

Haskell:

Sintaxe (Let)

Profa. Dra. Gina M. B. Oliveira

Expressões **let**

Uma **expressão let** é formada por uma lista de declarações e por uma expressão:

let *declarações* **in** *expressão*

O escopo das declarações é a expressão e o lado direito das declarações. Portanto os nomes introduzidos nas declarações só podem ser usados nas próprias declarações e na expressão.

O tipo da expressão **let** é o tipo da expressão que aparece no seu corpo.

O valor da expressão **let** é o valor da expressão que aparece no seu corpo, calculado em um contexto que inclui as variáveis introduzidas nas declarações.

A expressão **let** se estende à direita tanto quanto possível.

Expressoes **let** (cont.)

Exemplos:

```
Prelude> let a = 5 in (a-1)*(a+1)  
24
```

Expressoes **let** (cont.)

Exemplos:

```
Prelude> let {y=1+2;z=4+6} in y+z  
13
```

Expressoes **let** (cont.)

Exemplos:

```
Prelude> let r=3;s=6 in r^2+s^2  
45
```

Expressoes **let** (cont.)

Exemplos:

```
Prelude> let {a=1;b=(-5);c=6;r=sqrt(b^2-4*a*c);k=a/2} in ((-b+r)/k,(-b-r)/k)
(12.0,8.0)
```

Expressoes **let** (cont.)

Exemplos:

```
*Main> 4*let x = 5-2 in x*x  
36
```

Expressões **let** (cont.)

```
(let x = 5-2 ln x*x)^2  
~ 81
```


Expressoes **let** (cont.)

```
[ let square x ■ x*x in (square 5, square 3, square 2) ]  
~ [(25,9,4)]
```

Expressoes **let** (cont.)

```
( let a = 100; b = 200; c = 300 in a*b*c  
  , let foo = "Hey"; bar = "there!" in foo++bar  
  )  
~ (6000000, "Hey there!")
```

Expressões **let** (cont.)

```
let (a,b,c) = (1,2,3) in a+b+c  
~ 6
```

Expressões **let** (cont.)

```
let a = 1 in let b = 2 in a+b  
~ 3
```

Expressoes **let** (cont.)

```
let x = 7 in (let x = "foo" in x, x)  
~ ("foo", 7)
```

Expressoes **let** (cont.)

```
let  c = 10  
      (a,b) = (3*c, f2) f x = x+7*c  
in f a + f b  
~ 242
```

Expressões **let** (cont.)

Também é possível utilizar na definição de funções (arquivo .hs):

```
f a b = let {y=a*b;g x = (x+y)/y} in g (2*a) +g (3/b)
```

```
*Main> f 3 4  
2.5625
```

Ou ainda é possível aproveitar a “regra de layout” nas declarações do let:

```
f a b = let  
    y=a*b  
    g x = (x+y)/y  
in  
    g (2*a) +g (3/b)
```

```
*Main> f 3 4  
2.5625
```

Expressões **let** (cont.)

-- Cálculo da área da superfície de um cilindro

```
areaSuperficie r h = let  
    areaLado = 2* pi* r * h  
    areaBase = pi*r*r  
in  
    areaLado + 2*areaBase
```

```
*Main> areaSuperficie 1 2  
18.84955592153876
```


Diferenças entre **let** e **where**

Com **where** as definições são colocadas no final, e com **let** elas são colocadas no início.

let é uma expressão e pode ser usada em qualquer lugar onde se espera uma expressão.

Já **where não é uma expressão**, podendo ser usada apenas para fazer definições locais em uma definição de função.