



Haskell

Cristian Carvalho dos Reis

História

- Ano de criação: 1987
- Utiliza paradigma de linguagem funcional
- Origem do nome: homenagem ao lógico Haskell Curry.
- Inspiraram sua criação: Miranda e ML
- Criada na conferência de linguagens funcionais de 87

Características

- Fortemente tipada
- Avaliação preguiçosa
- Funções puras
- Não existência de variáveis
- Linguagem funcional
- Linguagem altamente recursiva
- list comprehension

Read/Write

- EXEMPLO EM HASKELL
`map (\x->2*x) [x|x<-[5,7,3]]`

- Exemplo em C
`#include<stdio.h>`

```
int main(void)
{
    int notas[3] = {5, 7, 3};

    printf("%d\n", notas[0]*2);
    printf("%d\n", notas[1]*2);
    printf("%d\n", notas[2]*2);
    getch();
    return 0;
}
```



Exemplos

- ver pasta presente no github.

Expressividade

- Em Haskell podemos utilizar funções Map, Filter e Reduce com facilidade. Em linguagem C, esses tipos de funções não existem (a não ser que se utilize tabela Hash ou alguma biblioteca específica). Um outro exemplo é a utilização de listas infinitas, que em haskell existe uma solução elegante para isso, em C não. Logo podemos dizer que **para esses tipos de situações** Haskell é mais expressivo que C.

Exemplo: O crivo de Eratóstenes

Neste exemplo em haskell é utilizado o conceito de lista infinita citado no slide anterior, em C por exemplo não existe solução elegante para lista infinitas

```
primos :: [Integer]
```

```
primos = crivo [2..]
```

```
crivo :: [Integer] -> [Integer]
```

```
crivo (p:xs) = p : crivo [x | x<-xs, x`mod`p/=0]
```

```
> take 10 primos
```

```
>[2,3,5,7,11,13,17,19,23,29]
```

Aplicações em Haskell

- Darcs
- Linspire
- Xmonad
- Bluespec SystemVerilog

Site oficial

- <https://www.haskell.org/>