CMPUT 174

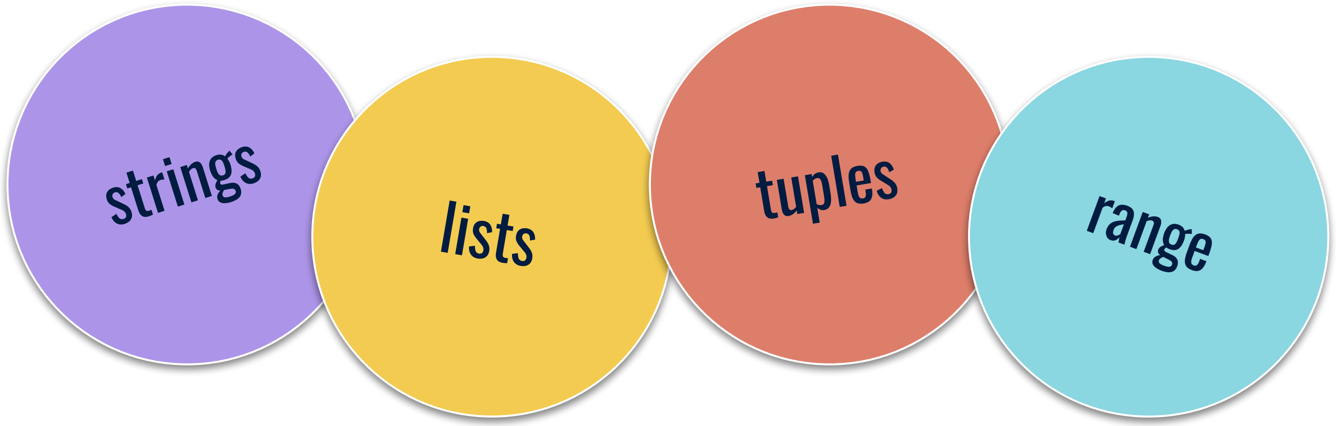
Sequences | For Loops

Lecture Outline

- ❑ Sequences
- ❑ Strings
- ❑ Lists
- ❑ Tuples
- ❑ Range
- ❑ *for* Statements

What are Sequences?

- Sequences are a generic term for an ordered collection
- You can refer to any item in a sequence using its position within the sequence
- The position or index of the item in the sequence *always* starts from `0`. There are 4 sequences that we will discuss in this course



Strings

- The `str` type represents a sequence of characters that are immutable – no operations can change the value of any string object
- The `str` type (or `str class`) is complex, and provides methods that can be applied to objects of type `str`

Strings

- Examples of expressions that evaluate to objects of type `str`

`"world"`	Strings can also be denoted by double quotations
`"What's up"`	To use apostrophes in the string, use double quotes
`"4.2"`	Evaluates to an object of type `str`
`"100% chance"`	Various symbols can also be included in a string
`'2+2'`	Evaluating this *doesn't* give an `int` object of 4, but rather an object of `str` type which represents the sequence of characters 2+2

Strings

- Strings have many methods and operations!

String Subscription

```
>>> 'Hello'[0]
    'H'
```

Evaluates to the str object that represents the character *H*

String Concatenation

```
>>> "Hello" + "World"
    "HelloWorld"
```

Evaluates to the str object that represents the sequence of characters

String Multiplication

```
>>> "Cat" * 3
    "CatCatCat"
```

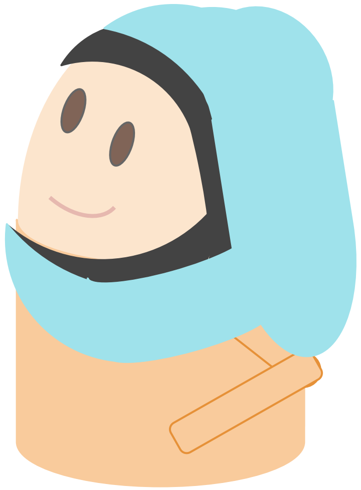
Evaluates to the str object that represents the sequence of characters

Lists

- The list type in Python represents a sequence of objects that may be of different types
- Lists don't hold the actual objects themselves, but rather the references to the objects
- The entries in a list are ordered and typically indexed by non-negative integers, starting at `0`
- Lists are mutable – references to the objects in the list can be changed

List Displays

```
[ expr, expr, expr ]
   0     1     2
```



Lists

- Examples of expressions that create list objects

A list can contain str objects

```
["apple", "pear", "strawberry", "peach"]
```

It can contain int or float objects

```
[45, 1, 74.0, 1]
```

Contain objects of different types, including lists

```
[["hello", "world"], 3, 5.0, "!!!"]
```

Lists

- Lists have many methods and operations!

List Subscription

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

Evaluates to the first item in the list

List Concatenation

```
>>> ["a", "b"] + ["c"]
    ["a", "b", "c"]
```

Evaluates to a new list object that holds references to str objects

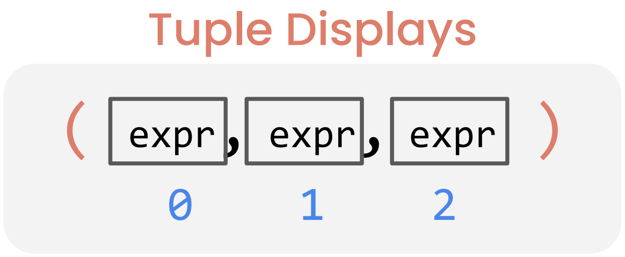
List Multiplication

```
>>> ['cat'] * 3
    ['cat', 'cat', 'cat']
```

Evaluates to a new list object that holds 3 references to the same str object

Tuples

- Tuples in Python are used to represent a sequence of objects
- Objects in a tuple are ordered, and indexed by non-negative integers, starting at `0`
- They're immutable – references inside a tuple object cannot be changed



Tuples

- Examples of expressions that create tuple objects

A tuple can contain `str` objects

```
("apple", "pear", "strawberry", "peach")
```

It can contain `int` or `float` objects

```
(45, 1, 74.0, 1)
```

Contain objects of different types, including lists & tuples

```
(["hello", "world"], (1, 2) 3, 5.0, "!!!")
```

Tuples

- Tuples have many methods and operations!

Tuple Subscription

```
>>> ('a', 'b', 'c')[1]
    b
```

Evaluates to the second item in the tuple

Tuple Concatenation

```
>>> (1, 2) + (3, 4)
    (1, 2, 3, 4)
```

Evaluates to a new tuple object that holds references to int objects

Tuple Multiplication

```
>>> ('cat') * 3
    ('cat', 'cat', 'cat')
```

Evaluates to a new tuple object that holds 3 references to the same str object

Range

- The range is a *built-in* function that generates a range object, which is an immutable sequence of numbers
- There are *two ways* of creating a range object:

```
range(stop)

range(start, stop[, step])
```

Common method – returns a sequence of numbers from `0` to `stop-1`

Range

- The most used way of creating a sequence with the range function, is when you want to:
    - Begin the sequence at `0`
    - Increment by 1
    - Stop before reaching the specified stop value

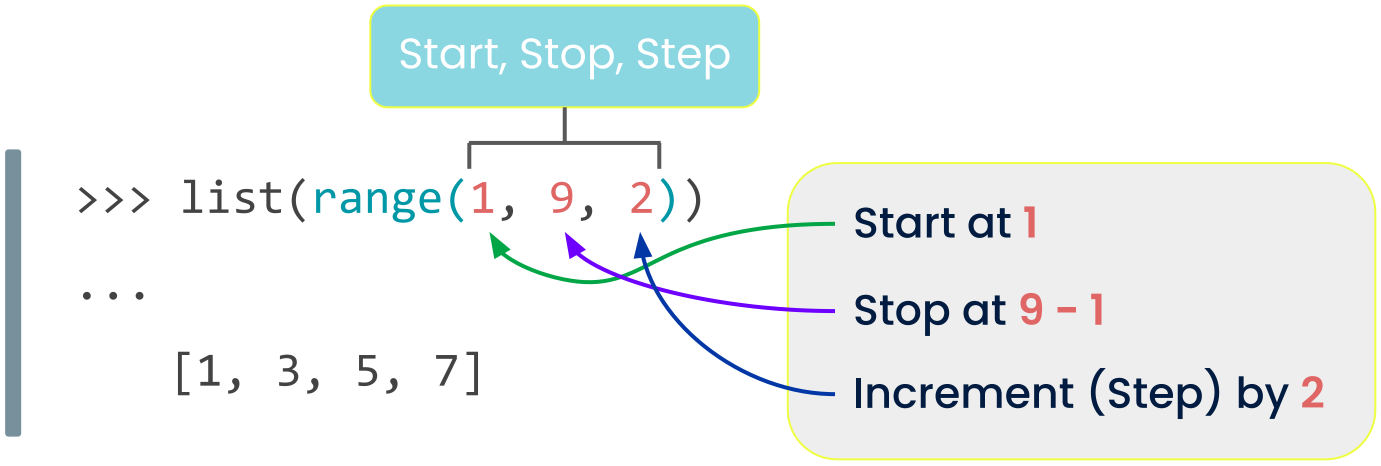
Start at `0` and go up to 4

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

Converting the range object to a list object!

Range

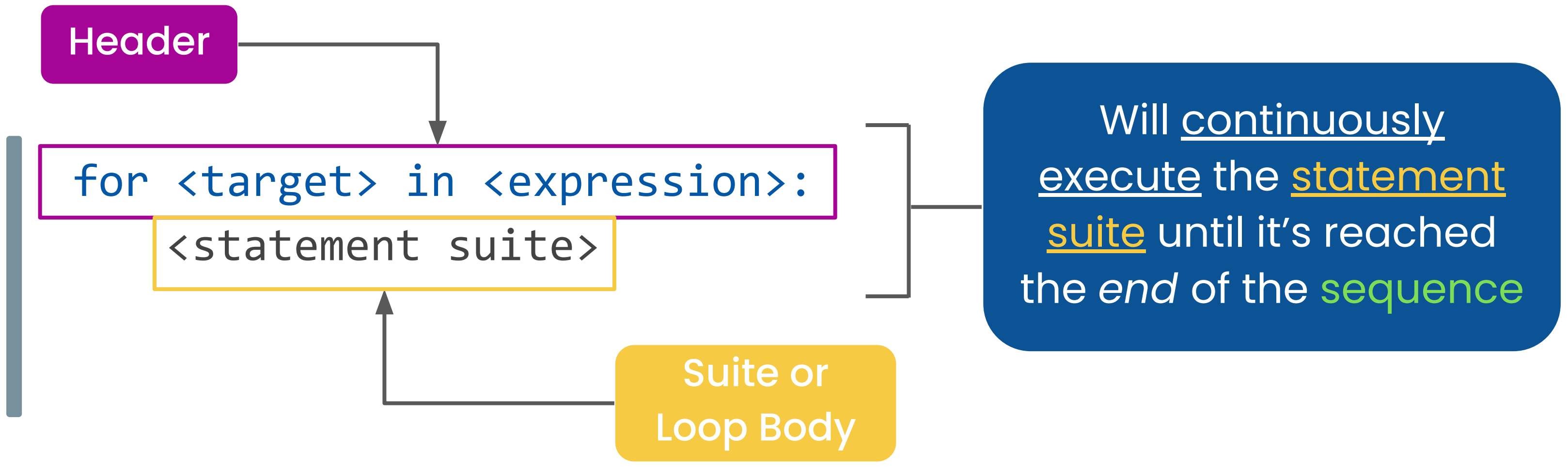
- Another way of creating a sequence with range, is when you want to:
  - Begin the sequence at a number other than `0`
  - Stop at one before the specified integer argument
  - Specify an increment other than 1



```
>>> list(range(1, 9, 2))
...
    [1, 3, 5, 7]
```

for Statements

- `for` **statements** are another type of Compound Statement
- It is a type of repetition statement (also called a "*loop*") that allows us to repeatedly evaluate a group of statements


Header
Suite or Loop Body
Will continuously execute the statement suite until it's reached the *end* of the sequence

```
for <target> in <expression>:
    <statement suite>
```

for Statements

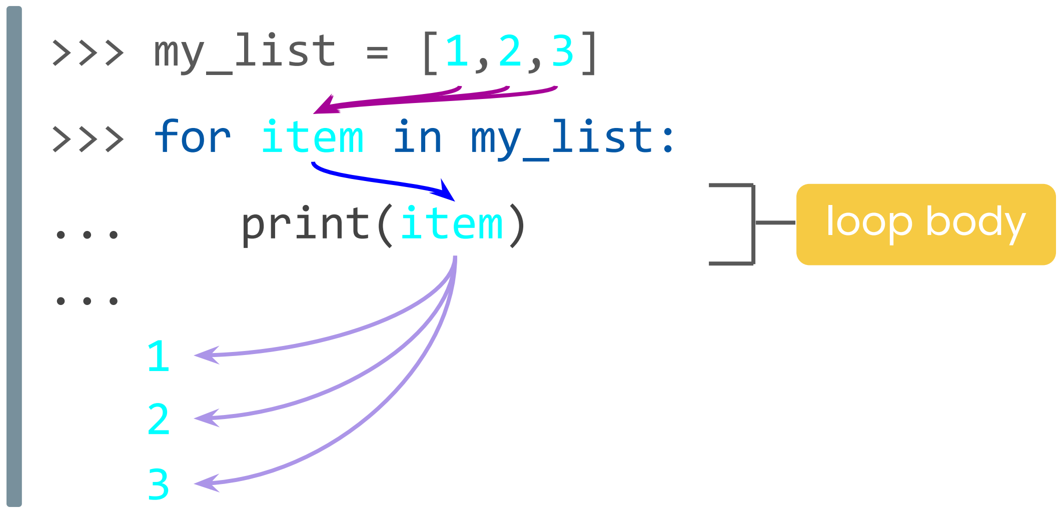
- For instance, we might want to perform operations on every element in a list
- We can also repeat code a specific number of times

★ `for` loops are great for these cases because the programmer knows how many times the loop needs to repeat!

Hence, performing definite iterations – the number of repetitions is specified *explicitly* in advance

for Statements

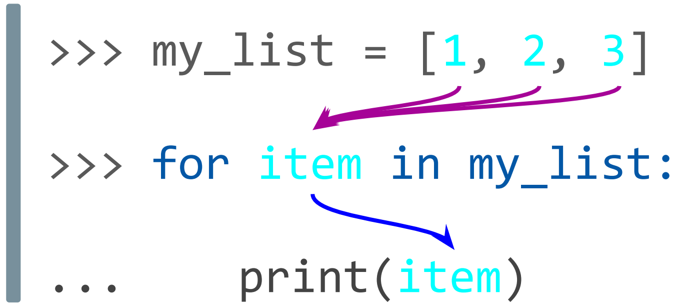
- `for` **statements** are used to iterate over sequences, such as **strings**, lists, tuples, or range *in order*
- The statement suite, or "*loop body*", will operate on each element of the sequence



```
>>> my_list = [1,2,3]
>>> for item in my_list:
...     print(item)
...
    1
    2
    3
```

for Statements

- The `for` **statement** works by binding the target (identifier) to each element of the sequence in order, and then evaluating the statement suite for that binding



```
>>> my_list = [1, 2, 3]

>>> for item in my_list:

...     print(item)
```

- The expression in a `for` **statement** needs to evaluate to a sequence, such as a string, list, tuples or a range

for Loops & Strings

- In the case of strings, the suite of the `for` statement will be evaluated for <u>each character of that string</u>

```
>>> word = "Hello"
>>> for letter in word:
...     print("The current letter is " + letter)
    The current letter is H
    The current letter is e
    The current letter is l
    The current letter is l
    The current letter is o
```

Evaluation of the suite has been completed – terminating the `for` loop

for Loops & Lists

- In the case of lists (similar to tuples), the suite of the `for` statement will be evaluated for each element in the list (or tuple)

```
>>> words = ['cat', 'computer', 'python']
>>> for word in words:
...     print(word + " has " + str(len(word)) + " letters")
...
    cat has 3 letters
    computer has 8 letters
    python has 6 letters
```

The suite has finished evaluating all elements of the list – terminating the `for` loop

for Loops & Range

- The call to the *built-in* range function evaluates to an immutable sequence of 5 int objects
- The suite of the for loop will be evaluated 5 times in this case

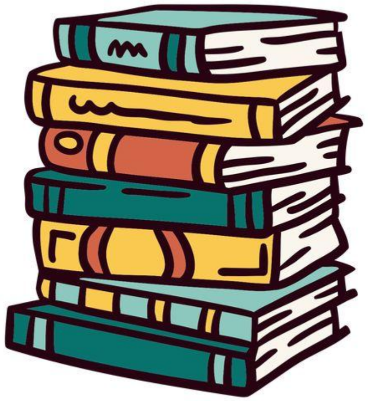
```
>>> for i in range(5):
...     print("Iteration", i)

    Iteration 0
    Iteration 1
    Iteration 2
    Iteration 3
    Iteration 4
```

Evaluates to a sequence of int objects from `0-4` **inclusive**

Finished evaluating the statement suite with `i` bound to this object – terminating the for loop

Reminder

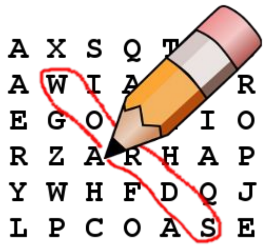
- *Online Activities:*
  - Assigned Readings:
    - Compound Statements
  - Week 3 Videos (3):
    - Lists and mutability
    - Tuples and mutability
    - For loop



Mr. Ratburn’s Classroom

Image Source: https://giphy.com/gifs/pbskids-arthur-back-to-school-pbs-kids-kbcLWrW6QPIl7SKwgc

Practice Problem 1!



for loop, list

```
'''Mr. Ratburn has created a word-search puzzle
for his students. He has asked them to identify
all 3 letter words in the puzzle that start and
end with the same letter. Mr. Ratburn needs help
to prepare an answer key for the puzzle.

Create a program that searches through a list of
words and prints all 3 letter words that start and
end with the same letter.
'''
```

Image Source: http://clipart-library.com/

Practice Problem 2!

for loop, range, list, list methods

```
'''Mr. Ratburn has given his students a list of
words.

He has asked them to scan the list in order and move
any word that starts with a vowel to the start of the
list.

At the end of this task, all words that start with a
vowel would be listed before words that do not
start with a vowel.

Mr. Ratburn needs help with the answer key. Create a
program that scans a list of words, does the
required task and prints the modified list of words
'''
```

Practice Problem 3!

for loop, range

```
'''Mr. Ratburn has come up with the following
challenging task for his students:

Write all odd numbers between 1 to 100 that are
divisible by 5 or 7.

Find and write the average of all even numbers
between 1 to 100 that are divisible by 5 or 7.

His plan is to nominate students who complete the
task correctly for the Annual Math Whiz Award.

Help Mr. Ratburn prepare an answer key by creating
a program that does the required task.
'''
```