

PotterScript

a linguagem que qualquer
bruxo que domina a arte das trevas
PRECISA SABER



Descrição

Quando escolhemos desenvolver o projeto de criação do PotterScript, nossa intenção foi maior que construir uma linguagem apenas. Quisemos construir um pouco da magia que nos encanta muito no mundo de Harry Potter, diretamente na lógica do código. Acreditamos, que, quando direcionamos nossos esforços para algo que gostamos, o processo de aprendizado se torna mais divertido e mais leve para nós. Escolher feitiços, criaturas, locais icônicos, etc. como palavras-chave e conceitos não foi apenas uma questão de criatividade. Mas, uma forma de trazer algo íntimo e pessoal para nós.

Nosso desejo era transformar o desafio de aprender os fundamentos de “compiladores” em uma experiência mais leve, quase como se estivéssemos folheando um dos documentos do bruxo que o nome não deve ser dito. Se o processo de aprendizado ficou um pouco mais divertido e didático, chegamos ao nosso objetivo.

Nessa jornada de criação, não estivémos sozinhos, gostaríamos de expressar nossa sincera e carinhosa gratidão ao professor Charles (ou Charlão, ou Mago Charles), que foi como um “Patronum” para nós, não só durante a criação do projeto, mas durante o semestre inteiro. Seus conselhos foram basicamente mapas do Maroto e a paciência e didática, a mais poderosa das “Poção Felix Felicis”, que nos deu confiança e vontade de aprender mais.

Esse projeto, se trata de um reflexo de nosso aprendizado, e um tributo à sua dedicação para nos ensinar, não apenas “como”, mas principalmente o “porquê” por trás de tudo.

Com muito carinho
Os criadores do PotterScript.

Índice

Essências Primordiais	Pag 2
Essenciais Primordiais(II)	Pag 3
Encantamento de Decisão	Pag 4
Encantamentos de Laço	Pag 5
Criação de feitiços	Pag 6
Comunicação Arcana	Pag 7
Magia dos Operadores	Pag 8
Operadores de Comparação	Pag 9
Operadores Lógicos	Pag 10

Essências Primordiais (I)

Todo feitiço, por mais simples ou complexo, manipula alguma forma de energia ou substância mágica.

No PotterScript, chamamos essas substâncias fundamentais de Essências ou como os trouxas diriam, Tipos de Dados.

Compreender essas essências é o primeiro passo para se tornar um conjurador habilidoso, pois cada uma possui propriedades e usos únicos.

Essência

int

Equivalência
Trouxa

íntero

Declaração

```
magic  
int idadedodragao = 732;  
endmagic
```

Atenção, Conjurador: Tentar armazenar um líquido precioso (um número com casas decimais) em um frasco destinado apenas a cristais ínteros (int) pode resultar na perda da parte líquida (truncamento)!

dec

float, double

```
magic  
dec TemperaturaDocao = 97.5;  
endmagic
```

- Ideal para armazenar cálculos que exigem precisão
- Utilize (.) para separar as casas, à moda trouxa.

boolean

boolean

```
magic  
boolean ChapeuDecidiu = true;  
endmagic
```

- Só pode conter True ou False
- Indispensável para feitiços como incêndio e crucio

Essências Primordiais (II)

Essencia

Equivalência
Troupa

Declaração

str

string

magic

str MeuPatrono = "Cervo";
endmagic

- O texto deve estar envolto em aspas duplas
- Você pode concatenar esse formato com um +

null

None, null

magic

str MapaDoMaroto = NULL;
endmagic

Tentar usar uma variável que contém NULL em certas operações mágicas (como cálculos aritméticos) pode resultar em um Erro de Vazio Inesperado (NullPointerException em dialeto trouxa). Tenha cautela!

Encantamentos de Decisão

Um verdadeiro bruxo sabe que nem toda magia deve ser desencadeada indiscriminadamente.

Muitas vezes, é preciso avaliar as circunstâncias, ponderar as condições e apenas então decidir qual encantamento é o mais apropriado. No PotterScript, os Encantamentos de Decisão permitem que seus programas façam exatamente isso, guiando o fluxo da magia com sabedoria e precisão.

Feitiço

Equivalência
Trouxa

Declaração

incendio,
deflexio &
protego

if,
else if &
else

```
magic
int nota = 75

incendio (nota >= 90) {
    revelio("Conceito A")
}

deflexio (nota >= 70) {
    revelio("Conceito B")
}

protego {
    revelio("Conceito C")
}
endmagic
```

- O bloco deflexio é opcional, e podemos encadear para fazer várias condições.
- O bloco protego é opcional e serve como "else", ou seja, executa se nenhuma condição anterior for verdadeira.

Encantamentos de Laço

Um verdadeiro bruxo sabe que nem toda magia deve ser desencadeada indiscriminadamente.

Muitas vezes, é preciso avaliar as circunstâncias, ponderar as condições e apenas então decidir qual encantamento é o mais apropriado. No PotterScript, os Encantamentos de Decisão permitem que seus programas façam exatamente isso,

guiando o fluxo da magia com sabedoria e precisão.

feitiço

accio

crucio

Equivalência
Trouxa

for

while

Declaração

```
magic
accio (int i = 0; i < 5; i = i + 1) {
    revelio("Contador: " + i)
}
endmagic
```

```
magic
int energia = 10
```

```
crucio (energia > 0) {
    revelio("Energia: " + energia)
    energia = energia - 1
}
endmagic
```

- Use accio quando o número de repetições é conhecido ou facilmente determinado por um contador progressivo.
- Use crucio quando a repetição depende de uma condição que pode mudar de forma menos previsível durante a execução do ciclo.

Criação de feitiços

Todo grande bruxo, de Merlin a Dumbledore, não apenas utilizava feitiços existentes, mas também dedicava tempo a criar os seus próprios encantamentos, personalizados para necessidades específicas. No PotterScript, você também possui este poder através da criação de funções, ou como preferimos chamá-las, seus próprios spells (feitiços).

Feitiço spell

Declaração

```
magic
spell int dobrarNumeroMagico (int numero) {
    int resultado;
    resultado = numero * 2;
    finite resultado; // Retorna o número dobrado
}
endmagic
```

- Utilize essa função para criar poderosos feitiços, como quiser.
- Poderosos feitiços da artes das trevas virão em breve.

Comunicação Arcana

Para que seus feitiços interajam com o mundo (ou com o conjurador), PotterScript oferece encantamentos para exibir informações e para receber dados.

feitiço

Equivalência
Trouxa

revelio

print

Sintaxe Arcana

magic

```
str nomeFeiticeiro = "Merlin";
int idade = 1500;
revelio("Saudações de " + nomeFeiticeiro + "!");
revelio("Idade estimada: " + idade + " anos.");
endmagic
```

- revelio: Palavra de poder.

- (<expressão>): O que será exibido (texto entre aspas, variáveis, números, etc.).

- :: finaliza o encantamento.

legilimens input, read

Sintaxe Arcana

magic

```
str nomeConjurador;
revelio("Ó nobre conjurador, qual o seu nome?");
legilimens(nomeConjurador);
revelio("Bem-vindo à Escola de Magia PotterScript, " + nomeConjurador + "!");
endmagic
```

- Nota Arcana: O legilimens geralmente captura a entrada como uma essência str.

- Se precisar de um número, você terá que aprender feitiços de conversão (se PotterScript os tiver no futuro!).

Magia dos Operadores

Operadores são símbolos especiais que realizam operações sobre um ou mais valores (operandos), como calcular, comparar ou combinar essências.

Símbolo	Feitiço	Exemplo
+	Adição	int soma = 5 + 3;
-	Subtração	int dif = 5 - 3;
*	Multiplicação	int prod = 4 * 6;
/	Divisão	dec quoc = 10 / 4;
%	Módulo	int resto = 10 % 3;

Operadores de Comparação

Usados para comparar dois valores, resultando em true ou false.
Essenciais para incendio, crucio, etc.

Símbolo	Feitiço	Exemplo
$=$	Igual a	nível = 9000
$!=$	Diferente de	nome != "Voldemort"
$>$	Maior que	idade > 17
\geq	Maior/igual	pontos >= 100
$<$	Menor	tempratura < 100
\leq	Menor/Igual	tempratura <= 100

Operadores Lógicos

Usados para combinar ou inverter expressões booleanas.

Palavra	Feitiço	Exemplo
and	€ lógico	tem varinha and sabe feitiço
or	OD lógico	eh bruxo or trouxa
not	NÃO Lógico	not portaTrancada