

FootLang

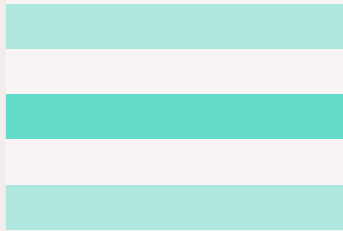
LINGUAGEM DE
PROGRAMAÇÃO PARA
CORREDORES



FootLang

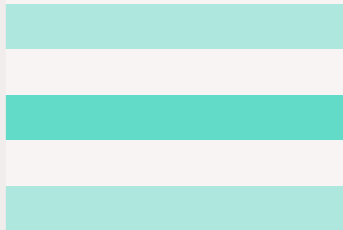
Objetivo

Apresentar FootLang, uma nova linguagem de programação inspirada no dinamismo e na estratégia das corridas



Inspiração

A popularidade crescente das corridas, seja em eventos esportivos ou como metáfora para desempenho e eficiência em várias indústrias.



Motivação

Por que FootLang?

Simplicidade e Velocidade

Desenvolvida para oferecer uma abordagem simplificada à programação, ideal para quem quer simular algumas corridas

Educação e Acessibilidade:

Facilitar o aprendizado de programação com uma sintaxe intuitiva, inspirando-se na clareza e na precisão necessárias nas estratégias de corrida.

CARACTERÍSTICAS

Eficiência em Tempo Real

Otimizada para aplicações que requerem respostas rápidas, como simulações e jogos de corrida.

Sintaxe Intuitiva

Uma sintaxe que reflete a velocidade e a estratégia das corridas, permitindo aos usuários implementar funcionalidades complexas com poucas linhas de código. Inspirado em C e Lua.

Multiplataforma:

Compatível com diversas plataformas, permitindo desenvolvimento flexível em ambientes Windows, Linux e macOS.

EXEMPLO

ENTRADA

```
1 runner1.energy = 100;
2 runner2.energy = 100;
3 runner1.velocity = 5;
4 runner2.velocity = 5;
5
6 volta = 1;
7 tempo = 0;
8
9 repeat (volta < 10) {
10     runner1.energy = runner1.energy - 10;
11     runner2.energy = runner2.energy - 10;
12
13     tempo = tempo + 1;
14
15     winner("volta: ");
16     winner(volta);
17
18     decide (runner1.energy > 50) {
19         runner1.velocity = runner1.velocity + 1;
20     }
21     decide (runner2.energy > 50) {
22         runner2.velocity = runner2.velocity + 2;
23     }
24
25     decide (runner1.energy == 0 || runner2.energy == 0) {
26         winner("Um dos corredores está sem energia!");
27         winner("Corrida interrompida!");
28         volta = 10;
29     }
30
31     volta = volta + 1;
32
33     winner(" ");
34 }
35
36 distancia1 = runner1.velocity * tempo;
37 distancia2 = runner2.velocity * tempo;
38
39 winner("chegada");
40
41 decide (distancia1 > distancia2) {
42     winner("Runner1 é o campeão! Distância: " .. distancia1 .. " metros, Tempo: " .. tempo .. " segundos");
43 } otherwise {
44     winner("Runner2 é o campeão! Distância: " .. distancia2 .. " metros, Tempo: " .. tempo .. " segundos");
45 }
```

ENTRADA

```
1 runner1.energy = 100;
2 runner2.energy = 100;
3 runner1.velocity = 5;
4 runner2.velocity = 5;
5
6 volta = 1;
7 tempo = 0;
8
9 repeat (volta < 10) {
10     runner1.energy = runner1.energy - 10;
11     runner2.energy = runner2.energy - 10;
12
13     tempo = tempo + 1;
14
15     winner("volta: ");
16     winner(volta);
17
18     decide (runner1.energy > 50) {
19         runner1.velocity = runner1.velocity + 1;
20     }
21     decide (runner2.energy > 50) {
22         runner2.velocity = runner2.velocity + 2;
23     }
24
25     decide (runner1.energy == 0 || runner2.energy == 0) {
26         winner("Um dos corredores está sem energia!");
27         winner("Corrida interrompida!");
28         volta = 10;
29     }
30
31     volta = volta + 1;
32
33     winner(" ");
34 }
35
36 distancia1 = runner1.velocity * tempo;
37 distancia2 = runner2.velocity * tempo;
38
39 winner("chegada");
40
41 decide (distancia1 > distancia2) {
42     winner("Runner1 é o campeão! Distância: " .. distancia1 .. " metros, Tempo: " .. tempo .. " segundos");
43 } otherwise {
44     winner("Runner2 é o campeão! Distância: " .. distancia2 .. " metros, Tempo: " .. tempo .. " segundos");
45 }
```

SAÍDA

```
trinta31@LAPTOP-30:~/logcomp/footlang$ python3 compilador.py example.txt
volta:
1

volta:
2

volta:
3

volta:
4

volta:
5

volta:
6

volta:
7

volta:
8

volta:
9

chegada
Runner2 é o campeão! Distância: 117 metros, Tempo: 9 segundos
```

CURIOSIDADES

Estrutura de Controle

A FootLang possui uma estrutura de controle específica que inclui declarações de repetição (**repeat**), decisão (**decide**), e uma estrutura de decisão alternativa (**otherwise**). Isso permite uma programação estruturada e controlada de fluxos lógicos dentro do programa, facilitando a implementação de lógicas complexas.

Identificadores Predefinidos

A linguagem utiliza identificadores predefinidos com atributos específicos, como `velocity` e `energy`. Esses atributos são utilizados diretamente no código, por exemplo, `runner1.energy` e `runner2.velocity`. Isso adiciona uma camada de semântica à linguagem, tornando-a adequada para simulações ou jogos que envolvem entidades com propriedades específicas.

Declaração de Vencedor

Um elemento único da FootLang é a declaração de vencedor (**winner**), que parece ser utilizada para exibir mensagens ou resultados durante a execução do programa. Por exemplo, a função **winner** é usada para anunciar o vencedor de uma corrida e para exibir mensagens durante a execução das voltas da corrida, conforme o código de exemplo fornecido.

OBRIGADO

FELIPE TRINTIM