**Python Programming (Meta)**

**Functional Programming**

* Functional programming uses a different paradigm than object-oriented programming and excels at processing large amounts of data at high speeds.
* There are two types of functions: traditional and pure. Pure functions always produce the same results for the same inputs, while traditional functions can modify global variables and change state.
* Functional programming does not change data outside the function's scope and aims to return the completed result instead of modifying input data.
* Functions in functional programming are considered standalone or independent, leading to clean and elegant code.
* Python treats functions as first-class citizens, allowing them to be assigned to variables, passed as arguments, or returned to callers.
* The "sorted" function in Python is an example of a built-in function that returns a sorted list from the input.
* Functional programming emphasizes reusable functions, saving development time.
* Custom functions can be created to perform specific tasks, as shown by the example of reversing the names of coffees using a user-defined "reverse" function.
* The "map" function in Python can apply a given function (like "reverse") to each item in a list automatically, simplifying iterations and transformations.

**Pure Functions in Functional Programming:**

* Functional programming uses a different paradigm than object-oriented programming and excels at processing large amounts of data at high speeds.
* There are two types of functions: traditional and pure. Pure functions always produce the same results for the same inputs, while traditional functions can modify global variables and change state.
* Functional programming does not change data outside the function's scope and aims to return the completed result instead of modifying input data.
* Functions in functional programming are considered standalone or independent, leading to clean and elegant code.
* Python treats functions as first-class citizens, allowing them to be assigned to variables, passed as arguments, or returned to callers.
* The "sorted" function in Python is an example of a built-in function that returns a sorted list from the input.
* Functional programming emphasizes reusable functions, saving development time.
* Custom functions can be created to perform specific tasks, as shown by the example of reversing the names of coffees using a user-defined "reverse" function.
* The "map" function in Python can apply a given function (like "reverse") to each item in a list automatically, simplifying iterations and transformations.