

# CSE 160: Nested Loops

## The range() Function

A typical 'for' loop does not use an explicit list:

```
for i in range(5):  
    ... body ...
```

`range(5)`: cycles through [0, 1, 2, 3, 4]  
where 5: Upper limit (exclusive)

`range(1, 5)`: cycles through [1, 2, 3, 4]  
where 1: Lower limit (inclusive)  
5: Upper limit (exclusive)

`range(1, 10, 2)`: cycles through [1, 3, 5, 7, 9]  
where 1: Lower limit (inclusive)  
10: Upper limit (exclusive)  
2: step (distance between elements)

## Introduction to for Loops

The for loop is used to iterate over a sequence (like a list, tuple, or string) or to repeat a block of code for a specified number of times. The general syntax for a for loop is:

```
for variable in sequence:  
    # code block to execute
```

- **variable**: This is the variable that takes the value of each item in the sequence during each iteration. - **sequence**: This is the data structure (like a list, range, or string) that the 'for' loop iterates over. - **Code Block**: The code inside the loop runs for each value in the sequence.

Example: Using a 'for' Loop with a List

```
fruits = ["apple", "banana", "cherry"]  
for fruit in fruits:  
    print(fruit)
```

This loop will print each fruit in the list on a new line.

Example: Using a 'for' Loop with the `range()` Function

The `range()` function is commonly used in for loops to repeat a block of code a specified number of times.

```
for i in range(3):  
    print("Iteration number", i)
```

This will print:

```
Iteration number 0  
Iteration number 1  
Iteration number 2
```

```
# Sum of a list of values
result = 0
for element in range(0, 10, 2):
    result = result + element
print("The sum is:", result)
```

Output(result):\_\_\_\_\_

```
# Sum of a list of values
result = 0
for element in range(5, 1, -1):
    result = result + element
print("The sum is:", result)
```

Output(result):\_\_\_\_\_

Answers: 20; 14

## Nested Loops

```
# Examine the code below, what values are printed??
for m in [1, 2, 3]:
    print("Before n loop, m is:", m)
    for n in [50, 100]:
        print("n is", n)
```

Output(result):\_\_\_\_\_

Try another one:

```
# How many statements does this loop contain?
for m in [0, 1]:
    print("Outer", m)
    for n in [2, 3]:
        print(" Inner", n)
        print(" Sum", m + n)
    print("Outer", m)
```

Output(result):\_\_\_\_\_

## Practice Problem (Fix It)

```
# Goal: print 1, 2, 3, ..., 28, 29, 30
for tens_digit in [0, 1, 2]:
    for ones_digit in [1, 2, 3, 4, 5, 6, 7, 8, 9]:
        print(tens_digit * 10 + ones_digit)
```

Question:

1. What does it actually print?
2. How can we change it to correct its output?

## Answers:

Answer `for` question one:

1, 2, 3, 4, 5, 6, 7, 8, 9

11, 12, 13, 14, 15, 16, 17, 18, 19

21, 22, 23, 24, 25, 26, 27, 28, 29

Answer `for` question two:

Add the number 10 after the number 9 at line 3.

## if Statements

### Basic Content:

An IF statement can have two results. The first result is if your comparison is True, and the second is if your comparison is False.

## Practice problems

```
# What is the output?  
val = 6  
if val > 0:  
    print("Positive")  
else:  
    print("Non-positive")  
  
Question:  
1. What does it actually print?
```

**Answer:**

Positive



## Nested if Statement

### Basic Structure:

```
if condition1:
    # Code to execute if condition1 is True
    if condition2:
        # Code to execute if condition1 and condition2
        # are True
    else:
        # Code to execute if condition1 is True but
        # condition2 is False
else:
    # Code to execute if condition1 is False
```

## Practice problems

```
# Nested if statement example
age = 20
has_permission = True

if age >= 18:
    print("You are an adult.")
    if has_permission:
        print("You can access this resource.")
    else:
        print("You don't have permission to access.")
else:
    print("You are a minor.")

Question:
1. What does it actually print?
2. Do you have any suggestions to improve this code?
```

## Answers:

Answer `for` problem one:  
You are an adult.  
You can `access` this resource.

Answer `for` problem two:  
Any suggestions