

Conditionals and Loops



If Statement

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

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

If statements always begin
with an “if”

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

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

True

If statements always begin
with an “if”

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 Statement

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

```
x=6  
if x>5:  
    print(x)
```

If statements always begin with an "if"

Conditional: must return boolean

":" ends statement

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

If Statement

```
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.

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 + Else Statement

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

Below 5

If condition is False, so code inside of else executes

If + Elif + Else Statement

How do I specify multiple conditions?...

If + Elif + Else Statement

How do I specify multiple conditions?...elif

If + Elif + Else Statement

How do I specify multiple conditions?...elif

```
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 + Else Statement

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

```
x == 'roger'
```

False

If + Elif + Else Statement

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

x == 'bugs'

False

If + Elif + Else Statement

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

x == 'bugs'

False

Since all if/elif conditions were False, else statement executes

If + Elif + Else Statement

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

Run!!!!

If + Elif + Else 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

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")
```

Nothing is printed since neither if nor elif condition is true

If + Elif + Else Statement

Example with multiple elif conditions:

```
x = 'killer rabbit'
season = "winter"
if x == 'roger':
    print("hello jessica")
elif x == 'bugs':
    print("What's up doc")
elif season == "winter":
    print("no bunnies")
else:
    print("Run!!!!")
```

If + Elif + Else Statement

Example with multiple elif conditions:

```
x = 'killer rabbit'
season = "winter"
if x == 'roger':
    print("hello jessica")
elif x == 'bugs':
    print("What's up doc")
elif season == "winter":
    print("no bunnies")
else:
    print("Run!!!!")
```



If + Elif + Else Statement

Example with multiple elif conditions:

```
x = 'killer rabbit'
season = "winter"
if x == 'roger':
    print("hello jessica")
elif x == 'bugs':
    print("What's up doc")
elif season == "winter":
    print("no bunnies")
else:
    print("Run!!!!")
```



If + Elif + Else Statement

Example with multiple elif conditions:

```
x = 'killer rabbit'
season = "winter"
if x == 'roger':
    print("hello jessica")
elif x == 'bugs':
    print("What's up doc")
elif season == "winter":
    print("no bunnies")
else:
    print("Run!!!!")
```



If + Elif + Else Statement

Example with multiple elif conditions:

```
x = 'killer rabbit'
season = "winter"
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

If + Elif + Else Statement

Example with multiple elif conditions:

```
x = 'killer rabbit'
season = "winter"
❶ if x == 'roger':
    print("hello jessica")
❷ elif x == 'bugs':
    print("What's up doc")
❸ elif season == "winter":
    print("no bunnies")
❹ else:
    print("Run!!!!")
```



no bunnies

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")
```

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")
```

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")
```



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")
```



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")
```



food menu

Specifying Multiple Conditions

How do you specify multiple conditions in a conditional statements?

Specifying Multiple Conditions - and

How do you specify multiple conditions in a conditional statements?

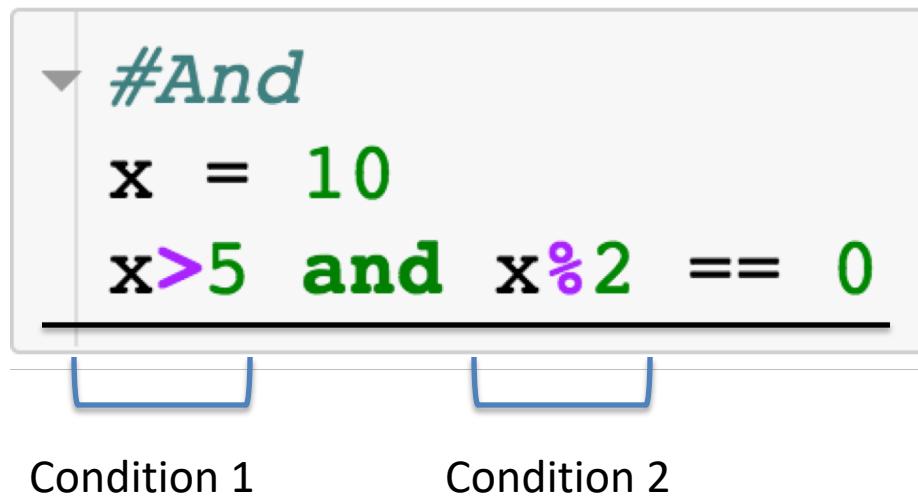
```
#And  
x = 10  
x>5 and x%2 == 0
```

Condition 1 Condition 2

Conditional statements
w/ “and”

Specifying Multiple Conditions - and

How do you specify multiple conditions in a conditional statements?



AND

- All conditions must evaluate to True
- Can have arbitrary number of conditions

Specifying Multiple Conditions - and

How do you specify multiple conditions in a conditional statements?

▼ *#And*

```
x = 10  
x>5 and x%2 == 0
```

True

▼ *#Another example*

```
name = 'Jake'  
name[0]=='J' and len(name)>5
```

False

Specifying Multiple Conditions - or

How do you specify multiple conditions in a conditional statements?

```
#or  
L = [1,2,3,4]  
L[0]=='a' or sum(L)!=5
```

Condition 1

Condition 2

Conditional statements
w/ "or"

Specifying Multiple Conditions - or

How do you specify multiple conditions in a conditional statements?

```
#or
L = [1,2,3,4]
L[0]=='a' or sum(L)!=5
```

The code block is enclosed in a light gray box. An arrow points from the word '#or' to the word 'or'. Below the box, two blue brackets extend downwards from the line 'L[0]=='a'' and the line 'sum(L)!=5', grouping them together. The bracket under 'L[0]=='a'' is labeled 'Condition 1' and the bracket under 'sum(L)!=5' is labeled 'Condition 2'.

Condition 1 Condition 2

OR

- At least one condition must evaluate to True
- Can have arbitrary number of conditions

Specifying Multiple Conditions

How do you specify multiple conditions in a conditional statements?

```
▼ #or
L = [1,2,3,4]
L[0]=='a' or sum(L)!=5
```

```
True
```

Specifying Multiple Conditions

How do you specify multiple conditions in a conditional statements?

```
▼ #or
L = [1,2,3,4]
L[0]=='a' or sum(L)!=5
```

```
True
```

```
▼ #And + or
(name=="Joe" or 'a' in name) and 5 not in L
```

```
True
```

Nested Conditionals + and/or

```
#Nested Conditionals
bat_avg = 0.312
hr = 39
rbi = 103
if bat_avg> 0.33:
    print("all-star")
else:
    if hr>40 or rbi>150:
        print("all-star")
    elif bat_avg>0.3 and hr>30 and rbi>100:
        print("all-star")
    else:
        print("nope")
```

Nested Conditionals

```
#Nested Conditionals
bat_avg = 0.312
hr = 39
rbi = 103
if bat_avg> 0.33:
    print("all-star")
else:
    if hr>40 or rbi>150:
        print("all-star")
    elif bat_avg>0.3 and hr>30 and rbi>100:
        print("all-star")
    else:
        print("nope")
```

all-star

For Loops

Use cases of for loops:

- Iterate over the elements of a list or string
 - For each element, perform some sort of operation - count or sum
- 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

- 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
```

Keyword "for"

Loop variable –
takes on each
value of loop
target

Loop target –
object to be
iterated over

For Loops w/ List

Can call loop variable whatever you want

```
num = 2
```

```
total = 0
```

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

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

Block of code to run each iteration of for loop

For Loops w/ List

```
num = 2
```

```
total = 2
```

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

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

For Loops w/ List

num = 4

total = 2

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

→ **for** num **in** list_nums:
 total+=num

For Loops w/ List

num = 4

total = 6

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

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

For Loops w/ List

num = 6

total = 6

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

→ **for** num **in** list_nums:
 total+=num

For Loops w/ List

num = 6

total = 12

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

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

For Loops w/ List

num = 8

total = 12

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

→ **for** num **in** list_nums:
 total+=num

For Loops w/ List

num = 8

total = 20

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

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

For Loops w/ Strings

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

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

total_words = count+1
```

For Loops w/ Strings

c = "C"

count = 0



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

For Loops w/ Strings

c = "C"

count = 0

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

total_words = count+1
```

For Loops w/ Strings

c = "h"

count = 0



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

total_words = count+1
```

For Loops w/ Strings

c = "h"

count = 0

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

total_words = count+1
```

For Loops w/ Strings

c = "e"

count = 0

...



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

total_words = count+1
```



For Loops w/ Strings

c = "e"

count = 0



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

total_words = count+1
```

For Loops w/ Strings

```
c = ""
```

```
count = 0
```



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



For Loops w/ Strings

c = ""

count = 0



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

```
total_words = count+1
```

For Loops w/ Strings

c = ""

count = 1



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

total_words = count+1
```

For Loops w/ Strings

c = "l"

count = 1



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

total_words = count+1
```



And so on...

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
- We will see other python objects that can be loop for targets

Range

Built-in range
function

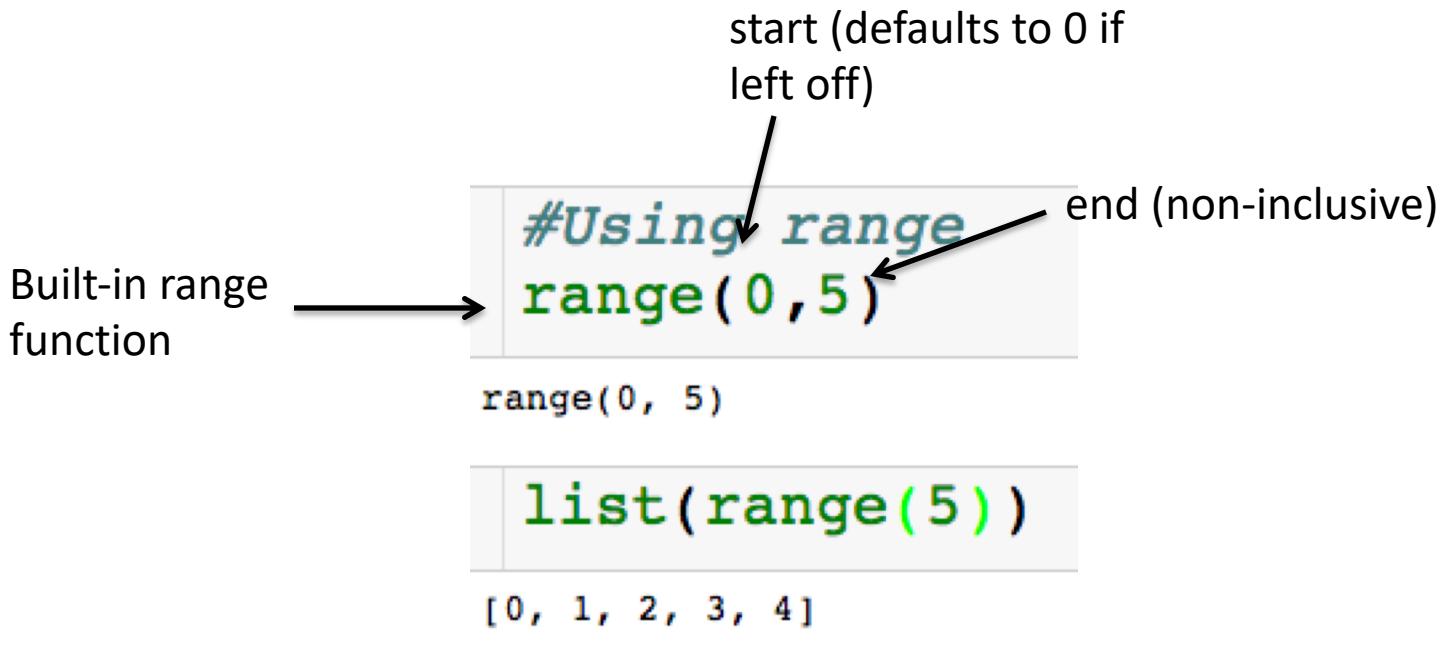
```
#Using range
range(0,5)
range(0, 5)

list(range(5))
[0, 1, 2, 3, 4]
```

start (defaults to 0 if left off)

end (non-inclusive)

Range



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

For Loops w/ Range

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

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

For Loops w/ Range

even_nums = []

i = 0



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

For Loops w/ Range

even_nums = []

i = 0

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

For Loops w/ Range

even_nums = [0]

i = 0

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

For Loops w/ Range

even_nums = [0]

i = 1

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

For Loops w/ Range

even_nums = [0]

i = 1

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

For Loops w/ Range

even_nums = [0]

i = 2

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

For Loops w/ Range

even_nums = [0]

i = 2

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

For Loops w/ Range

even_nums = [0, 2]

i = 2

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

For Loops w/ Range

even_nums = [0, 2]

i = 3

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

And so on...

For Loops w/ Range

- Using range() as loop target to do something n times.
 - Loop variables will not be used
- Get n random samples of $U[0,1]$

For Loops w/ Range

- Using range() as loop target to do something n times.
 - Loop variables will not be used
- Get n random samples of U[0,1]

Loop will run n times

```
n = 10
random_nums = []
for i in range(n):
    x = np.random.uniform()
    random_nums+=[x]
```

For Loops w/ Range

- Using range() as loop target to do something n times.
 - Loop variables will not be used
- Get n random samples of U[0,1]

Loop will run n times

```
n = 10
random_nums = []
for i in range(n):
    x = np.random.uniform()
    random_nums+= [x]
```

Generates U[0,1]

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 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]
```

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]
```

Why would I ever need to do this?