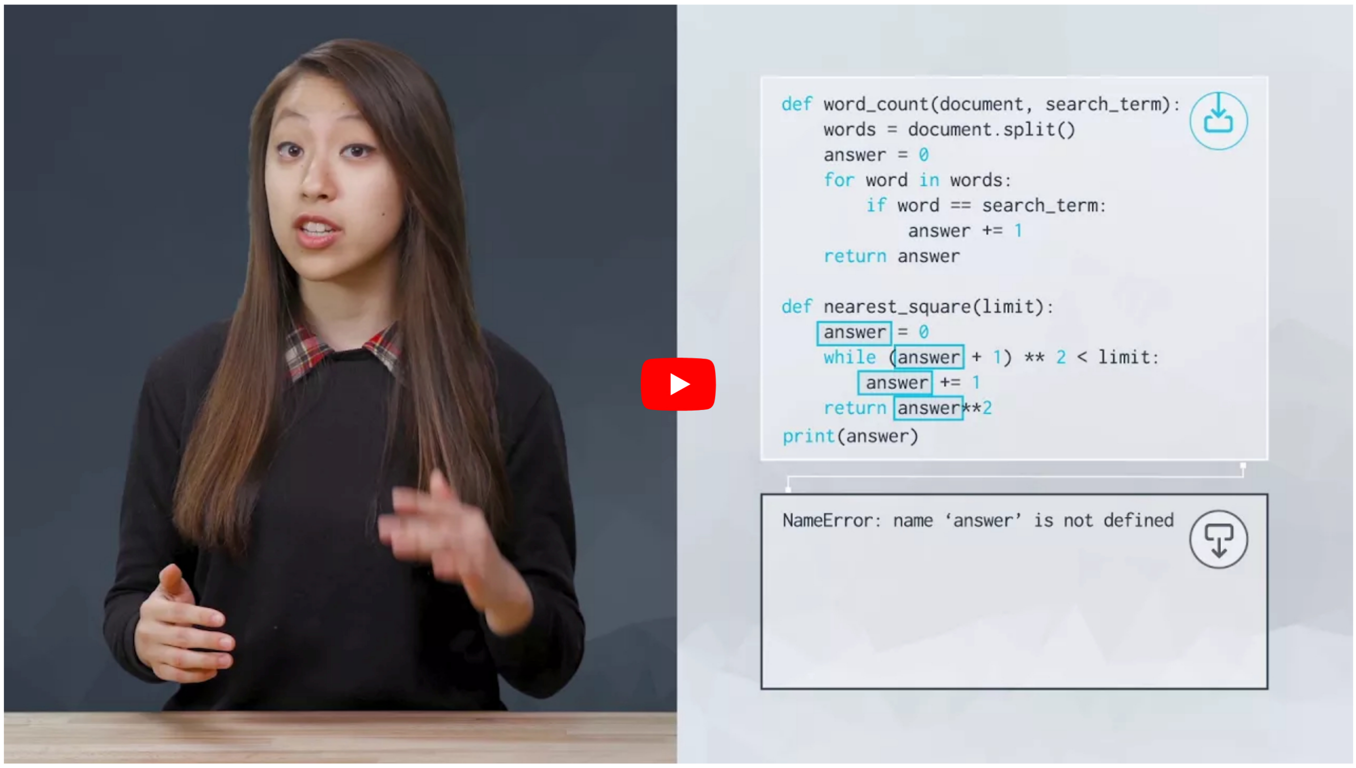
Variable Scope



```
def word_count(document, search_term):
    words = document.split()
    answer = 0
    for word in words:
        if word == search_term:
            answer += 1
    return answer

def nearest_square(limit):
    answer = 0
    while (answer + 1) ** 2 < limit:
        answer += 1
    return answer**2
print(answer)
```

```
NameError: name 'answer' is not defined
```

Variable Scope

Variable scope refers to which parts of a program a variable can be referenced, or used, from.

It's important to consider scope when using variables in functions. If a variable is created inside a function, it can only be used within that function. Accessing it outside that function is not possible.

```
## This will result in an error
def some_function():
    word = "hello"

print(word)
```

In the example above and the example below, `word` is said to have scope that is only **local** to each function. This means you can use the same name for different variables that are used in different functions.

```
## This works fine
def some_function():
    word = "hello"

def another_function():
    word = "goodbye"
```

Variables defined outside functions, as in the example below, can still be accessed within a function. Here, `word` is said to have a **global scope**.

```
## This works fine
word = "hello"

def some_function():
    print(word)
    print(word)

some_function()```
Notice that we can still access the value of the global variable `word` within this function. However, the value of a global

Scope is essential to understanding how information is passed throughout programs in Python and really any programming languag
```

More on Variable Scope

When you program, you'll often find that similar ideas come up again and again. You'll use variables for things like counting, iterating and accumulating values to return. In order to write readable code, you'll find yourself wanting to use similar names for similar ideas. As soon as you put multiple piece of code together (for instance, multiple functions or function calls in a single script) you might find that you want to use the same name for two separate concepts.

Fortunately, you don't need to come up with new names endlessly. Reusing names for objects is OK as long as you keep them in separate scope.

Good practice: It is best to define variables in the smallest scope they will be needed in. While functions *can* refer to variables defined in a larger scope, this is very rarely a good idea since you may not know what variables you have defined if your program has a lot of variables.

← Previous Next →

Give Page Feedback