Olá, Tartaruguinhas!

O gráfico de tartaruga é baseado em uma metáfora bem simples. Imagine que você tem uma tartaruga que entende inglês. Você pode dar comandos simples à tartaruga tais como “siga em frente” ou “vire à direita”. A medida que a tartaruga se move, se o seu rabo está virado para baixo, tocando o chão, ela desenha uma linha (deixa um rastro) ao se mover. Se você mandar a tartaruga levantar o rabo ela ainda pode se mover mas não vai deixar nenhum rastro.

**Turtles Robots (Robôs Tartaruga)**

Uma classe de robôs educacionais

● Ciência da computação, engenharias, eletrônica etc Idealizados em 1940

Podem possuir sensores

Obedecem a uma sequência de comandos

**Turtles Graphics (Gráficos de**

**tartaruga)**

Baseado na ideia das tartarugas robôs

Inserido na linguagem LOGO por Seymour Papert

A tela (display) é um plano cartesiano

A sequência de comandos movimentam o cursor

● O cursor pode ser qualquer coisa, geralmente é uma tartaruga

Os cursores podem, ou não, pintar sua trajetória

**E o Python?**

Python possui o módulo Turtle

Fácil de usar

Basta import turtle para começar a brincar

**Porque**

**aprender a programar?**

Saber programar ajuda

● Entendimento cognitivo e na formação do pensamento lógico

● Aprendizado da matemática e do inglês

● Socialização

○ Pode-se mostrar a “coisa legal que você fez” para seus conhecidos

○ Programar em grupo é mais simples, fácil e divertido ● Movimenta a curiosidade

● Ensina a ser organizado

● Ensina limites e regras

● Ensina a aprender

**Comandos básicos**

Iniciando a Brincadeira Criando uma área para brincar ● turtle.Screen() 

Cadê a tartaruga?

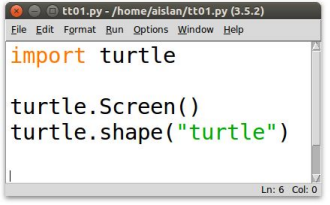
● turtle.showturtle()

● turtle.st()

E se eu quiser escondê-la?

● turtle.hideturtle()

● turtle.ht()

Mas Isso Não É Uma Tartaruga!!! Mudando o formato da tartaruga 

● turtle.shape([valor])

○ “arrow”

○ “turtle”

○ “circle”

○ “square”

○ “triangle”

○ “classic”

**Movimentando a tartaruga**

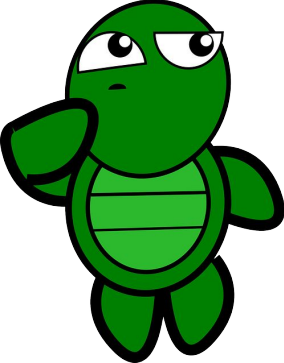
Para frente

● turtle.forward(valor) ● turtle.fd(valor)

Para trás

● turtle.backward(valor) ● turtle.bk(valor)

● turtle.back(valor)

Girando a tartaruga Trabalhando com ângulos 

● Graus (padrão)

○ turtle.degrees()

● Radianos

○ turtle.radians()

Girando para a direita

● turtle.right(valor) ou turtle.rt(valor) Girando para a esquerda

● turtle.left(valor) ou turtle.lt(valor)

Indo de ponto a ponto

turtle.goto(valor)

turtle.setpos(valor)

turtle.setposition(valor)

Valor é um par de coordenadas cartesianas tal qual 0,0

**Riscar ou não riscar?**

Para riscar, abaixe a caneta

● turtle.pendown()

● turtle.pd()

● turtle.down()

Para não riscar, levante a caneta

● turtle.penup()

● turtle.pu()

● turtle.up()

**Largura do Traço**

● turtle.pensize(valor)

● turtle.width(valor)

Valor deve ser um número positivo

Cores Cor da Caneta

● turtle.pencolor([valor])

Cor do preenchimento

● turtle.fillcolor([valor])

Valor pode ser

● Uma string como “red”, “green” ou

“#336699”

● Uma tupla rgb (100, 200, 150)

Mais alguns comandos interessantes Limpando a tela

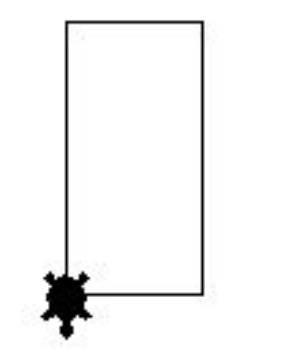
● turtle.clear()

Começando tudo de novo

● turtle.reset()

Levando a tartaruga para (0, 0)

● turtle.home()

Exercício 

Utilize os comandos aprendidos e desenhe um retângulo de largura 50 e altura 100.

Exercício

Desenhe um quadrado de lado 100 utilizando laço de repetição.

Exercício

Desenhe um triângulo de lado 150 e ângulos de 60º

**Desafio** Desenhe um:

● Triângulo equilátero: possui os três lados com

medidas iguais

● Triângulo isósceles: possui dois lados com

medidas iguais

● Triângulo escaleno: possui os três lados com

medidas diferentes

● Triângulo acutângulo: possui todos os ângulos

com medidas menores que 90º

● Triângulo retângulo: possui um ângulo com

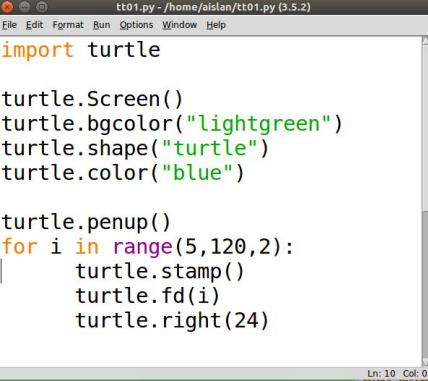
medida igual a 90º

● Triângulo obtusângulo: possui um ângulo

obtuso, maior que 90º

**Desafio**

Desenhe uma casa

**Desafio** 

Teste o código do lado e explique o que acontece

Obrigado!