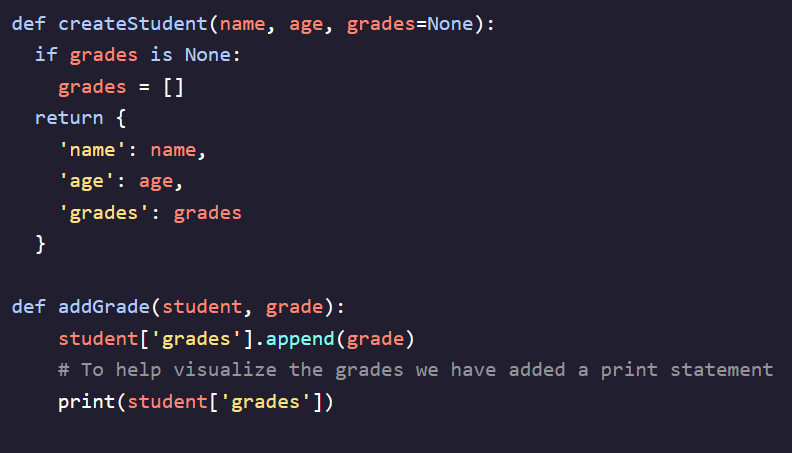
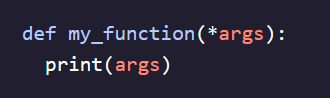
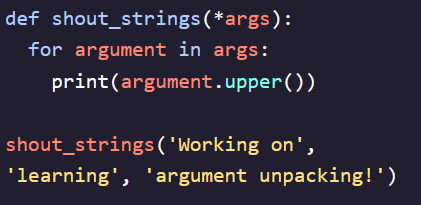
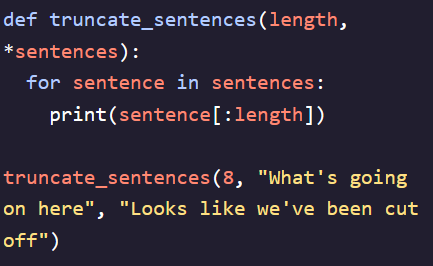
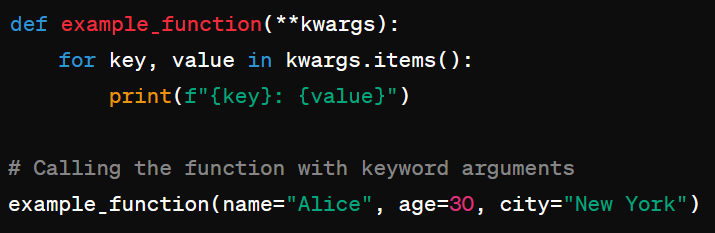
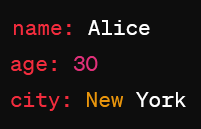
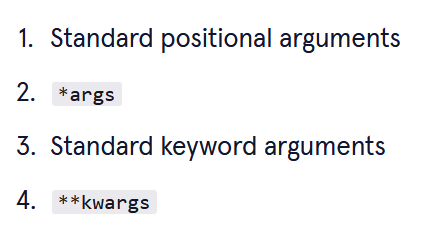
**Mutables and Functions:**

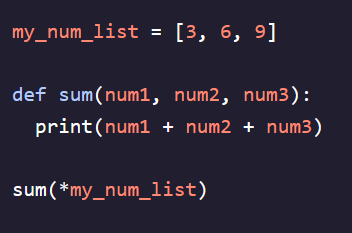
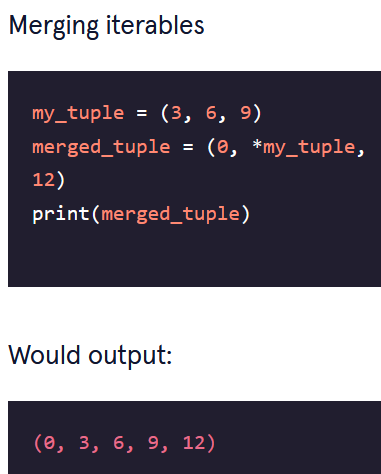
- It is a common mistake in Python to set a mutable object as the default value of an argument  
- A *gotcha* is a counterintuitive feature of a programming language that often leads to mistakes in programs  
- A *Mutable Object* – refers to various containers in Python that are intended to be changed (lists that we can append/delete, sets, and dictionaries)  
- *Immutable Objects* – tuples, int, float, strings

- When we call a function, the default value provided for parameters are only created once and used for each subsequent call of the function (anytime it is accessed later it is the same list that gets modified)   
- If we want to utilize an empty list as a potential default argument value we initiate it with “None” and write code to modify it   


- ***Unpacking Operator (\*)***– Allows us to give our functions a variable number of arguments by performing  ***Positional Argument Packing***- *args –* shorthand for arguments, can use \*args to store multiple arguments in a function (*args is arbitrary and can be replaced with any variable name)*- \* will store the arguments passed into the function as a tuple and allows it to accept any number of arguments  
   

*- \*\*kwargs* – Takes any keyword arguments that are not explicitly listed in the function definition and collects them into a dictionary named {kwargs}  
- Name is completely arbitrary and can be anything  
- *Keyword Arguments* – Used to pass arguments into a function by explicitly stating their parameter names (name = ‘Dave’)  
- Positional arguments **MUST** come before \*\*arguments  
- Using the *f* parameter stands for *format* and allows you to place placeholder values in strings  
- It’s a standard dictionary so can use all iteration methods you would with a dictionary  
 

- Order for using \**args* and \*\**kwargs* in one function definition  


- Can also use in function call to unpack arguments into specific variables/individual arguments  
 

- Can even use to unpack variables directly into built-in-python functions

