

**Grupo de Pesquisa em Álgebras de Clifford e Processamento de Sinais e Imagens**

**Edjair Aguiar Gomes Filho – Representante Discente pelo Centro Acadêmico de Engenharia  
da Computação**

**Universidade Federal do Vale do São Francisco**

**Minicurso de Linguagem Python**

**Aula 2 - Módulos**

**Sumário**

<b>1) MÓDULOS .....</b>	<b>2</b>
<b>2) ACESSANDO ATRIBUTOS DE MÓDULOS.....</b>	<b>2</b>
<b>3) MÓDULOS PADRÕES .....</b>	<b>3</b>
<b>4) OBSERVAÇÕES .....</b>	<b>4</b>
<b>5) EXERCÍCIOS.....</b>	<b>4</b>

## 1) MÓDULOS

Módulos são arquivos de código Python com a extensão `.py`, que podem ser importados por outros módulos.

A modularização de um programa é algo bastante importante por duas razões principais:

- É possível reutilizar o código escrito em outras aplicações;
- Estando o código do programa organizado, temos maior facilidade no controle e no entendimento da aplicação;

Quando importamos um módulo para nosso código, as declarações do arquivo importado (suas variáveis, suas funções e etc.) ficam disponíveis para serem usadas. É mais ou menos como a importação de uma biblioteca na linguagem `.C`, por exemplo.

Nós chamamos um módulo no nosso programa utilizando a linha de comando

```
import nomeDoMódulo
```

A extensão `.py` deve ser omitida.

## 2) CRIANDO E ACESSANDO ATRIBUTOS DE MÓDULOS

Cada módulo tem suas próprias declarações de variáveis, funções, classes... cada uma delas carrega com si seu *nome*. Podemos acessar as variáveis e funções do módulo no nosso programa. Para isso, utilizamos a notação

```
nomeDoMódulo.nomeDaFuncao
```

EXEMPLO: Podemos abrir o editor de texto e escrever uma função simples como na Figura 1. Salvamos o nome do arquivo como `teste.py`.

```
1 def f():
2     print("Alô!");
```

Figura 1 – Código do módulo `teste.py`

ATENÇÃO: Salve o arquivo do módulo em uma pasta fácil de acessar, pois antes de chamar o interpretador *python* precisaremos acessar a pasta pelo prompt de comando. Eu salvei na pasta Documentos. Dessa forma, primeiro acessei a pasta onde salvei o arquivo e só então chamei o interpretador *python*, como visto na Figura 2.

```
C:\Users\USUARIO>cd Documents
C:\Users\USUARIO\Documents>python
```

Figura 2 – Precisamos acessar a pasta onde o arquivo foi salvo antes de chamar o python

Agora que salvamos nosso módulo, vamos chamá-lo para nosso programa, escrevendo `import teste`. Logo em seguida, chamamos nossa função `f` digitando `teste.f()`. O resultado podemos ver na Figura 3.

```
>>> import teste
>>> teste.f()
Alô!
```

Figura 3– Chamamos a função `f()` através do módulo `teste.py`

### 3) MÓDULOS PADRÕES

O Python traz uma biblioteca padrão de módulos, embutidas no interpretador da linguagem. Esses módulos permitem o acesso a algumas operações que não fazem parte do núcleo da linguagem, mas estão no interpretador por eficiência.

Por exemplo: funções matemáticas não estão tradicionalmente incluídas no escopo de funções de entrada e saída da linguagem. Nesse caso, temos duas opções: implementá-las manualmente; ou importar o módulo `math`, que disponibiliza grande parte das funções matemáticas que conhecemos na calculadora científica.

<code>math.exp(x)</code>	$e^x$
<code>math.log10(x)</code>	Logaritmo do parâmetro $x$ na base 10
<code>math.pow(x, y)</code>	$x^y$
<code>math.sqrt(x)</code>	$\sqrt{x}$
<code>math.degrees(x)</code>	Converte $x$ radianos para graus
<code>math.radians(x)</code>	Converte $x$ graus para radianos

Tabela 1 – Algumas funções do módulo `math`

Na Figura 4 temos um exemplo da utilização do módulo `math`, utilizando a função `math.sqrt`.

```
>>> import math
>>> print(math.sqrt(25))
5.0
```

Figura 4 – Cálculo de raiz quadrada com função `math.sqrt`

#### 4) OBSERVAÇÕES

- O editor de texto usado é o Visual Studio Code. Entretanto é possível utilizar qualquer outro editor de texto, desde que a extensão do arquivo seja salva como .py. Outros editores de texto que podem ser usados: WordPad, Notepad++.
- A importação de módulos é custosa computacionalmente, portanto o Python só a realiza uma vez. Caso editemos o módulo durante a execução do nosso código e precisemos que a edição seja utilizada, podemos forçar que o módulo seja importado novamente usando o comando `reload(nomeDoMódulo)`.

#### 5) EXERCÍCIOS

1. A Tabela 2 mostra todas as funções trigonométricas que são disponibilizadas pelo módulo padrão `math`. Importe esse módulo e teste essas funções a seu agrado. **Dica:** Essas funções utilizam valores em *radianos*. Você pode brincar com a conversão de valores entre radianos e graus utilizando as funções citadas na Tabela 1!

<code>math.sin(x)</code>	Seno de x, em radianos
<code>math.cos(x)</code>	Cosseno de x, em radianos
<code>math.tan(x)</code>	Tangente de x, em radianos
<code>math.asin(x)</code>	Arco seno de x, em radianos
<code>math.acos(x)</code>	Arco cosseno de x, em radianos
<code>math.atan(x)</code>	Arco tangente de x, em radianos
<code>math.dist([Px, Py], [Qx, Qy])</code>	Distância euclidiana entre dois pontos p e q. Px, Py e Qx, Qy são as coordenadas x e y de ambos os pontos, respectivamente.

Tabela 2 – Funções trigonométricas do módulo `math`

2. A função abaixo nos retorna a soma da série de Fibonacci que mais se aproxima no valor  $n$  que passamos como parâmetro.

```
def fib(n):  
    a, b = 0, 1  
    while a < n:  
        print(a, end=' ')  
        a, b = b, a+b  
    print()
```

Por exemplo, se chamarmos a função com o valor 1000, os números retornados serão 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987. Esse exemplo pode ser visto na Figura 5 abaixo.

```
>>> import fibo  
>>> fibo.fib(1000)  
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Figura 5 – Retorno da função fib para  $n=1000$

Copie o código da função fib e salve num arquivo .py chamado *fibo*. Em seguida importe o arquivo fibo.py como módulo (lembrando que a importação de módulo omite a extensão .py) e teste a função *fib* para os valores de  $n$  que você achar melhor! O intuito do exercício é mostrar como a importação de módulos facilita a definição de funções e melhor distribuição de um programa em pequenos arquivos.