

# Scope





# Scope

## WHAT IS IT??

Every variable we work with in Python has a scope or boundary where it can be used. There are specific rules to how variables are scoped based on where they are initially defined.

A large, bold, yellow letter 'L' with a simple, blocky design.

Lexical

A large, bold, red-orange letter 'E' with a simple, blocky design.

Enclosing

A large, bold, red letter 'G' with a simple, blocky design.

Global

A large, bold, dark red letter 'B' with a simple, blocky design.

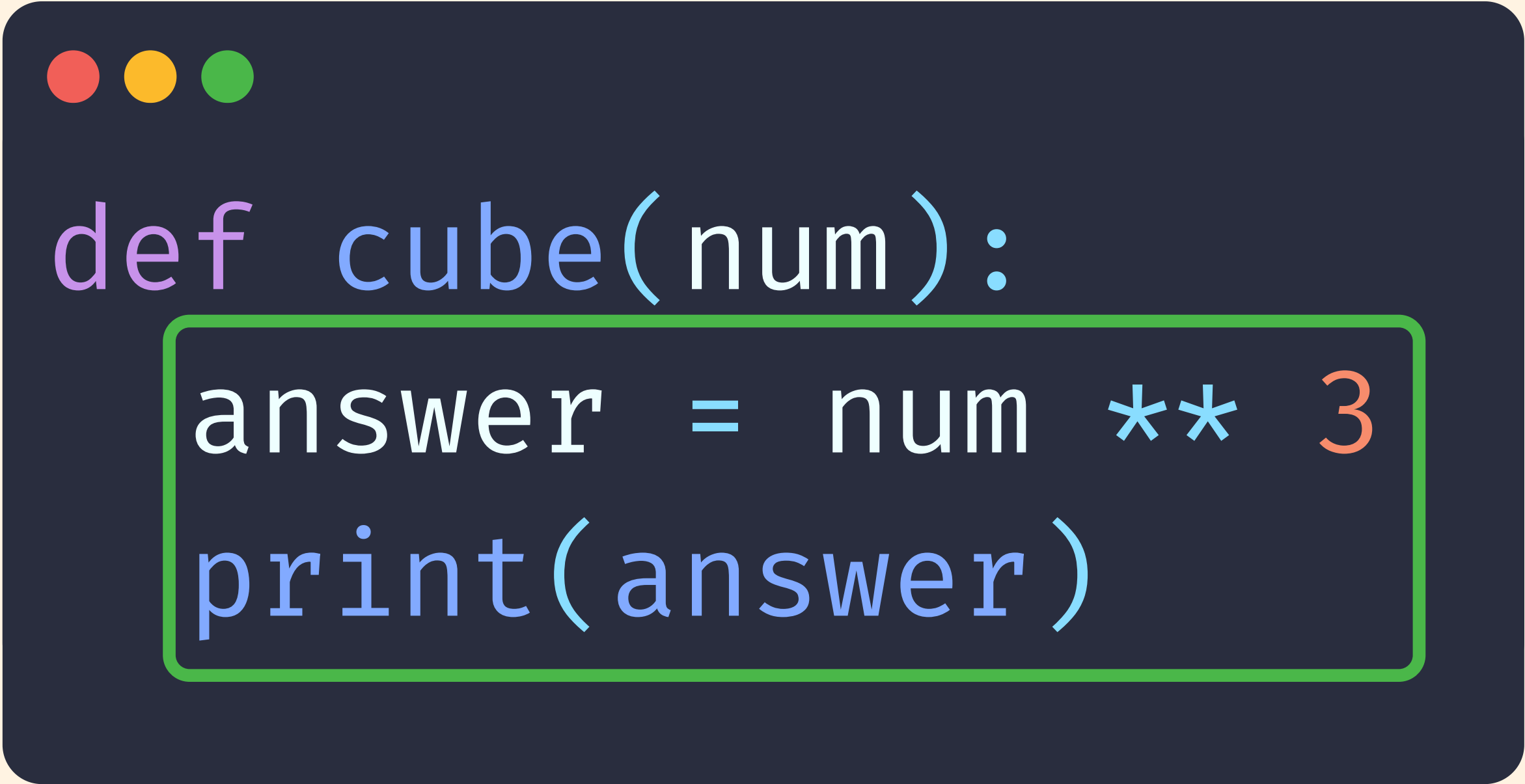
Built-In

# Global Scope

```
movie = "Amadeus"  
def review():  
    print(movie)  
    def inner():  
        print(movie)
```

Variables declared outside of functions are in the global scope. All functions have access to them.

# Local Scope



```
def cube(num):  
    answer = num ** 3  
    print(answer)
```

Variables defined in a function are **scoped to that function**.  
They are not available outside that function!

# Enclosing Scope

```
def outside():  
    a = 10  
    def inside():  
        print("a is: ", a)  
    inside()
```

Nested "inner" functions have access to variables declared in outer parent functions

# Built-In Scope



```
str("5")
```

All the built-in objects in Python are in the Built-In Scope. We have access to them anywhere!

A large, bold, yellow letter 'L'.

Lexical

A large, bold, red-orange letter 'E'.

Enclosing

A large, bold, red letter 'G'.

Global

A large, bold, dark red letter 'B'.

Built-In

