local and global variables

Before we dive into the differences between local and global variables, let's first recall what a variable is in Python.

A variable is a named location in memory that stores a value. In Python, we can assign values to variables using the assignment operator =. For example:

```
x = 5
y = "Hello, World!"
```

Now, let's talk about local and global variables.

A local variable is a variable that is defined within a function and is only accessible within that function. It is created when the function is called and is destroyed when the function returns.

On the other hand, a global variable is a variable that is defined outside of a function and is accessible from within any function in your code.

Here's an example to help clarify the difference:

```
x = 10 # global variable

def my_function():
  y = 5 # local variable
  print(y)

my_function()
print(x)
print(y) # this will cause an error because y is a local variable and is not accessible outside of the function
```

In this example, we have a global variable x and a local variable y. We can access the value of the global variable x from within the function, but we cannot access the value of the local variable y outside of the function.

The global keyword

Now, what if we want to modify a global variable from within a function? This is where the global keyword comes in.

The global keyword is used to declare that a variable is a global variable and should be accessed from the global scope. Here's an example:

```
x = 10 # global variable

def my_function():
 global x
 x = 5 # this will change the value of the global variable x
 y = 5 # local variable

my_function()
print(x) # prints 5
print(y) # this will cause an error because y is a local variable and is not accessible outside of the function
```

In this example, we used the global keyword to declare that we want to modify the global variable x from within the function. As a result, the value of x is changed to 5.

It's important to note that it's generally considered good practice to avoid modifying global variables from within functions, as it can lead to unexpected behavior and make your code harder to debug.

I hope this tutorial has helped clarify the differences between local and global variables and how to use the global keyword in Python.