Homework Problems

5-1. Conditional Tests: Write **ten** conditional tests, half (5) should evaluate to `True` and the other half should evaluate to `False`. Print a statement describing each test and your prediction for the results of each test. Your code should look something like this:

```
car = 'subaru'
print("Is car =='subaru'? I predict True.")
print(car == 'subaru')

print("\nIs car =='audi'? I predict False.")
print(car == 'audi')
```

In [1]:
```
x = 1
print("Is x == 1? I predict True.")
print(x == 1)

print("Is x > 0? I predict True.")
print(x > 0)

print("Is x+1 > 1? I predict True.")
print(x+1 > 1)

print("Is x != 0? I predict True.")
print(x != 0)

print("Is x < 100? I predict True.")
print(x < 100)

print("Is x == 0? I predict False.")
print(x == 0)

print("Is x-1 > 0? I predict False.")
print(x-1 > 0)

print("Is x < 0? I predict False.")
print(x < 0)

print("Is x != 1? I predict False.")
print(x != 1)

print("Is x == 'a string'? I predict False.")
print(x == 'a string')
```

```
Is x == 1? I predict True.
True
Is x > 0? I predict True.
True
Is x+1 > 1? I predict True.
True
Is x != 0? I predict True.
True
Is x < 100? I predict True.
True
Is x == 0? I predict False.
False
Is x-1 > 0? I predict False.
False
Is x < 0? I predict False.
False
Is x != 1? I predict False.
False
Is x == 'a string'? I predict False.
False
```

5-2. More Conditional Tests: Write at least one test for each of the following conditions:

- Tests for equality and inequality with strings

- Tests using the `lower()` method
- Numerical test involving equality, greater than, less than, etc.
- Tests using the `and` keyword and the `or` keyword (use both in the same statement)
- Test whether an item is in a list
- Test whether an item is not in a list

In [2]:
```
name1 = "Andrew"
name2 = "andrew"

print("name1 == name2:", name1 == name2)
print("name1.lower() == name2.lower():", name1.lower() == name2.lower())

x = 1
y = 2
my_list = [1]

print("x < y:", x < y)
print("(x == 0 or x == 1) and y > x:", (x == 0 or x == 1) and y > x)
print("x in my_list:", x in my_list)
print("y not in my_list:", y not in my_list)
```

```
name1 == name2: False
name1.lower() == name2.lower(): True
x < y: True
(x == 0 or x == 1) and y > x: True
x in my_list: True
y not in my_list: True
```

5-3. Alien Colors #1: Imagine an alient was just shot down in a game. Create a variable called `alien_color` and assign it a value of `'green'`, `'yellow'`, or `'red'`.

- Write an `if` statement to test whether the alient's color is green. If it is, print a message that the player just earned 5 points.
- Change the value of `alien_color` so the `if` test fails. (The version that fails will have no output)

In [3]:
```
alien_color = 'green'
if alien_color == 'green':
    print("You just earned 5 points!")

alien_color = 'red'
if alien_color == 'green':
    print("You just earned 5 points!")
```

```
You just earned 5 points!
```

5-4. Alien Colors #2: Set `alien_color` a value of your choice, then write an `if-else` chain as follows:

- If the alien's color is green, print a statement that the player just earned 5 points for shooting the alien.
- If the alien's color isn't green, print a statement that the player just earned 10 points.

- Change the value of `alien_color` and make sure you understand the output.

In [4]:
```
alien_color = 'yellow'

if alien_color == 'green':
    print("You just earned 5 points for shooting the alien!")

else:
    print("You just earned 10 points!")
```

```
You just earned 10 points!
```

5-5. Alien Colors #3: Turn the `if-else` chain above into an `if-elif-else` chain as follows:

- If the alien is green, print a message that the player earned 5 points.
- If the alien is yellow, print a message that the player earned 10 points.
- If the alien is red, print a message that the player earned 15 points.
- Change the value of `alien_color` and run the code multiple times. Make sure you understand the output

In [5]:
```
alien_color = 'red'

if alien_color == 'green':
    print("You just earned 5 points for shooting the alien!")

elif alien_color == 'yellow':
    print("You just earned 10 points!")

else:
    print("You just earned 15 points!")
```

```
You just earned 15 points!
```

5-6. States of Life: Write an `if-elif-else` chain that determines a person's stage of life. Set a value for the variable `age`, and then:

- If the person is less than 2 years old, print a message the the person is a baby.
- If the person is at least 2 years old but less than 4, print a message that the person is a toddler.
- If the person is at least 4 years old but less than 13, pint a message that the person is a kid.
- If the person is at least 13 years old but less than 20, print a message that the person is a teenager.
- If the person is at least 20 years old but less than 65, print a message that the person is an adult.
- If the person is age 65 or older, print a message that the person is an elder.

In [6]:
```
age = 65

if age < 2:
    print("You are a baby")
elif 2 <= age < 4:
    print("You are a toddler")
elif 4 <= age < 13:
    print("You are a kid")
elif 13 <= age < 20:
    print("You are a teenager")
elif 20 <= age < 65:
    print("You are an adult")
else:
    print("You are an elder")
```

```
You are an elder
```

5-7. Favorite Fruit: Make a list of your favorite fruits (at least 4 items). Then write at least three independent `if` statements that check for certain fruits in your list. If the fruit is in your list, the `if` block should print a statement, such as *You really like bananas!*

In [7]:
```
fruits = ['watermelon', 'apple', 'peach', 'orange']
if 'watermelon' in fruits:
    print("You really like watermelons!")
if 'blueberries' in fruits:
    print("You really like blueberries!")
if 'peach' in fruits:
    print("You really like peaches!")
```

```
You really like watermelons!
You really like peaches!
```

5-8. Hello Admin: Make a list of five usernames, including the name `'admin'`. Imagine you are writing code that will print a greeting to each user after they log in to a websie. Loop through the list, and print a greeting to each user:

- If the username is `'admin'`, print a speacial greeting, such as *Hello admin, would you like to see a status report?*
- Otherwise, print a generic greeting, such as *Hello Jaden, thank you for logging in again.*

In [8]:
```
usernames = ['admin', 'andrew', 'dug', 'david']
for name in usernames:
    if name == 'admin':
        print(f"Hello admin, would you like to see a status report?")
    else:
        print(f"Hello {name}, thank you for logging in again.")
```

```
Hello admin, would you like to see a status report?
Hello andrew, thank you for logging in again.
Hello dug, thank you for logging in again.
Hello david, thank you for logging in again.
```

5-9. No Users: Copy the code above and add an `if` test to make sure the list of users is not empty.

- If the list is empty, print the message *We need to find some users!*
- Remove all of the usernames from your list, and make sure the correct message is printed

In [9]:
```
usernames = []
if not usernames:
    print("We need to find some users!")
else:
    for name in usernames:
        if name == 'admin':
            print(f"Hello admin, would you like to see a status report?")
        else:
            print(f"Hello {name}, thank you for logging in again.")
```

```
We need to find some users!
```

5-10. Checking Usernames: Do the following to create a program that simulates how websites ensure that everyone has a unique username.

- Make a list of five or more usernames called `current_users` .
- Make another list of five usernames called `new_users` . Make sure one or two of the new usernames are also in the `current_users` list.
- Loop through the `new_users` list to see if each new username has already been used. If it has, print a message that the person will need to enter a new username. If a username has not been used, print a message saying that the username is available.
- Make sure your comparison is case insensitve. If `'John'` has been used, `'JOHN'` should not be accepted.

In [10]:
```
current_users = ['ANDREW', 'ryan', 'TYLER', 'thomas', 'Catherine', 'remi']
new_users = ['maeve', 'Andrew', 'Josh', 'catherine', 'connor']

for x in range(len(current_users)):
    current_users[x] = current_users[x].lower()

for user in new_users:
    if user.lower() in current_users:
        print(f"Sorry, {user}, is already taken. Please enter a new username"
    if user.lower() not in current_users:
        print(f"Yay! {user} is available!")
```

```
Yay! maeve is available!
Sorry, Andrew, is already taken. Please enter a new username
Yay! Josh is available!
Sorry, catherine, is already taken. Please enter a new username
Yay! connor is available!
```

5-11. Ordinal Numbers: Ordinal numbers indicate their position in a list, such as *1st* or *2nd*. Most ordinal numbers end in *th*, except 1, 2, and 3.

- Store the numbers 1 through 9 in a list.
- Loop through the list.
- Use an `if-elif-else` chain inside the loop to print the proper ordinal ending for each number. Your output should read "1st 2nd 3rd 4th 5th 6th 7th 8th 9th", and each result should be on a separate line.

In [11]:
```
numbers = list(range(1,10))
for number in numbers:
    if number == 1:
        print("1st")
    elif number == 2:
        print("2nd")
    elif number == 3:
        print("3rd")
    else:
        print(f"{number}th")
```

```
1st
2nd
3rd
4th
5th
6th
7th
8th
9th
```

5-12: Styling `if` statements: Review your code above and make sure you styled your conditional tests appropriately

5-13: Your ideas: Write down at least three ideas for programs you would like to write (games, websites, applications, etc).

- Drone
- Autonomous Car
- Robotic Arm