

Python - Enums

O termo 'enumeração' refere-se ao processo de atribuição de valores constantes fixos a um conjunto de strings, de modo que cada string possa ser identificada pelo valor vinculado a ela. A biblioteca padrão do Python oferece o módulo **enum**. A classe Enum incluída no módulo **enum** é usada como classe pai para definir a enumeração de um conjunto de identificadores - convencionalmente escritos em letras maiúsculas.

Exemplo 1

```
from enum import Enum

class subjects(Enum):
    ENGLISH = 1
    MATHS = 2
    SCIENCE = 3
    SANSKRIT = 4
```

No código acima, "assuntos" é a enumeração. Possui diferentes membros de enumeração, por exemplo, subject.MATHS. Cada membro recebe um valor.

Cada membro é um objeto dos assuntos da classe de enumeração e possui atributos de nome e valor.

```
obj = subjects.MATHS
print (type(obj), obj.value)
```

Isso resulta na seguinte **saída** -

```
<enum 'subjects'> 2
```

Exemplo 2

O valor vinculado ao membro enum nem sempre precisa ser um número inteiro; também pode ser uma string. Veja o exemplo a seguir -

```
from enum import Enum

class subjects(Enum):
    ENGLISH = "E"
    MATHS = "M"
```



```
GEOGRAPHY = "G"  
SANSKRIT = "S"
```

```
obj = subjects.SANSKRIT  
print (type(obj), obj.name, obj.value)
```

Ele produzirá a seguinte **saída** -

```
<enum 'subjects'> SANSKRIT S
```

Exemplo 3

Você pode iterar pelos membros do enum na ordem em que aparecem na definição, com a ajuda de um loop **for** -

```
for sub in subjects:  
    print (sub.name, sub.value)
```

Ele produzirá a seguinte **saída** -

```
ENGLISH E  
MATHS M  
GEOGRAPHY G  
SANSKRIT S
```

O membro enum pode ser acessado com o valor exclusivo atribuído a ele ou por seu atributo name. Portanto, subject("E") e subject["ENGLISH"] retornam subject.ENGLISH membro.

Exemplo 4

Uma classe enum não pode ter o mesmo membro aparecendo duas vezes; no entanto, mais de um membro pode receber o mesmo valor. Para garantir que cada membro tenha um valor exclusivo vinculado a ele, use o decorador @ **unique** .

```
from enum import Enum, unique  
  
@unique  
class subjects(Enum):  
    ENGLISH = 1  
    MATHS = 2  
    GEOGRAPHY = 3  
    SANSKRIT = 2
```



Isso gerará uma exceção da seguinte forma -

```
@unique
```

```
^^^^^
```

```
raise ValueError('duplicate values found in %r: %s' %
```

```
ValueError: duplicate values found in <enum 'subjects'>: SANSKRIT -> MATHS
```

A classe Enum é uma classe que pode ser chamada, portanto você pode usar o seguinte método alternativo para definir enumeração -

```
from enum import Enum
```

```
subjects = Enum("subjects", "ENGLISH MATHS SCIENCE SANSKRIT")
```

O construtor **Enum** usa dois argumentos aqui. O primeiro é o nome da enumeração. O segundo argumento é uma string que consiste em nomes simbólicos de membros de enumeração, separados por um espaço em branco.