1. Defining a Function: When you create a function in Python, you're basically making a mini-program that you can use later. You start with `def`, then give your function a name, and tell it what to do inside.

2. Reasons for Using Functions: Functions help you keep your code neat and organized. They also save you time by letting you reuse bits of code you've already written. Plus, they make your code easier for others (or future you) to understand.

3. Types of Functions in Python: In Python, there are different kinds of functions. Some are already built into Python, like `print()` for showing stuff on the screen. Others you can make yourself with `def`.

4. Advantages of User-Defined Functions: When you make your own functions, you can write code once and use it many times. This makes your code easier to read and fix if something goes wrong.

5. Rules in Declaring a Function in Python: When you make a function, give it a name that makes sense, don't use words that Python already knows, and make sure your code lines up properly.

6. Python Function Syntax:

def function\_name(inputs):

"""description of what the function does"""

# The actual code for the function goes here

return output

7. Function Argument and Parameter: When you create a function, you can give it inputs. These are called parameters. When you use the function, you give it actual values, called arguments. Think of it like ordering food: the menu (function definition) lists the choices (parameters), and when you order (use the function), you tell the waiter what you want (arguments).

8. The Return Statement: When a function is finished doing its thing, it can give you back a result. That's what `return` does. It's like a cook giving you the meal you ordered.