Topic 66 - Modules

**What**

* A **module** is a separate Python file that contains functions, classes, and other reusable code. By storing code in modules, you can organize your programs better and make them easier to maintain.

**Why**

* **Code Reusability**: Modules allow you to write a function once and use it across multiple programs.
* **Simplicity**: They help to keep the main program short and easier to read.
* **External Libraries**: You can use code written by others by importing their modules.

**How**

1. **Creating a Module**  
   You can create a module by saving Python code in a .py file. For example, a file called calculations.py:

python

Copy code

# calculations.py

def calc\_tax(sales\_total, tax\_rate):

tax = sales\_total \* tax\_rate

return tax

1. **Using a Module in Your Main Program**  
   To use functions or classes from a module, you import it in your main program:

python

Copy code

import calculations

1. **Calling Functions from a Module**  
   After importing the module, you call functions from it using the module name followed by a dot (.):

python

Copy code

tax\_for\_this\_order = calculations.calc\_tax(sales\_total=101.37, tax\_rate=0.05)

* + Here, calculations.calc\_tax is the function from the calculations.py module.

1. **Advantages**
   * You can use the same module in different programs, ensuring that you don't have to duplicate your code.
   * Modules keep your main program concise and organized, making it easier to read and maintain.

**Things to Remember**

* The **module filename** must have a .py extension (e.g., calculations.py).
* When importing, omit the .py extension. You just use the module's name, e.g., import calculations.
* Modules help avoid repetition and provide the benefit of using well-tested, reusable code across projects.