# Practice Quiz: For Loops

**TOTAL POINTS 4**

1. **Fill in the blanks to make the factorial function return the factorial of n. Then, print the first 10 factorials (from 0 to 9) with the corresponding number. Remember that the factorial of a number is defined as the product of an integer and all integers before it. For example, the factorial of five (5!) is equal to 1*2*3*4*5=120. Also recall that the factorial of zero (0!) is equal to 1.**

1 / 1 point

```
1   def factorial(n):
2       result = 1
3       for x in range(1, 1+n):
4           result = result * x
5       return result
6
7   for n in range(10):
8       print(n, factorial(n))
9
```

Run

Reset

✓ **Correct**

> Great work! The pieces of code you're tackling keep getting more complex, you're doing a great job!

2. **Write a script that prints the first 10 cube numbers (x**3), starting with x=1 and ending with x=10.**

1 / 1 point

```
1
2   for x in range(1,11):
3     cube = x*x*x
4     print(cube)
```

Run

Reset

✓ **Correct**

> You nailed it! You got the code to print the first 10 cubes.

**3.** Write a script that prints the multiples of 7 between 0 and 100. Print one multiple per line and avoid printing any numbers that aren't multiples of 7. Remember that 0 is also a multiple of 7.

```
1   m = 0
2
3   for i in range(0,101, 7):
4
5     print(i)
6
7
8
```

Run

Reset

✓ **Correct**

Awesome! You're getting Python to do all the work for you.

**4.** The retry function tries to execute an operation that might fail, it retries the operation for a number of attempts. Currently the code will keep executing the function even if it succeeds. Modify the code so that it stops trying after the operation succeeded.

```
1   def retry(operation, attempts):
2
3   for n in range(attempts):
4     if operation():
5       print("Attempt " + str(n) + " succeeded")
6       break
7     else:
8         print("Attempt " + str(n) + " failed")
9
10  retry(create_user, 3)
11  retry(stop service, 5)
```

Run

Reset

✓ **Correct**

Well done, you! You've fixed the code to stop executing once the function is successful.