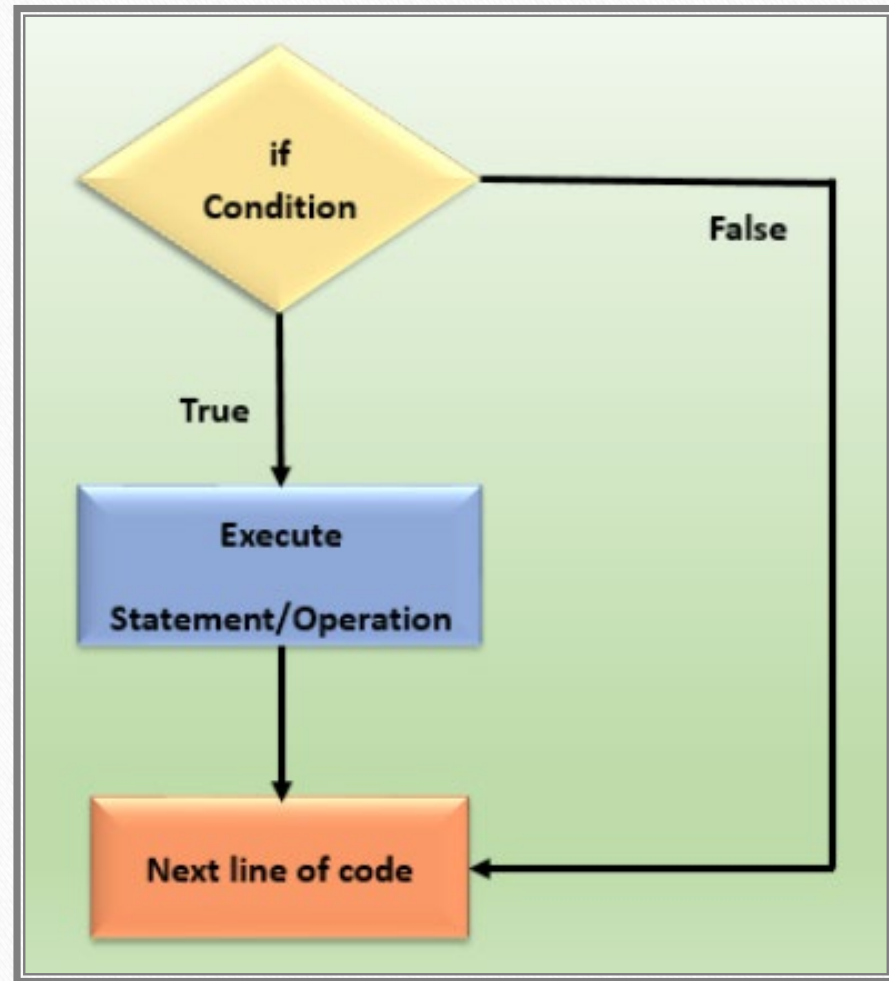


# Conditional Statements and Loops

---

## If Statement

Execute a certain block of code only if a certain condition is true.



Operator	Name	Example
==	Equal	5 == 5
!=	Not equal	26 != 3
>	Greater than	100 > 67
<	Less than	89 < 216
>=	Greater than or equal to	90 >= 54
<=	Less than or equal to	23 <= 77

# Comparison Operators

---



# If Statement

Execute a certain block of code only if a certain condition is true.

```
x=6
```

```
if x>5:  
    print(x)
```

Conditional: must  
return boolean

“:” ends statement

If statements always begin  
with an “if”

If condition is true, then indented line of  
code will run

# If Statement

Execute a certain block of code only if a certain condition is true.

```
x=6
```

```
if x>5:  
    print(x)
```

```
x = 6  
#is x greater than 5?  
x>5
```

True

```
x = 6  
#is x greater than 5?  
if x>5:  
    print(x)
```

6

## If + Else Statement

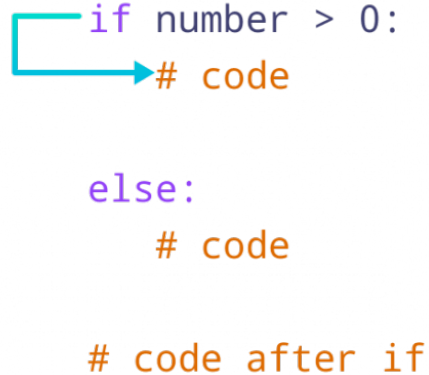
- Else statement allows us to specify separate directions if the condition with the if is False.

### Condition is True

```
number = 10
if number > 0:
    # code

else:
    # code

# code after if
```

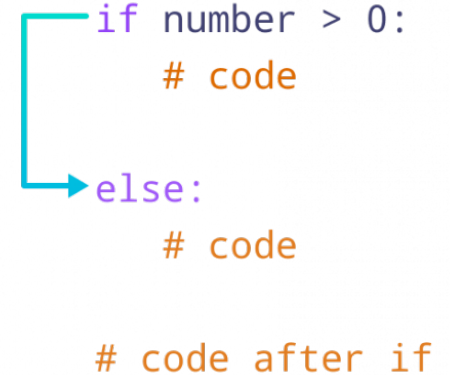


### Condition is False

```
number = -5
if number > 0:
    # code

else:
    # code

# code after if
```





# If + Else Statement

Else statement allows us to specify separate directions if the condition with the if is False.

```
x = 4
#is x greater than 5?
if x>5:
    print(x)
else:
    print("Below 5")
```

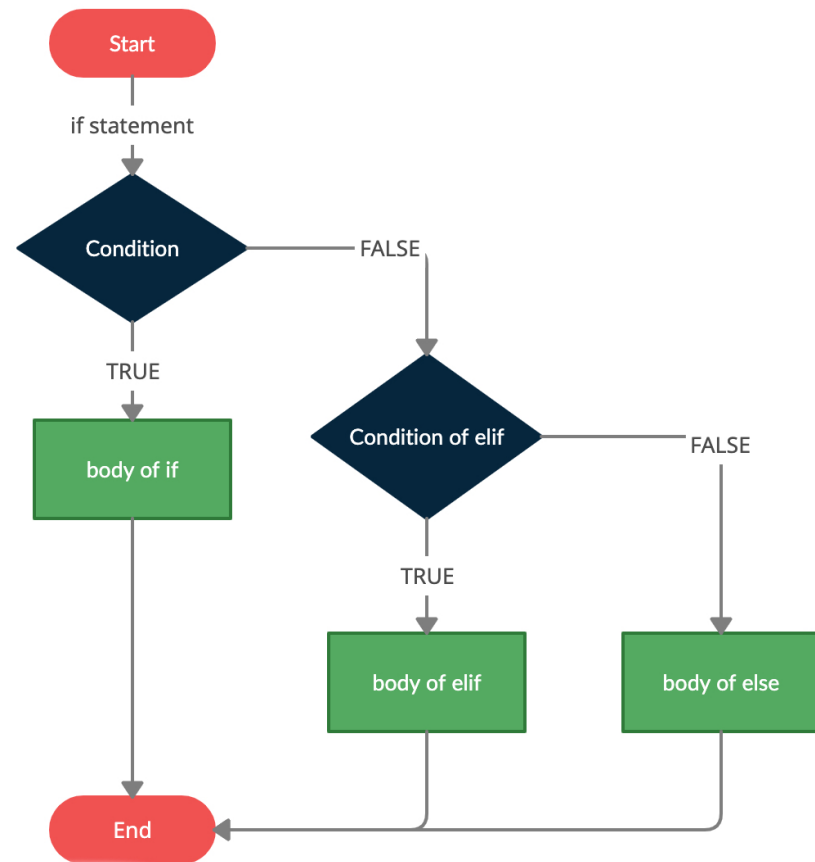
Else **always** executes when the if condition is False.

Match if + else by indentation

Else is optional, meaning an if statement doesn't have to have an accompanying else.

# If + Elif + Else Statement

How do I specify multiple conditions?...





# If + Elif + Else Statement

---

```
x = 'killer rabbit'
▼ if x == 'roger':
    print("hello jessica")
▼ elif x == 'bugs':
    print("What's up doc")
▼ else:
    print("Run!!!!")
```

- Comes between if and else
- Can have arbitrary number of elifs
- First if/elif condition that is true is executed and only this block is executed
  - If no if/elif condition is true then else is executed
- Indentation matches up if/elif/else statements

## If + Elif Statement

---

We can just have an if + elif, with no else:

```
x = 'killer rabbit'
▼ if x == 'roger':
    print("hello jessica")
▼ elif x == 'bugs':
    print("What's up doc")
```

# If + Elif + Else Statement

## 1st Condition is True

```
let number = 5
if number > 0 :
    # code
elif number < 0 :
    # code
else :
    # code
# code after if
```

A flowchart illustrating the execution of the 'if' statement when the first condition is true. A blue arrow starts from the 'if' line and points to the '# code' block. Another blue arrow starts from the 'elif' line and points to the '# code after if' block, bypassing the 'elif' and 'else' blocks.

## 2nd Condition is True

```
let number = -5
if number > 0 :
    # code
elif number < 0 :
    # code
else :
    # code
# code after if
```

A flowchart illustrating the execution of the 'if' statement when the second condition is true. A blue arrow starts from the 'if' line and points to the '# code' block. Another blue arrow starts from the 'elif' line and points to the '# code after if' block, bypassing the 'elif' and 'else' blocks.

## All Conditions are False

```
let number = 0
if number > 0 :
    # code
elif number < 0 :
    # code
else :
    # code
# code after if
```

A flowchart illustrating the execution of the 'if' statement when all conditions are false. A blue arrow starts from the 'if' line and points to the '# code' block. Another blue arrow starts from the 'else' line and points to the '# code after if' block, bypassing the 'else' block.



# If + Elif + Else Statement



```
if x == 'roger':  
    print("hello jessica")
```



```
elif x == 'bugs':  
    print("What's up doc")
```



```
elif season == "winter":  
    print("no bunnies")
```

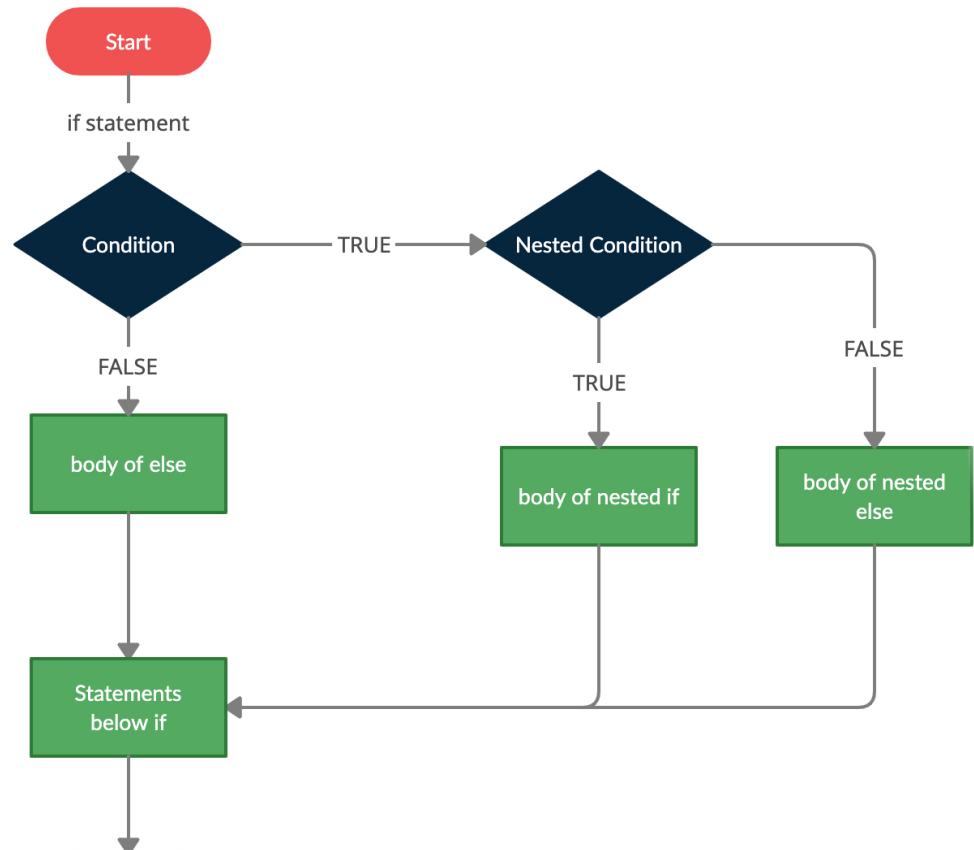


```
else:  
    print("Run!!!!")
```

no bunnies

# Nested Conditionals

We can have if statements nested within if statements:



# Nested Conditionals

---

We can have if  
statements nested  
within if  
statements:

```
▼ #Nested if statement  
age = 22  
activity = "eat"  
  
▼ if age >=21:  
▼     if activity == 'eat':  
        print("food menu")  
▼     elif activity == "drinks":  
        print("drink menu")  
▼     else:  
        print("wrong place")  
▼ else:  
    print("You are underaged")
```



## Specifying Multiple Conditions

- How do you specify multiple conditions in a conditional statements?

- not

x	not x
False	True
True	False

- and

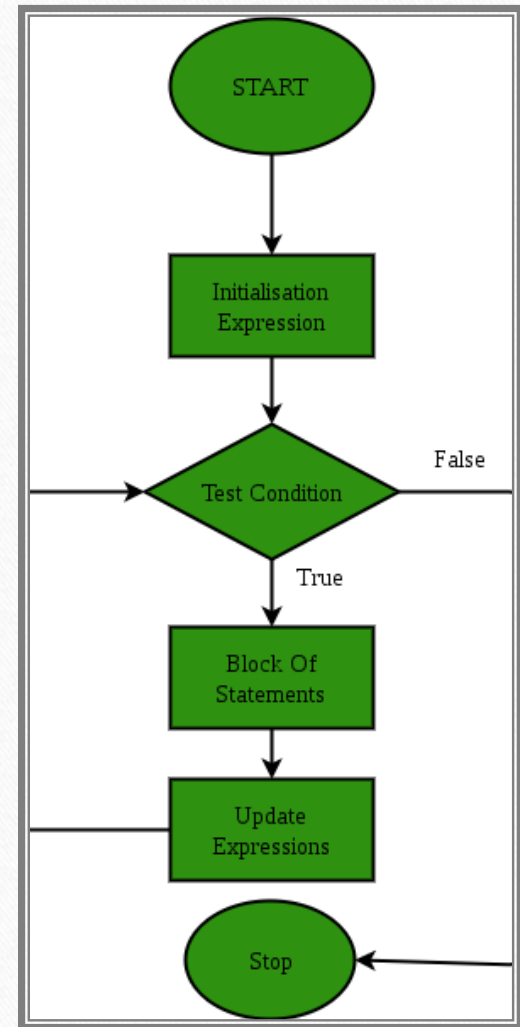
x	y	x and y
False	False	False
False	True	False
True	False	False
True	True	True

- or

x	y	x or y
False	False	False
False	True	True
True	False	True
True	True	True

# Loop Statement

Execute a certain block of code continuously until a certain condition is not true.



# For Loops

Use cases of  
for loops

Iterate  
over

Iterate over the elements of a list or string

- For each element, perform some sort of operation - count or sum

Perform

Perform some action a specified number of times

- Deal out 5 cards
- Have a student go through all 100 lockers



# For Loops w/ List

---

- Compute sum of elements of a list
- Use for loop to iterate over the elements of the list

```
list_nums = [2,4,6,8]
total = 0

for num in list_nums:
    total+=num
```

# For Loops w/ List

```
list_nums = [2,4,6,8]  
total = 0
```

```
for num in list_nums:  
    total+=num
```

Keyword “for”

Loop variable – takes  
on each value of loop  
target

Loop target – object  
to be iterated over

# For Loops w/ Strings

---

```
sentence = "Charlie likes walks"  
count = 0  
for c in sentence:  
    if c == " ":  
        count+=1  
  
total_words = count+1
```

- Now the loop target is a string
- Compute the number of words in sentence



# Summary

For Loops  
w/ Strings  
or Lists

The loop target can be a string or a list

- If the loop target is a list – iterate over the elements of the list by increasing index
- If the loop target is a string – iterate over the characters of the string by increasing index

# Range

- Can use range(0,n) as loop target
- Need consecutive integers
- Need to repeat something n times

Built-in range  
function

```
#Using range
range(0, 5)

list(range(5))

[0, 1, 2, 3, 4]
```

start (defaults to 0 if  
left off)

end (non-inclusive)

# For Loops w/ Range

---

```
even_nums = []  
for i in range(5):  
    if i%2==0:  
        even_nums+= [i]
```

- Using range() as loop target to get consecutive integers
- Compute even numbers  $\geq 0$



# For Loops w/ Range

How do I iterate over the indices of a list using a for loop?

```
L = [3, "S", 5, [7, 8, 9]]
```

```
▼ for i in range(len(L)):  
    print(L[i])
```

```
3  
S  
5  
[7, 8, 9]
```

# Break Statement

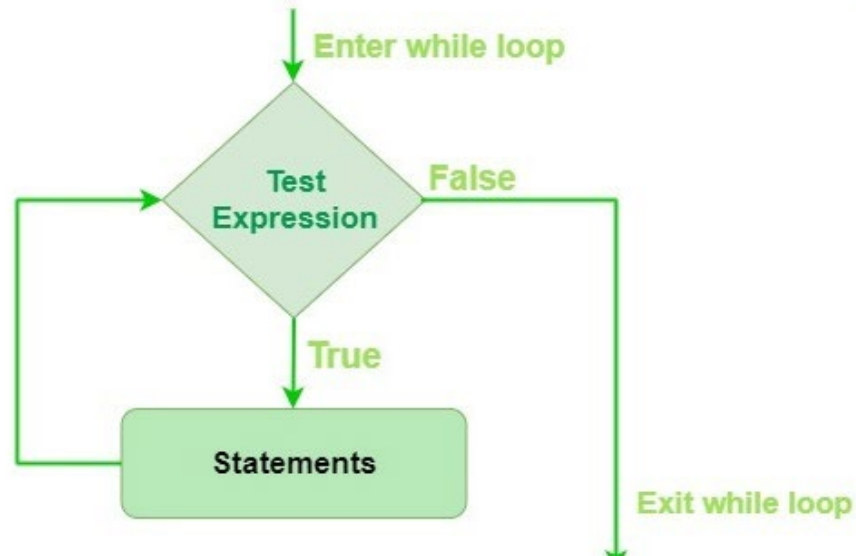
---

Jump out of nearest loop

- Check if x is prime – just need to find one factor

```
x= 15
flag=0
for i in range(2,x):
    if x%i ==0:
        flag=1
        mult = i
        break

if flag==0:
    print("%d is prime" %x)
else:
    print("%d is multiple of %d" %(mult, x))
```







# While Loops

---



# While Loop

Python3

 `# Python program to illustrate`  
 `# while loop`  
`count = 0`  
 `while (count < 3):`  
 `count = count + 1`  
  `print("Hello Geek")`

**Output**

```
Hello Geek
Hello Geek
Hello Geek
```

# While Loop with List

---

Python3

```
# checks if list still  
# contains any element  
a = [1, 2, 3, 4]  
  
while a:  
    print(a.pop())
```

```
4  
3  
2  
1
```

# While Loop with Control Statement

Python3

```
# break the loop as soon it sees 'e'  
# or 's'  
i = 0  
a = 'geeksforgeeks'  
  
while i < len(a):  
    if a[i] == 'e' or a[i] == 's':  
        i += 1  
        break  
  
    print('Current Letter :', a[i])  
    i += 1
```

Output

Current Letter : g