

TennisScript - Uma Linguagem Inspirada no Tênis

Gabriel de Araújo Alves



Apresentação

AGENDA



- 01 - Motivação
- 02 - Características
- 03 - Curiosidades
- 04 - Exemplos

01



MOTIVAÇÃO

MOTIVAÇÃO

1. Paixão pelo Tênis e Programação:

- A junção da paixão pelo esporte Tennis aliada a uma descrição divertida da linguagem foram da grande valia para a criação

2. Sintaxe Única:

- Não há uma diferença significativa de comandos mas as diferenças de sintaxe tornam a linguagem única e também divertida



02

CARACTERÍSTICAS

01 - CARACTERÍSTICAS

- Linguagem compartilhada com manutenção de alguns tokens go ou python e modificação principalmente das estruturas de condição e Loops.
- Declaração de variáveis : var -> ball
- Estruturas condicionais: if -> serve e else ->fault
- Loops: for -> set
- Tipagem: string-> Match e int -> Point

01-CARACTERÍSTICAS

- Outras Equivalências:
 - Println -> UmpireCall
 - Scanln -> UmpireAsk

03

CURIOSIDADES



03 - CURIOSIDADES

LINGUAGEM LIGADA A
ESPORTES

TOKENS COMO MATCH E
POINT, DENTRE OUTROS,
SÃO BEM INCOMUNS



Ser uma linguagem ligada aos esportes já é algo um pouco incomum dentro do universo da programação, de maneira que, um int virar Point e um string virar Match não se vê em todo lugar. Logo, a construção propôs tornar algumas estruturas da programação em Go e Python mais inclusivas aos amantes do esporte .



04

EXEMPLOS



04 - EXEMPLOS

Declaração de
Variaveis(var_dec.tsc

)

ball x Point = 3

ball y Point = 7

ball z Point

z = x + y

UmpireCall(z)

#OUTPUT: 10

String(str.tsc)

ball str1 Match

ball str2 Match

str1 ="Hello "

str2 = "World"

UmpireCall(str1,str2)

#OUTPUT: Hello Word



04 - EXEMPLOS

Mix de estruturas(mix.tsc)

```
ball num Point  
    num = 10  
serve num > 5 {  
    num = num * 2  
} fault {  
    num = num + 3  
}
```

```
set i = 0; i < num; i=i+1 {  
    UmpireCall(i)  
}
```

#OUTPUT: 0 até 19 indo de 1 em 1



04 - EXEMPLOS

Loops (loop.tsc)

```
ball sum Point=0  
set i = 1; i < 6 ; i = i + 1 {  
    sum=sum+i  
}  
UmpireCall(sum)
```

#OUTPUT: 10

Condicionais(cond.tsc)

```
ball a Point  
a = 5  
  
serve a > 3 {  
    a = 2  
} fault {  
    a = 7  
}
```

UmpireCall(a)

#OUTPUT: 2

OBRIGADO