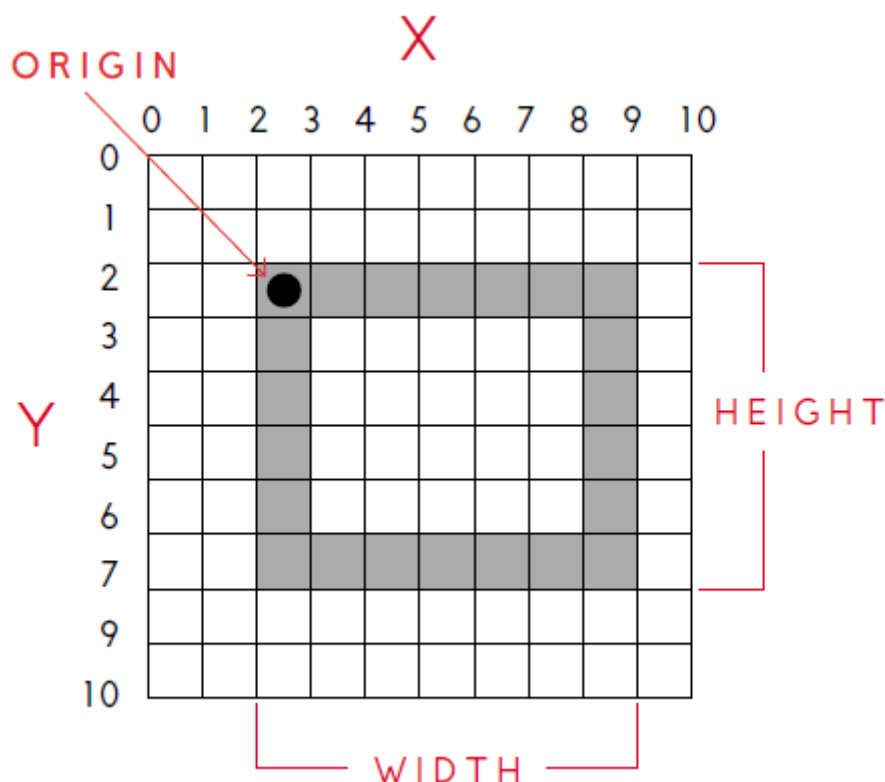


Figura 1 Plano cartesiano e como o Processing interpreta a localização da figura.

- 2º passo: comece seu código escrevendo as duas funções principais: *setup()* e *draw()*. Dentro da função *setup*, entre {}, defina o tamanho do quadrado que você desenhará com a função *size(x,y)*;
- 3º passo: dentro da função looping *draw()*, entre {}, escreva a função *rect()*, e para defini-la melhor, coloque os parâmetros, que são explícitos abaixo:

*rect(distância em pixels da origem-eixo x, distância em pixels da origem-eixo y, largura do quadrado em pixel, altura do quadrado em pixel);*

exemplo: *rect(100,100,20,30);*

Após esses três primeiros passos aperte **run** para testar sua obra de arte!

- 4º passo: agora vamos colocar cor no desenho! A função *fill()* aceita 3 parâmetros que são relativos a um código RGB.

Clique em **ferramentas** e depois em **selector de cor** para criar uma cor e gerar um código RGB.

Exemplo:

```
fill(0,255,0);
```

**Observação: todas as funções precisam terminar com ponto e vírgula ( ; ), essa pontuação faz parte da sintaxe e sem ela o computador não consegue entender o seu código.**

- 5º passo: crie vários quadrados diferentes, com tamanhos diferentes e posições variadas, e então brinque com a ordem que você coloca a função *fill(R,G,B)*, e você vai ver que dependendo da ordem, o resultado muda;
- 6º passo: a linha que define o quadrado que você desenhou recebe o nome de stroke. Se você utilizar a função *noStroke()* no topo do corpo do seu código, ele removerá o traçado de todas as formas que você está criando, mas basta adicionar a função *stroke(R,G,B)* para você adicionar a linha de contorno de volta a uma forma específica com um parâmetro RGB para a cor da linha de contorno;
- 7º passo: agora que você já está familiarizado com as várias funções e possibilidades, crie um desenho complexo usando processing.

## Referências

RUNBERG, Derek. *The SparkFun guide to Processing : create interactive art with code*. San Francisco, CA: No Starch Press, Inc, 2015. Print.