Modules in Hindi - Python in Hindi

Python - Modules

Python में कुछ inbuilt Modules होते है जैसे कि , math, random | हर module की अलग-अलग पहचान होती है | math module में mathematics से related कुछ functions होते है जैसे कि, sin(), cos() और sqrt() इत्यादी.

Python में Modules ये एक तरह का program ही होता है | modules में function और classes होते है |

Python में दो प्रकार के Modules होते है |

1. In-built Modules
2. User-Defined Modules

1. In-built Modules

Python में 350+ Modules होते है | ये modules अलग-अलग काम के लिए बने होते है |

Python के सभी in-built modules देखने के लिए ,

```
help("modules")
```

2. User-Defined Modules

Creating Module in Python

निचे एक program दिया गया है | उसके file का नाम 'arithmetic.py' है | जो नाम file का होता है वही नाम 'module' का होता है | 'arithmetic' module में add, sub, mul और div नाम के functions दिए गए है |

Source Code :
File Name : arithmetic.py

```
def add(a, b):
    return a + b

def sub(c, d):
    return c - d

def mul(e, f):
    return e * f

def div(g, h):
    return g / h
```

Importing Module in Python

Python में current program पर module को import करना हो तो 'import' keyword करना पड़ता है | जिस module को import किया जाता है उसके सारे functions और classes; current program पर इस्तेमाल किये जा सकते है |

Syntax for import Statement

```
import module_name1, module_name2, ...,module_nameN
```

Example पर arithmetic ये user-defined module को import किया गया है और उसके सभी functions(add,sub,mul,div) का इस्तेमाल किया गया है |

अगर किसी भी module के function या class को current program पर access करना हो तो module_name के बाद dot(.) operator और उसके बाद function का या class का नाम देना पड़ता है |

Source Code : Output :

```
10
-2
24
0.6666666666666666
```

import_as Statement for Changing Module Name

Syntax :

```
import module_name as alias_name
```

Module के नाम को change करना हो तो import_as statement का इस्तेमाल किया जाता है |

Example पर arithmetic module का नाम change करके myModule रख दिया गया है |

Source Code : Output :

```
10
-2
24
0.6666666666666666
```

from_import Statement

Syntax :

```
from module_name import name1, name2,.., name2
```

अगर current program पर बिना module name से function को या class को access करना हो तो 'from_import' statement का इस्तेमाल किया जाता है |

Example पर arithmetic module के एक ही functions का इस्तेमाल किया गया है |

For Example, Output :

```
10
```

from_import Statement with *(asterisk)

अगर current program पर बिना module के सभी functions या classes को access करना हो तो 'from_import' के साथ *(asterisk) का इस्तेमाल किया जाता है |

Syntax :

```
from module_name import *
```

For Example, Output :

```
10
-2
24
0.6666666666666666
```

Deleting imported Module

Module को current program से remove करना हो तो 'del' operator का इस्तेमाल किया जाता है |

Syntax :

```
del module_name
```

Example पर arithmetic module को remove किया गया है |

Source Code : Output :

```
10
   print(arithmetic.mul(4, 6))
NameError: name 'arithmetic' is not defined
```

dir() Function

dir() function ये module में कौन-कौन से names defined किये गए है वो सभी inbuilt names ascending order में list के रूप में return करता है | Example पर add, div, mul और sub ये functions भी अलग से ascending order में return किये गए है |

Source Code : Output :

```
['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'add', 'div', 'mul', 'sub']
```

Accessing Module's names(attributes)

जैसे module की मदद से function को access किया जाता है वैसे ही modules के names को access किया जाता है |

names को access करने के लिए module_name.names(attributes) इस तरह से access किया जाता है |

For Example, Output :

```
C:\Users\UD\AppData\Local\Programs\Python\Python36-32\arithmetic.py
arithmetic
```

reload() Function

Python में Interpreter द्वारा जब module के functions या class को execute किया जाता है तब उसके value को जबतक session शुरू है तबतक change नहीं किया जा सकता |

जैसे कि,
myModule.py Output In Shell

```
>>> import myModule
>>> myModule.func()
Hello
```

After Changing value of Module's func() function
Python में session के दौरान सिर्फ एक बार ही module को import किया जा सकता है |

Example में func() function की value को change किया गया है लेकिन shell में जो पहला output था वही output display किया गया है |

myModule.py Output In Shell

```
>>> import myModule
>>> myModule.func()
Hello
```

Reloading a Module using reload() function

reload() का इस्तेमाल module को refresh या reload करने के लिए किया जाता है | ये function module के session को restart कर देता है |

Example पर reload() function से 'myModule' इस module को reload किया गया है |

myModule.py Output In Shell

```
>>> import myModule,importlib
>>> importlib.reload(myModule)
<module 'myModule' from 'C:\\Users\\UD\\AppData\\Local\\Programs\\Python\\Python36-
32\\myModule.py'>
>>> myModule.hello()
Hiiii
```