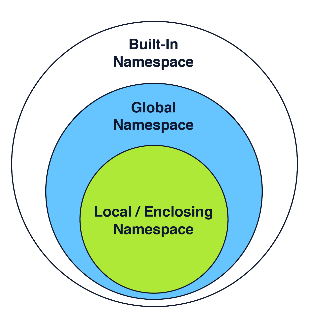
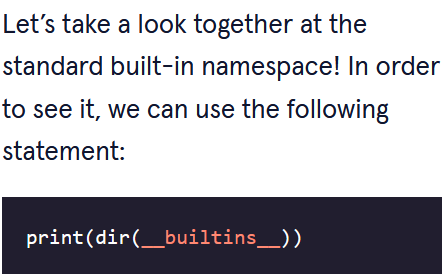
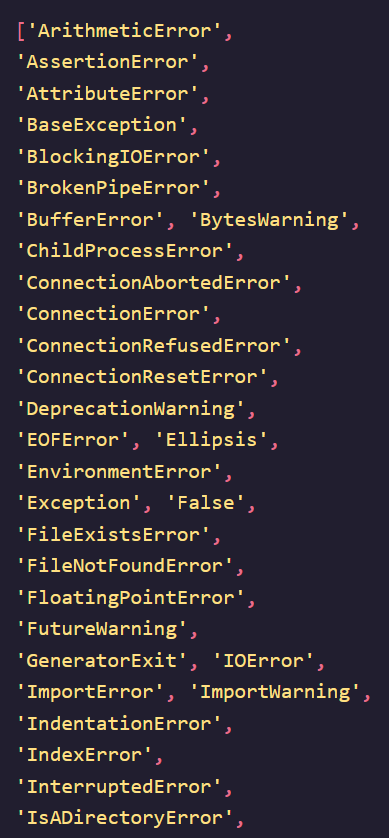
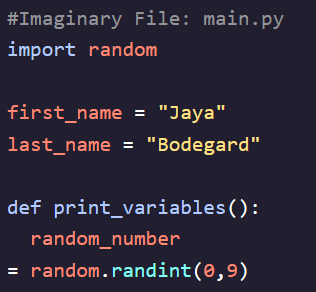
**Names and Namespaces:**

- *names* – the identifier for an object such as a variable or function (variable name – color = “cyan”)  
- *namespace* – A collection of names and the objects they reference  
 - Python creates a dictionary with the keys as the defined names and the mapped values as the objects they reference  
 - Python would search the namespace defined above for a key named color and provide the value to be run in our program. Thus we would see the output of  
    
  
**Built-In Namespace:**

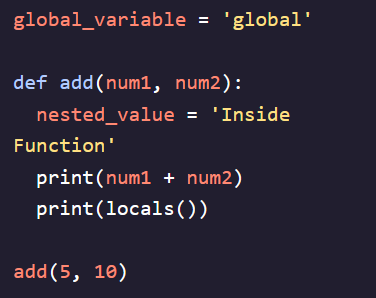
- One of the four main namespaces that exist within Python – called using *print(dir(\_\_builtins\_\_))*  
- Names in this level can be called in any program (*print(), str())*- Whenever we run a Python application, it provides a built-in namespace for the duration of the program. The objects in that namespace are universally accessible   
  

- Contains 152 universal names  
  
- Contains many of Pythons built-in functions  
  
- Contains many built in exceptions/errors  
  
- Contains constants like True and False

**Global Namespace:**

- Contains all non-nested variable names assigned in program (first line) – called using *globals()*  
- In this example: first\_name, last\_name, and print\_variables are global namespaces but random\_number is not because it is nested within the print\_variables function  
- Also includes namespaces for any imported modules: import random  


**Local Namespace:**

- These are namespaces that exist only within a function (the nested values) – called using *locals()*  
- In the example below they comprise: num1, num2, and nested\_value  
- Must be called nested inside of a function otherwise will function as a *global()* call  
- Only exist within a function until it is done executing  


**Enclosing Namespace:**

- A special type of local namespace that encompasses functions (*enclosed functions)* nested inside of functions *(enclosing functions)* – called using *locals()*- In the example below it encompasses: inner\_function  
- Must be called nested inside of a function otherwise will function as a *global()* call  
- Only exist within a function until it is done executing  
