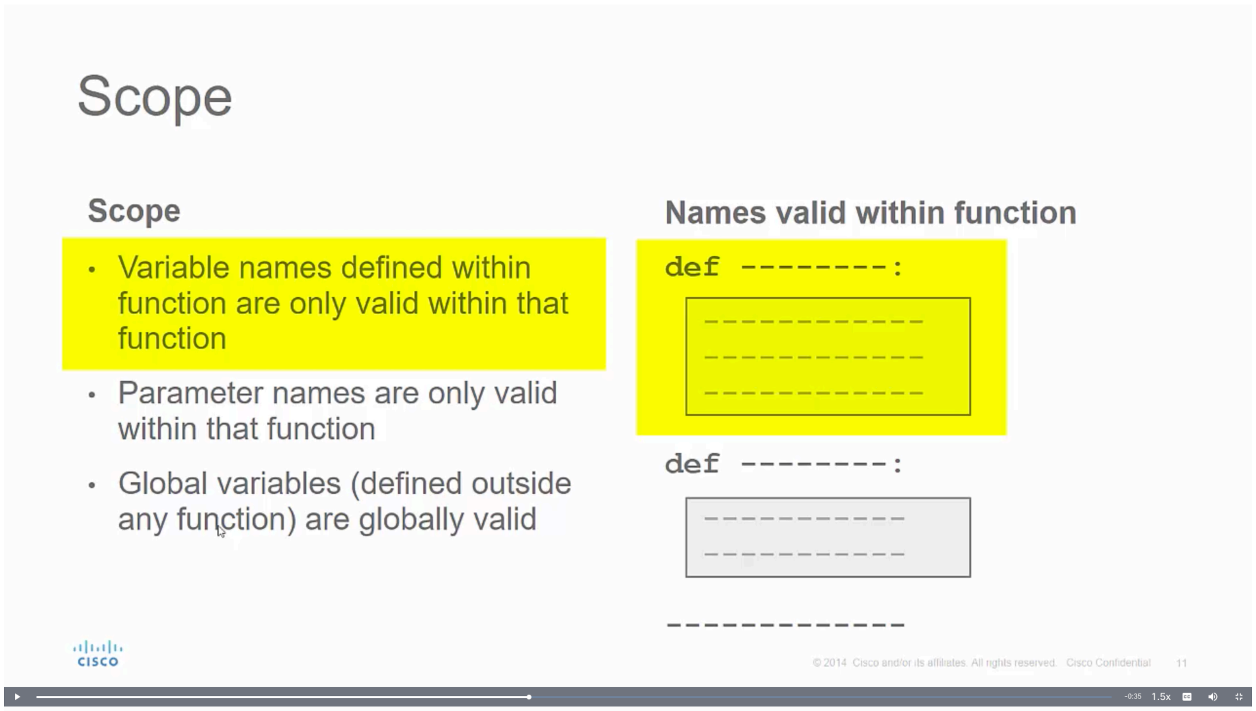


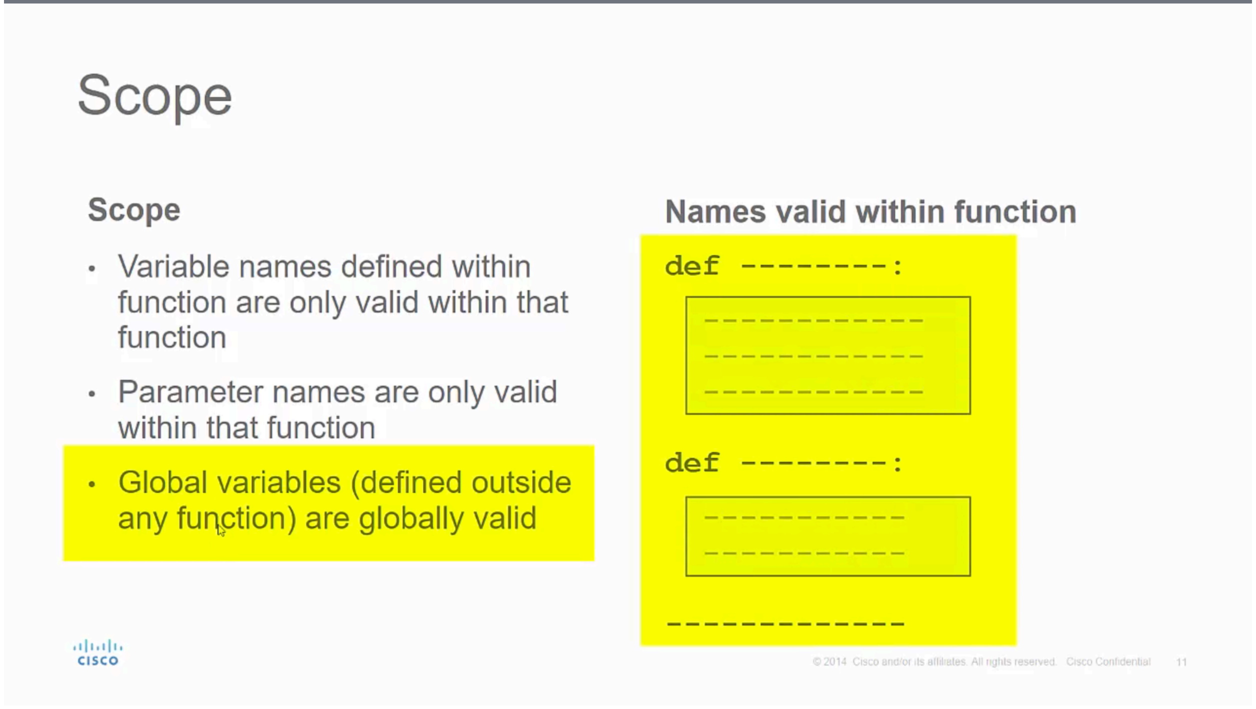
## Scope

**Scope**

- Variable names defined within function are only valid within that function
- Parameter names are only valid within that function
- Global variables (defined outside any function) are globally valid

**Names valid within function**

```
def --------:
    ------------
    ------------
    ------------
def --------:
    -----------
    -----------
-------------
```



## Scope

**Scope**

- Variable names defined within function are only valid within that function
- Parameter names are only valid within that function
- Global variables (defined outside any function) are globally valid

**Names valid within function**

```
def --------:
    ------------
    ------------
    ------------
def --------:
    -----------
    -----------
-------------
```

© 2014 Cisco and/or its affiliates. All rights reserved. Cisco Confidential 11

The idea of 'scope' when dealing with software involves the question of visibility. For example:

- If a variable is defined inside a function, it is not visible to code outside the function.
- If a variable is defined outside of a function, it is visible to code inside a function if the variable is global.
- A variable 'global' if it is defined outside of any function.

What this means is that variables that are defined within a function are only visible inside the function. If you want your data to be known outside the function, you should return it as a return value.

If you define a global variable, it is visible to all other code in the same module. However, good programming practice dictates that you try not to overuse global variables, and accessing them within functions is not generally a good idea. A better solution is to pass the necessary values into the function as parameters.

It is important to structure your application in such a way as to make it easy to distinguish the following items:

- **Functions:** Functions should be defined at the beginning of your application or module.
- **Main code:** Main code should be placed at the bottom of your application file.
- **Global variables:** Global variables are accessible from all code. For readability, global variables should either be initially defined (for example, set to an empty list, dictionary, and so on) at the very top of your application file, or at the very top of your main code.

The following is an example of how you may want to structure your application, now that you are using functions and creating applications that are larger and require some level of organization. Application file:

```
#--- func1 ------------------------------------------------------------
def func1(...) # function definition for func1

#--- func2 ------------------------------------------------------------
def func2(...) # function definition for func2

#--- main -------------------------------------------------------------
global_list = []  # global variables outside of function scope

...
x = func1(...)
---
y = func2(...)
```

Eventually some things will come down to personal preference. But whatever your preference may be, make sure that it is easy to read and understand, for your own benefit, and for the benefit of others who may come after and wish to maintain or make use of your code.