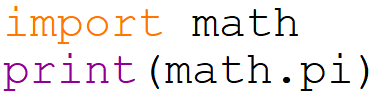
1. Uma biblioteca é o conjunto de módulos, sendo eles um arquivo *.py* com instruções em Python. As bibliotecas adicionam funções novas que não estão presentes no Python original, assim otimizando e possibilitando variadas aplicações.
2. Para fazer o *import* de uma biblioteca, logo no início do programa adicione o termo *import* + *o nome da biblioteca* que deseja adicionar. Como no exemplo a seguir:



Neste exemplo adicionamos a biblioteca *math*, com a função *print* imprimimos o valor de pi.

1. Primeiramente para entender sobre bibliotecas para implementação de interfaces gráficas em Python precisamos falar sobre o que são interfaces gráficas. Conhecida mais como GUI (do inglês *Graphical User Interface*) é um tipo de interface do usuário que permite a interação dele com os dispositivos digitais por meio de elementos gráficos, como por exemplo ícones (como o da câmera do seu celular). Assim o que aparece são esses elementos e não a de linha de comando.

Portanto, para os programadores o GUI é muito importante para facilitar o manuseio para quem está usando. Para o desenvolvimento na linguagem existem algumas bibliotecas para este desenvolvimento, como:

* *WxWidgets;*
* *Tkinter;*
* *Kivy;*
* *PyGTK;*
* *PySide;*
* *QT.*

1. No Python é possível usar o recurso "*def*" para definir uma função, que pode ser chamada dentro do código pelo nome escolhido. Assim que uma função é chamada, tudo que está dentro dela vai ser executado novamente. Por exemplo:



Nesse caso, quando chamamos a função, "*função\_print*", o programa ira imprimir "Olá Mundo", pois é isso que definimos dentro da função.

1. Como citamos antes a biblioteca *Tkinter* é usada no Python para o desenvolvimento das interfaces gráficas. É uma biblioteca com fácil uso e seus recursos, além disso ela é nativo da linguagem Python, então só precisamos importa-la, da seguinte forma:



1. Anexado arquivo exemplo comentado como Exemplo1\_COMENTADO.py
2. *Turtle* é uma biblioteca que nos permite fazer desenhos na tela usando comandos no código. Tem origem como extensão da linguagem “*Logo”*, a qual tem como parecer um robô que o usuário pode ter o controle por meio de comandos de movimentação.

Alguns recursos básicos da biblioteca são (o termo *n* representa o valor inserido pelo usuário):

* Métodos para movimentação e rotação
* *turtle.forward(n)*, tem como finalidade fazer a caneta se movimentar para frente tantos (n) pixels.
* *turtle.backward(n),* tem como finalidade fazer a caneta se movimentar para trás tantos (n) pixels.
* *turtle.left(n)*, tem como finalidade de girar a caneta para esquerda para alterar a posição em graus (n).
* *turtle.right(n)*, tem como finalidade de girar a caneta para direita para alterar a posição em graus (n).
* Método para a criação de um círculo
* *turtle.circle(n)*, tem como finalidade desenhar um círculo em concordância com o raio (n) inserido pelo usuário.
* Método adicionais
* *turtle.shape(n)*, tem como finalidade alterar o formato da caneta. Podendo ser: “*arrow(flecha)*”, “*blank (invisível)*”, “*circle (círculo)*” ,“*classic (clássica)*” ,“*square (quadrado)*” ,“*triangle (triângulo)*” e “*turtle (tartaruga)*”.
* *turtle.speed(n)*, tem como finalidade alterar a velocidade em que a caneta vai percorrer. Tem como opções “*fastest*”, “*fast*”, “*normal*”, “*slow*”, “*slowest*”, ou pode ser preenchida por números, quanto maior mais rápido.
* *turtle.color(n)*, tem como finalidade alterar a cor que o desenho será feito (o preenchimento), há uma tabela com o nome das cores que podem ser usadas.