

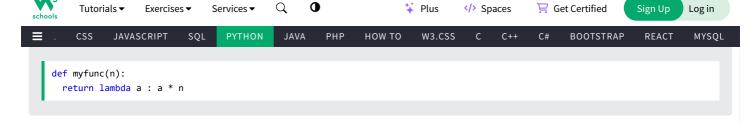
Lambda functions can take any number of arguments:

```
Example
Multiply argument a with argument b and return the result:
  x = lambda a, b : a * b
  print(x(5, 6))
  Try it Yourself »
```

```
Example
Summarize argument a, b, and c and return the result:
  x = lambda a, b, c : a + b + c
  print(x(5, 6, 2))
  Try it Yourself »
```



05/07/24, 14:28 Python Lambda



Use that function definition to make a function that always doubles the number you send in:

## Example

```
def myfunc(n):
    return lambda a : a * n

mydoubler = myfunc(2)

print(mydoubler(11))

Try it Yourself »
```

Or, use the same function definition to make a function that always triples the number you send in:

## Example

```
def myfunc(n):
    return lambda a : a * n

mytripler = myfunc(3)

print(mytripler(11))

Try it Yourself »
```

Or, use the same function definition to make both functions, in the same program:

## Example

```
def myfunc(n):
    return lambda a : a * n

mydoubler = myfunc(2)
mytripler = myfunc(3)

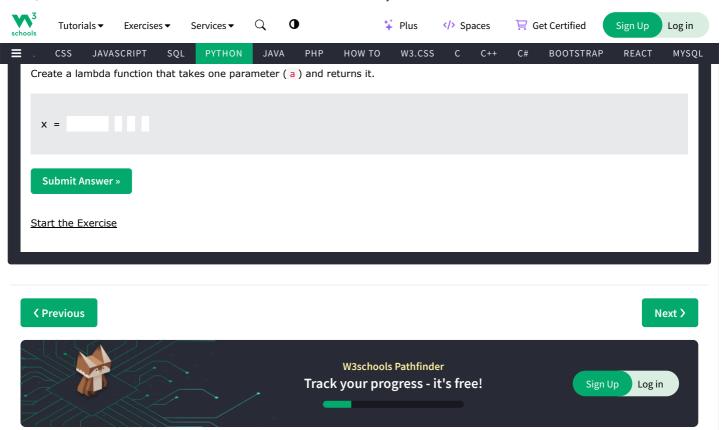
print(mydoubler(11))
print(mytripler(11))
Try it Yourself »
```

Use lambda functions when an anonymous function is required for a short period of time.

## **Test Yourself With Exercises**



05/07/24, 14:28 Python Lambda













05/07/24, 14:28 Python Lambda

