

Video 20 : Static and Modules

In this tutorial I'll cover static methods, static variables, how to make your own modules, import and from.

Static Methods

Static methods allow access without the need to initialize a class. They should be used as utility methods, or when a method is needed, but it doesn't make sense for the real world object to be able to perform a task.

Here I'll create a Sum object that is only used to provide a utility get_sum() function.

CODE

```
class Sum:

    # You use the static method decorator to define that a
    # method is static
    @staticmethod
    def get_sum(*args):

        sum_1 = 0

        for i in args:
            sum_1 += i

        return sum_1

def main():

    # Call a static method by proceeding it with its class
    # name
    print("Sum :", Sum.get_sum(1,2,3,4,5))

main()
```

Static Variables

Fields declared in a class, but outside of any method are static variables. Their value is shared by every object of that class.

CODE

```
class Dog:

    # This is a static variable
    num_of_dogs = 0

    def __init__(self, name="Unknown"):
        self.name = name
```

```

        # You reference the static variable by proceeding
        # it with the class name
        Dog.num_of_dogs += 1

    @staticmethod
    def get_num_of_dogs():
        print("There are currently {} dogs".format(Dog.num_of_dogs))

def main():

    spot = Dog("Spot")

    doug = Dog("Doug")

    spot.get_num_of_dogs()

main()

```

Modules

Your Python programs will contain a main program that includes your main function. Then you will create many modules in separate files. Modules also end with .py just like any other Python file.

CODE

```

# ----- sum.py -----
def get_sum(*args):
    sum_1 = 0

    for i in args:
        sum_1 += i

    return sum_1

# ----- End of sum.py -----

# You can import by listing the file name minus the py
import sum

# Get access to functions by proceeding with the file
# name and then the function you want
print("Sum :", sum.get_sum(1,2,3,4,5))

```

FROM

You can use from to copy specific functions from a module. You can use from sum import * to import all functions. You can also import multiple functions by listing them after import separated by commas.

CODE

```
from sum import get_sum
```

```
# You don't have to reference the module name now  
print("Sum :", get_sum(1,2,3,4,5))
```