

MODULES

INTRODUCTION: Modules in Python are simply Python files with the .py extension, which implement a set of functions. Modules are imported from other modules using the import command. To import a module, we use the import command.

KEYWORD AND IDENTIFIERS : Keywords are the reserved words in Python. We cannot use a keyword as variable name, function name or any other identifier.

KEYWORD:

They are used to define the syntax and structure of the Python language. In Python, keywords are case sensitive.

In python we have 33 keywords

print : print to console

while: controlling the flow of the program

for: iterate over items of a collection in order that they appear

break: interrupt the (loop) cycle, if needed

continue: used to interrupt the current cycle, without jumping out of the whole cycle. New cycle will begin.

if: used to determine, which statements are going to be executed.

elif: stands for else if. If the first test evaluates to False, then it continues with the next one

else: is optional. The statement after the else keyword is executed, unless the condition is True

is: tests for object identity

not: negates a Boolean value

and: all conditions in a Boolean expression must be met

or: at least one condition must be met.

import: import other modules into a Python script

as: if we want to give a module a different alias

from: for importing a specific variable, class or a function from a module

def: used to create a new user defined function

return: exits the function and returns a value

lambda: creates a new anonymous function

global: access variables defined outside functions

try: specifies exception handlers

except: catches the exception and executes codes

finally: is always executed in the end. Used to clean up resources.

raise: create a user defined exception

del: deletes objects

pass: does nothing

assert: used for debugging purposes

class: used to create new user defined objects

exec: executes Python code dynamically

yield: is used with generators

IDENTIFIERS: A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9). Python does not allow punctuation characters such as @, \$, and % within identifiers.

- **Naming conventions for identifiers:** Class name should start with uppercase and all other identifiers should start with lowercase.
- Starting an identifier with an underscore defines private.
- Starting an identifier with two leading underscores indicates a strongly private identifier.
- If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

IMPORTING A MODULE: We can use python source file as a module name in another module using import statement

Syntax: import module_name

E.g: Lets consider a source file with name hello.py

```
def abc():
    print("hello abc")

def cdf():
    print("hello cdf")
```

Now take another source file with xyz.py

```
import hello #from hello it will import all the modules
from hello import abc # this will import one abc named module.
```

MATH MODULES: Math is the standard module for mathematical functions available in python.

Syntax: import math

E.g: import math
math.sqrt(4) # gives the square root of the value 4

Math module contains following predefined functions

math.ceil: Return the ceiling of x, the smallest integer greater than or equal to x. If x is not a float it will return the integer value

math.copysign: Return a float with the magnitude (absolute value) of x but the sign of y.

math.fabs: returns the absolute value of x

math.factorial(x): Return x factorial. Raises ValueError if x is not integral or is negative.

math.floor: Returns the floor of x, the largest integer less than or equal to x, if x is not a float it will return integer value

math.gcd: Returns the greater common division value of the give numbers

math.isfinite :Returns true if x is infinite and false if x is not infinite .

math.exp: Returns the exponent value

math.log: Returns the log value

math.pow: returns the power value, x raise to the power y.

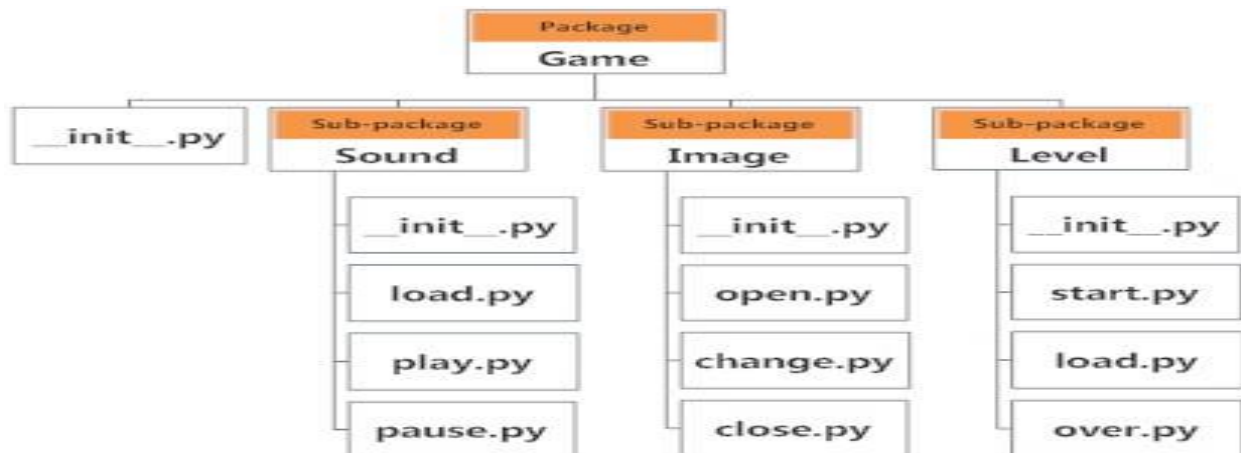
math.sqrt: Returns the square root of the value.

RANDOM MODULE: Random element is used to pick any random number from the given range or from a list.

Syntax: import random
print (random.randint(0,5))#here it picks a random integer number between 0-5

PACKAGES: Packages are namespaces which contain multiple packages and modules themselves. They are simply directories, but with a twist. Each package in Python is a directory which MUST contain a special file called `__init__.py`. This file can be empty, and it indicates that the directory it contains is a Python package, so it can be imported the same way a module can be imported.

Syntax: `import abc` #here abc is an package
`from abc import xyz` # here from abc package we are importing xyz module



We can import modules from packages using the dot (.) operator To import the start module in the example.

```
import Game.Level.start
```

Now if this module contains a function named `difficulty()`.

```
Game.Level.start.select_difficulty(2)
```

We can even call it in another way

```
from Game.Level import start
```

Now we can call function using

```
start.select_difficulty(2)
```

We can create a file named `main.py`, this `main.py` contains the `.py` files with directories and we can execute in `main.py` file