MODALIDADE:	Aprendizagem +	Escolha um item.	
CURSO:	Técnico/a Especialista em Aplicações Informáticas de Gestão		
UC:	Desenvolver projeto de tecnologias e aplicações informáticas de gestão	CÓDIGO UC:	00891
FORMADOR/A:	Bruno Silva	DATA:	

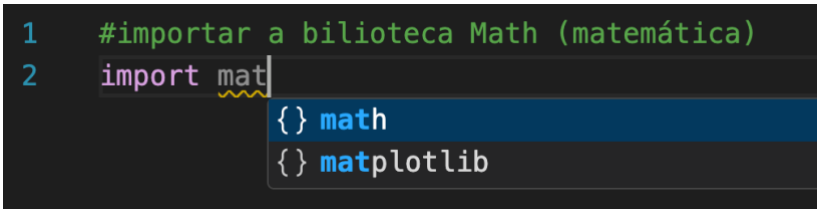
Os módulos em Python são arquivos que contêm funções, classes e variáveis que permitem organizar, reutilizar código. Importados com a palavra-chave **import**, podemos utilizar módulos padrão (que já vêm embutidos com o python) ou externos (instalados via gestor de pacotes pip).

Principais Módulos da Biblioteca Padrão (Built-in):

- **os / sys**: Interação com o sistema operativo e interpretador;
- **math**: Funções matemáticas (seno, cosseno, raiz quadrada, potências, etc...);
- **datetime**: Manipulação de datas e horas;
- **random**: Geração de números aleatórios;
- **json / csv**: Leitura e escrita de arquivos de dados;
- **pathlib**: Manipulação de caminhos de arquivos de forma simplificada;
- **entre outros**;

Importação de módulos

No topo do programa, devem invocar o nome do módulo utilizando a palavra reservada **import** e depois o nome do módulo:



Reparem que a biblioteca math **já vem com a linguagem Python**. A medida que escrevem, esta surge na janela de ajuda. **Mas, será preciso chamar o nome do módulo todas as vezes que queira ir buscar uma função especifica?**

Não

Em vez de chamar pelo nome completo, podemos dar um nome alternativo (e mais simplificado). Como tal, **vamos criar um nome mais curto** que vai **representar tudo o nome da biblioteca**, ajudando na hora da invocação.

A frente do nome da biblioteca podem colocar a palavra reservada **as** com o nome alternativo a frente:

import nome_biblioteca_a_usar **as** nome_alternativo

```
#importar a bilioteca numpy
import numpy as np
```

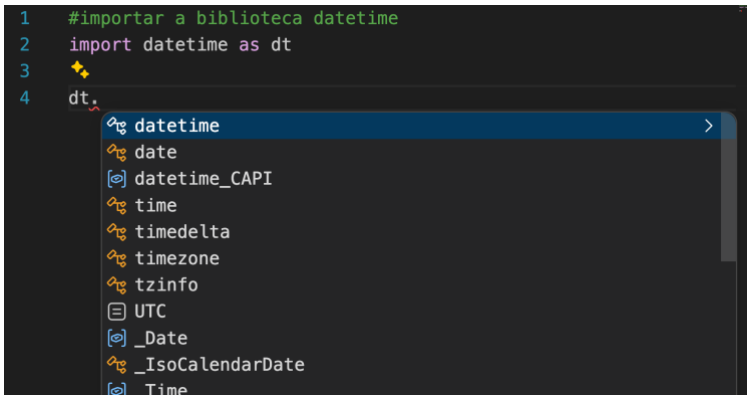
Podemos utilizar esta estratégia para otimizar os nomes das bibliotecas que podem ser muito grandes e facilitar a nossa vida ☺

Importação apenas uma função de um módulo

Os módulos podem ter várias funções para serem utilizadas. Como tal, podemos importar uma função bem especifica. No exemplo mais abaixo, queremos importar a biblioteca da hora/tempo mais conhecida como **datetime** (e com o nome abreviado dt):

```
1   #importar a biblioteca datetime
2   import datetime as dt
```

Mas a biblioteca tem imensas funções embutidas:



Neste caso, queremos apenas invocar a **função para dar a hora e data atual** (que também chama datetime) do sistema operativo. Como tal, atrás do nome import vamos colocar a instrução from e qual a função especifica que queremos dentro do módulo datetime:

```
1   #importar a biblioteca datetime
2   from datetime import datetime as dt
3
4   print(dt.now())
```

Módulo random – Geração de números aleatórios

O Python pode gerar números aleatórios com recurso ao módulo ***random***. Para tal deverá sempre que possivel, importar este módulo na primeira do seu código. Exemplo:

```
1   import random
2
```

Quando geramos um número aleatório com a função random(), este vai gerar um número decimal no intervalo de números decimais 0 e 1. Exemplo:

```
import random
numero = random.random()
print(numero)
```

```
1 0.0650868
```

Eventualmente, se desejar um número aleatório apenas com números inteiros, temos de ir por outra solução.

Geração de números aleatórios (apenas inteiros) – Função randint()

Para gerar números inteiros, vamos invocar a função ***randint()*** do módulo ***random***. Esta função leva ***dois parâmetros***: um valor ***mínimo*** e um valor ***máximo***. Para obter um número aleatório entre 1 e 10, vamos colocar o valor mínimo 1 e o valor máximo 10, respetivamente:

```
1 import random
2 numero = random.randint(1, 10)
3 print(numero)
```

Resultado: 3

O código do Python vai retornar um número inteiro aleatório entre 1 e 10, de cada vez que executar o programa (incluindo 1 e 10).

1. **Construa um programa para simular a geração de 1 número inteiro (entre 1 e 100) e 1 número decimal.**

2. **Construa um programa para simular a geração da face de um dado, considerando que um dado tem 6 faces.**

Módulo Math

Funções matemáticas sem a necessidade da biblioteca Math:

- min() e max()
- abs()
- pow()
- round()

Funções matemáticas com a necessidade da biblioteca Math:

- sqrt()
- Pi
- Entre outras ...
- Mais fórmulas: https://www.w3schools.com/python/module_math.asp ou https://docs.python.org/pt-br/3.10/library/math.html

3. Elabore um programa para pedir um número decimal ao utilizador (com 3 casa decimais), e de seguida, faça as seguintes operações:

- Converta o número para inteiro com a função **int(valor)**;
- Mostre o número apenas com 1 casa decimal com a função **round(parametro_1, paramtero_2)**, do qual recebe dois parâmetros:
  - **1º Parâmetro:** colocar o valor a trabalhar;
  - **2º Parâmetro:** indicar quantas casas decimais pretende preservar;
- Calcule a potência do número com a função **pow(parametro_1, paramtero_2)**, do qual recebe dois parâmetros:
  - **1º Parâmetro:** colocar o valor a trabalhar;
  - **2º Parâmetro:** qual o expoente da potência;
- Calcule a raiz quadrada com a função **math.sqrt(valor)**;
- Multiplique o valor do número com a constante **math.pi**;

- Arredonde o número com a função **round(parametro_1, paramtero_2)**, do qual recebe dois parâmetros:
  - **1º Parâmetro:** colocar o valor a trabalhar;
  - **2º Parâmetro:** indicar quantas casas decimais pretende preservar para depois arredondar;