

# interpretador Hall

## Fazendo uma tabela de senos

O interpretador Hall disponibiliza algumas funções trigonométricas, a saber:

- **Pi**
- **Seno**
- **Cosseno**
- **Tangente**
- **ArcoSeno**
- **ArcoCosseno**
- **ArcoTangente**
- **GrausParaRadianos**
- **GrausParaGrados**
- **RadianosParaGraus**
- **RadianosParaGrados**
- **GradosParaGraus**
- **GradosParaRadianos**

Vamos exemplificar abaixo a criação de uma tabela de senos. Iremos fazer a tabela considerando o arco expresso em graus e também em radianos. Veja o código:

```
Tabela de senos (graus)  
algoritmo()  
{  
    real x,fx;  
  
    para (x:=0 ate 360 incr 15)  
    {  
        fx := seno(x,"g");  
        escreva("x = ",x," -> fx = ",fx);  
    }  
}
```

Analisando o código acima...

Inicialmente são declaradas duas variáveis de nomes **x** e **fx**, ambas do tipo **real**. A variável **x** representará o valor do ângulo (ou do arco) e a variável **fx** representará o valor da *função-de-x*, nesse caso, receberá o valor do **seno** de **x**.

Em nosso exemplo, estamos considerando a medida do arco em **graus** e fazemos o ângulo variar de 0 a 360 com incrementos de 15. A variável que

representará o ângulo é a variável **x** e é ela que controla o laço para.

Dentro do laço é calculado o valor da função seno com a seguinte chamada:

```
fx := seno(x,"g");
```

Essa chamada é interpretada da seguinte maneira: A função **Seno** irá retornar o valor do *seno* do ângulo **x** e esse ângulo está expresso em graus. A letra **g** entre as aspas ("g") ou também entre apóstrofes ('g') é o parâmetro que especifica a unidade de medida do ângulo **x**. Há duas possibilidades possíveis no interpretador: **g** para graus e **r** para radianos.

Após a chamada, os valores são escritos na tela. Veja a tela de execução abaixo:

## Tabela de senos (graus)

C:\ Prompt de comando - hall



interpretador de algoritmos v-1.0

# informe o nome do arquivo: tabsenos

```
x = 0.000000 -> fx = 0.000000
x = 15.000000 -> fx = 0.258819
x = 30.000000 -> fx = 0.500000
x = 45.000000 -> fx = 0.707107
x = 60.000000 -> fx = 0.866025
x = 75.000000 -> fx = 0.965926
x = 90.000000 -> fx = 1.000000
x = 105.000000 -> fx = 0.965926
x = 120.000000 -> fx = 0.866025
x = 135.000000 -> fx = 0.707107
x = 150.000000 -> fx = 0.500000
x = 165.000000 -> fx = 0.258819
x = 180.000000 -> fx = 0.000000
x = 195.000000 -> fx = -2.588190E-01
x = 210.000000 -> fx = -5.000000E-01
x = 225.000000 -> fx = -7.071068E-01
x = 240.000000 -> fx = -8.660254E-01
x = 255.000000 -> fx = -9.659258E-01
x = 270.000000 -> fx = -1.000000E+00
x = 285.000000 -> fx = -9.659258E-01
x = 300.000000 -> fx = -8.660254E-01
x = 315.000000 -> fx = -7.071068E-01
x = 330.000000 -> fx = -5.000000E-01
x = 345.000000 -> fx = -2.588190E-01
x = 360.000000 -> fx = -0.000000
```

O mesmo programa para o ângulo expresso em radianos está disponível abaixo, acompanhe...

```
Tabela de senos (radianos)
algoritmo()
{
    real x,fx;
    real d;

    d := 15.0*PI()/180.0;

    para (x:=0 ate 2*pi() incr d)
    {
        fx := seno(x,"r");
        escreva("x = ",x," -> fx = ",fx);
    }
}
```

Analisando...

No exemplo acima, a tabela dos senos é construída mas, agora o ângulo é expresso em **radianos**. Veja a chamada: **fx := seno(x,"r");** onde o parâmetro **"r"** é o que especifica que o ângulo **x** está sendo passado em radianos.

O exemplo é exatamente o mesmo acima, o ângulo começa com 0 (zero) e percorre toda a circunferência ( $360=2\pi$ ) com incrementos de 15 em 15. O incremento foi calculado na variável **d** e foi colocado fora do laço para evitar que o valor fosse re-calculado repetidamente em toda iteração do laço. Uma vez que o valor da variável **d** é constante não faz sentido recalculá-lo toda vez. Esse procedimento aumenta a velocidade de execução do algoritmo.

O resultado da execução acima pode ser visto abaixo...

## Tabela de senos (radianos)

C:\ Prompt de comando - hall



interpretador de algoritmos v-1.0

# informe o nome do arquivo: tabsenos2

```
x = 0.000000 -> fx = 0.000000
x = 0.261799 -> fx = 0.258819
x = 0.523599 -> fx = 0.500000
x = 0.785398 -> fx = 0.707107
x = 1.047198 -> fx = 0.866025
x = 1.308997 -> fx = 0.965926
x = 1.570796 -> fx = 1.000000
x = 1.832596 -> fx = 0.965926
x = 2.094395 -> fx = 0.866025
x = 2.356194 -> fx = 0.707107
x = 2.617994 -> fx = 0.500000
x = 2.879793 -> fx = 0.258819
x = 3.141593 -> fx = -0.000000
x = 3.403392 -> fx = -2.588190E-01
x = 3.665191 -> fx = -5.000000E-01
x = 3.926991 -> fx = -7.071068E-01
x = 4.188790 -> fx = -8.660254E-01
x = 4.450590 -> fx = -9.659258E-01
x = 4.712389 -> fx = -1.000000E+00
x = 4.974188 -> fx = -9.659258E-01
x = 5.235988 -> fx = -8.660254E-01
x = 5.497787 -> fx = -7.071068E-01
x = 5.759587 -> fx = -5.000000E-01
x = 6.021386 -> fx = -2.588190E-01
x = 6.283185 -> fx = -0.000000
```