Testes LFA

Primeiro de tudo, todas as variáveis são globais, primeiro estarei dando alguns exemplos de como colocar valor em uma variável:

1 - a=2; O jeito mais comum, colocar diretamente

2 - Depois de declarar uma função do tipo, def funcao(a,b,c,d) if a<b then c else d; na hora de chamar a mesma, usa-se funcao(30,20,10,40); e então os valores irão substituir os valores na mesma ordem: a= 30;b=20;c=10;d=40;

Estas são as formas de colocar valor em uma variável, a questão é que como são globais você não apenas mudará o valor delas dentro da função como em toda parte;

Chamada de função:

1 - É possível chamar uma função não declarada, porém ela irá retorna um erro com o nome da função;

2 - Não é preciso passar os parâmetros pela função, porém caso sua função necessite de alguma variável, por exemplo, def funcao(a,b,c) c red yellow fill { repeat a {l b f c}};, caso você não passe os parâmetros, os valores ficaram como string e na hora de chamar a função dará um erro, caso não possa ser uma string, caso contrário os valores serão feitos como strings mesmo, porém quando já declarou as variáveis anteriormente, a, b e c, os valores serão substituídos, pelos valores dos mesmos.

Declaração de função:

1 - A declaração de função é simples, def nome(parametros) corpo; , sendo o nome qualquer um, porém não é possível nomes com números, parâmetros pode ou não ter, e o corpo pode ser uma instrução, que faz o desenho, um átomo, um if, ou um conj, “2<3”;

2 - Caso crie uma função que use variáveis nela, e não recebe nenhum parâmetro, def funçao() a b c, se chamar a função passando parâmetros,funcao(a,b,c) para tentar substituir nos lugares das variáveis, estes parâmetros serão inúteis para tal função. Só será lido algum valor caso as mesmas variáveis já tenham sido declaradas;

Declaração de If:

1 - O if só pode ser usado com maior, menor, etc., do tipo que tem 2 tipos de if’s, if 2<3 then a else b; if 2 then a else b; , sendo o else não obrigatório, porém, quando não tem else e o caso cai em else, ele retorna uma string vazia;

Instruction:

A parte mais “importante” do código, já que é ela que faz o objetivo da DSL, desenhar no Turtle, a várias formas de usar o programa, mas primeiramente vamos falar sobre cada parte da instrução separadamente.

Change_Color:

Como o nome diz é nela que muda a cor da “tartaruga”, podendo passar dois parâmetros, o primeiro é a cor da linha, borda, o segundo é a cor que vai pintar o desenho feito, para chamar a função tem que usar a letra “c”, por exemplo, c red, pinta a linha de vermelho, c red yellow, pinta a linha de vermelho, e assim que acabar o desenho, colore de amarelo, caso o desenho não seja uma linha apenas. Por default a cor é preta;

Fill:

A instrução Fill é para preencher o desenho com a cor da tartaruga, caso passe a segunda cor, será ela a cor preenchida, por default é primeira cor;

Repeat:

Repete uma certa instrução N vezes, como já disse, fiz o Repeat apenas para ser usado dentro de uma instrução.

Reset:

Esta instrução retorna o desenho e a cor para o valor inicial, a posição inicial e o desenho vazio.

Movement:

A instrução que faz a tartaruga se mover, tem 4 jeitos que ela pode se mover, para frente, “f”, para trás ,”b”, girar para direita, “r”, girar para esquerda, “l”.

Como usar as instruções:

Começarei falando do reset, não é possível usar o reset sozinho, “reset”, porque ele vai achar que é uma variável, para usar o reset apenas tem que ter alguma instrução, antes ou depois, por exemplo: reset c red; reset f 200; (Vai resetar, colocar a cor vermelha, depois resetar e andar para frente 200 com a cor preta *default). Pode ser até mesmo o próprio reset, “reset reset;”

Todo movement tem que ter um valor numérico na frente, é possível colocar uma “string” que depois se tornará uma variável com um valor numérico, caso não usar valor numérico retornará um erro dizendo que não é int.

O repeat apenas precisa usar ele e a instrução que quer que repita, porém por algum erro, o repeat apenas funciona direito com mais de um movimento; Não é possível usar mais de um repeat dentro de outro ou numa mesma instrução, caso tente, apenas o último repeat será feito.

A change color só funciona com cor escrita em inglês, c red, c blue, c purple, c brown etc.

O Fill é responsável por colorir o desenho ao final, por isso só é possível usar ele no começo, de duas maneiras, a primeira é envolver ele em tudo, ou depois de change_color, em outras posições ele irá parar o desenho, na hora que ele for lido;

Alguns exemplos de teste para usar durante o programa:

repeat 3 {f100 l90} -> Faz um triângulo

repeat 4 {f100 l90} -> Faz um quadrado

repeat 5{f100 l72} -> Faz um pentágono

repeat 6{f100 l72} -> Faz um hexágono

Declare uma função : def desenha(a,b,c) repeat a { f b r c};

Usa a função declarada: desenha(3,100,90); ->triângulo

Faça vários outros desenhos utilizando a mesma função, sempre bom usar o reset reset;* para apagar o desenho anterior; *como já foi dito apenas um “reset” não funciona;

Uma estrela por exemplo: desenha(9,100,200) ou desenha(36,200,170)

Faça desenhos seguidos, como círculos dentro de outro círculo:
desenha(36,100,10) e vai diminuindo o “f”;

Uma espiral desenha(18,20,5) e vai aumentado o R;*nesse caso quando R é 20 ele faz um círculo completo;

Lembra que o repeat não funciona duas vezes? Não é que não funciona, apenas não faz uma repetição dentro da outra do jeito certo, porém é possível fazer alguns desenhos estranhos com isso, tente usar esse movimento nove vezes:

repeat 10 { f 200 l 170 repeat 10 { r 170 f 200 }} -> repita esse passo 9 vezes, usar Fill em alguns exemplos, como este, não é muito agradável;

Eu acabei esquecendo de fazer a cor da instrução aceitar variáveis: Para mudar isso troque a função leCores por esta:

```
def leCores(lista,pos):
    try:
        cor1 = VariaveisGlobal[lista[pos][0]]
    except:
        cor1 = lista[pos][0]
    try:
        cor2 = VariaveisGlobal[lista[pos][1]]

    except:
        try:
            cor2 = lista[pos][1]
        except:
            cor2 = cor1

    turtle.color(cor1,cor2)
    return
```

Mesmo após isso ficará com alguns bugs, caso a cor comece com R, B, F ou L,(minúsculo) o programa pode não reconhecer as cores e da erro na hora de inserir a usar a variável;

Mas após trocar esta parte do código, poderá ser feito funções do tipo:

def desenhaColorido(CorI,CorII,a,b,c) c CorI CorII { repeat a { f b r c}};

desenhaColorido(red,yellow, 36, 200, 170);

Lembrando algumas cores podem bugar e não funcionará direito a chamada da função, irá funcionar com os valores anteriores, use C maiúsculo no nome da variável para não confundir com a instrução change_color;

No mais ai está alguns teste, o If em si está mais como um verdadeiro ou falso, o then e o else não foram feitos para receber um assign, então não é possível usar:
if a<b then c=a else c=b; deixando o If uma função meio esquecida na maior parte do programa, porém sendo possível ainda utilizar.