

- ✓ 5. Variable Scope
- ✓ 6. Quiz: Variable Scope
- ✓ 7. Solution: Variable Scope
- ✓ 8. Documentation
- ✓ 9. Quiz: Documentation
- ✓ 10. Solution: Documentation
- ✓ 11. Lambda Expressions
- ✓ 12. Quiz: Lambda Expressions
- ✓ 13. Solution: Lambda Expressions
- ✓ 14. [Optional] Iterators and Genera...
- ✓ 15. [Optional] Quiz: Iterators and G...
- ✓ 16. [Optional] Solution: Iterators a...
- ✓ 17. [Optional] Generator Expressions
- ✓ 18. Conclusion



[Optional] Solution: Iterators and Generators

Quiz Solution: Implement `my_enumerate`

```
lessons = ["Why Python Programming", "Data Types and Operators", "Control Flow", "Func
tions", "Scripting"]

def my_enumerate(iterable, start=0):
    count = start
    for element in iterable:
        yield count, element
        count += 1

for i, lesson in my_enumerate(lessons, 1):
    print("Lesson {}: {}".format(i, lesson))
```

Output:

```
Lesson 1: Why Python Programming
Lesson 2: Data Types and Operators
Lesson 3: Control Flow
Lesson 4: Functions
Lesson 5: Scripting
```

Quiz Solution: Chunker



```
def chunker(iterable, size):  
    """Yield successive chunks from iterable of length size."""  
    for i in range(0, len(iterable), size):  
        yield iterable[i:i + size]  
  
for chunk in chunker(range(25), 4):  
    print(list(chunk))
```

Output:

```
[0, 1, 2, 3]  
[4, 5, 6, 7]  
[8, 9, 10, 11]  
[12, 13, 14, 15]  
[16, 17, 18, 19]  
[20, 21, 22, 23]  
[24]
```

[NEXT](#)