Strings and Lists

CSCI 1030U - Intro to Computer Science
@IntroCS

Randy J. Fortier
@randy_fortier

OntarioTech UNIVERSITY

Outline

- Strings
- Lists

OntarioTech UNIVERSITY

Strings

OntarioTech UNIVERSITY

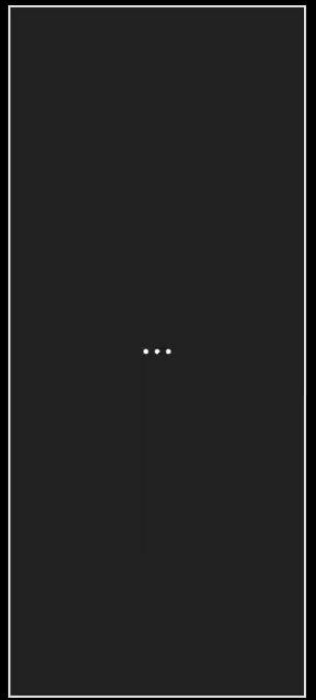
Strings

- Strings are sequences of characters
  - Each character is represented internally by 1 or more bytes, encoded as UTF-8
  - Byte strings allow you to see the internal byte state

```
name = "Carla Rodriguez"
name_bytes = b"Carla Rodriguez"
```

Strings

```
pokemon_name = 'Tepig'
pokemon_type = b'Fire'
base_damage = 20
damage_multiplier = 2.0
damage_msg = f'{base_damage*damage_multiplier} (x{damage_multiplier:.2f})'
dialogue = '''I like shorts!
They are comfy and easy to wear!'''
```



Strings - Indexing

- Characters of any string can be accessed using the `[]` operator
  - Inside the square brackets, the index of the element is provided
  - Indices begin at `0`
    - i.e. `str[0]` is the first character in `str`, `str[1]` is the second
  - You can also use negative numbers, which count from the end of the string

```
name = "Carla Rodriguez"
print(name[0])
print(name[-1])
```

OntarioTech UNIVERSITY

Strings - Slices

- The slice operator, `[:]`, can be used to select a subset of the characters of a string
  - e.g. `[:4]` – Selects the first 4 characters (indexes 0, 1, 2, and 3)
  - e.g. `[3:]` – Selects all but the first 3 characters (indexes 4, 5, 6, …)
  - e.g. `[2:5]` – Selects characters with indexes 2, 3, and 4

```
name = "Carla Rodriguez"
print(name[3:6])
print(name[:4])
print(name[5:])
```

OntarioTech UNIVERSITY

Strings - Stepped Slices

- The slice operator, `[:]`, can also be used with a step
  - e.g. `str[2:8:2]` – Selects characters with indexes 2, 4, and 6

```
name = "Carla Rodriguez"
print(name[2:8:2])
print(name[1:10:3])
```

OntarioTech UNIVERSITY

Strings - Functions

- `len(name)` – Returns the number of characters in `name`

```
name = "Carla Rodriguez"
print(len(name))
```

OntarioTech UNIVERSITY

Strings - Concatenation

- To concatenate two strings:
  - `str1 + str2`

```
name1 = "Carla"
name2 = "Rodriguez"
print(name1 + name2)
```

Strings - Repetition

- To repeat a string 5 times:
  - `str * 5`
  - `5 * str`

```
print('-'*10)
print(10*'-')

num = 8
print('ABC' * n)
```

For Loops Revisited

- A common use for *for* loops is iterating on a string:

```
name = "Carla Rodriguez"
for letter in name:
    print(letter)
```

For Loops Revisited

- The `range()` function is also useful for iterating over a sequence of indices:

```
name = "Carla Rodriguez"
for i in range(len(name)):
    print(name[i])
```

OntarioTech UNIVERSITY

Coding Exercise 03b.1

- Write some code that takes a full name (format: First Last), and separates the two names into their own variables

OntarioTech UNIVERSITY

Lists

OntarioTech UNIVERSITY

Lists - Indexing

- Elements of any list can be accessed using the `[]` operator
  - Inside the square brackets, the index of the element is provided
  - Indices begin at 0
    - i.e. `marks[0]` is the first element in `marks`, `marks[1]` is the second
  - You can also use negative numbers, which count from the end of the list

```
nums = [0,1,2,3,4,5,6,7,8,9]
print(nums[3])
print(nums[-1])
```

OntarioTech UNIVERSITY

Lists - Insertion

- Elements can be added to a list
  - Insert at the end of a sequence using the `append()` function
  - Insert at an arbitrary position using the `insert()` function

```
nums = [0,2,4,6,8]
nums.insert(4, 7)
print(nums)
nums.append(10)
print(nums)
```

Lists - Deletion

- Elements can be removed from a list
  - Remove any element from a list using the `remove()` function

```
nums = [0,2,4,6,8]
nums.remove(4)
```

OntarioTech UNIVERSITY

Lists - Deletion

- Elements can be removed from a list
  - Remove the last element from a list using the `pop()` function (without any arguments)

```
nums = [0,2,4,6,8]
nums.pop()
print(nums)
```

OntarioTech UNIVERSITY

Lists - Deletion

- Elements can be removed from a list
  - Remove an element from any position in the list using the `pop()` function with an index argument

```
nums = [0,2,4,6,8]
nums.pop(1)
print(nums)
```

OntarioTech UNIVERSITY

Lists - Slices

- The slice operator, `[:]`, can be used to select a subset of the elements of a list
  - e.g. `[:4]` – Selects the first 4 elements (indexes 0, 1, 2, and 3)
  - e.g. `[3:]` – Selects all but the first 3 elements (indexes 4, 5, 6, …)
  - e.g. `[2:5]` – Selects elements with indexes 2, 3, and 4

```
nums = [1,2,3,4,5,6,7,8,9,10]
print(nums[3:6])
print(nums[:4])
print(nums[5:])
```

OntarioTech UNIVERSITY

Lists - Stepped Slices

- The slice operator, `[:]`, can also be used with a step
  - e.g. `[2:8:2]` – Selects elements with indexes 2, 4, and 6

```
nums = [1,2,3,4,5,6,7,8,9,10]
print(nums[2:8:2])
print(nums[1:10:3])
```

OntarioTech UNIVERSITY

Lists - Functions

- Some useful sequence functions:
  - `len(name)` – Returns the number of elements in `name`
  - `max(marks)` – Returns the largest number in `marks`
  - `min(marks)` – Returns the smallest number in `marks`
  - `sum(marks)` – Returns the sum of the numbers in `marks`

```
nums = [1,2,3,4,5,6,7,8,9,10]
print(len(nums))
print(max(nums))
print(min(nums))
print(sum(nums))
```

OntarioTech UNIVERSITY

Lists - Functions

- Some useful sequence functions:
  - `range(10)` – Returns a list, including numbers `0-9`
  - `range(2,10)` – Returns a list, including numbers `2-9`
  - `range(2,10,2)` – Returns a list, including numbers `2`, `4`, `6`, and `8`

```
list = [0,1,2,3,4,5,6,7,8,9]
for x in list:
    print(x)

for x in range(0,10):
    print(x)
```

OntarioTech UNIVERSITY

Lists - Membership

- To test if elements exist in a list:
  - `3 in [1,2,3,4,5]` – Returns `True`, since `3` is in `[1,2,3,4,5]`
  - `3 not in [1,2,3,4,5]` – Returns `False`, since `3` is in `[1,2,3,4,5]`

```
nums = [1,2,3,4,5,6,7,8,9,10]
if 3 in nums:
    print('Found three')
```

OntarioTech UNIVERSITY

Lists - Membership

- To test if elements exist in a list:
  - `[1,2,3,4,5].index(3)` – Returns `2`, the index of the `3` in `[1,2,3,4,5]`
  - `[1,2,3,4,5].index(8)` – Generates an error, since `8` is not in `[1,2,3,4,5]`
  - `[1,2,3,1,2,3,1,2,3].count(3)` – Returns `3`, the count of `3`s in the list

```
nums = [1,2,3,4,5,6,7,8,9,10]
print(nums.index(3))
#print(nums.index(20))
print(nums.count(3))
```

OntarioTech UNIVERSITY

Lists - Concatenation

- To concatenate two lists:
  - `seq1 + seq2`

```
nums1 = [1,2,3,4,5]
nums2 = [6,7,8,9,10]
print(nums1 + nums2)
```

OntarioTech UNIVERSITY

Lists - Repetition

- To repeat a list 5 times:
  - `myList * 5`
  - `5 * myList`

```
print([1,2,3]*10)
print(10*[1,2,3])

n = 8
print(['Bob', 'Smith'] * n)
```

For Loops Revisited

- A common use for *for* loops is iterating on a list:

```
nums = [1,2,3,4,5]
for element in nums:
    print(element)
```

OntarioTech UNIVERSITY

For Loops Revisited

- The `range()` function is also useful for iterating over a list of numbers:

```
for x in range(0,10):
    print(x)

list = [1,2,3,4,5,6,7]
for i in range(len(list)):
    print(list[i])
```

OntarioTech UNIVERSITY

Lists of Lists (Matrices)

- List elements can be of any type:
  - Numbers
  - Characters
  - Boolean values
  - Strings
  - Tuples (discussed later)
  - Even other lists

OntarioTech UNIVERSITY

Lists of Lists (Matrices)

- Lists of lists (aka matrices, 2D lists):

```
list1 = [2,3,4,5]
list2 = [2,4,6,8]
list3 = [5,4,3,2]
matrix1 = [list1, list2, list3]
matrix2 = [[1,2,3],[4,5,6],[7,8,9]]
print(matrix2[1][2])   # prints 6
```

Coding Exercise 03b.2

- Write some code that takes a list of floating point numbers, and prints the average of all of the numbers in the list

OntarioTech UNIVERSITY

Coding Challenge 03b.1

- Write some code that takes a list of integers, and prints the average of all of the *even* numbers in the list

OntarioTech UNIVERSITY

Hacker's Corner: List Comprehension

- A Python shorthand for creating lists using a mathematical description (similar to set comprehension)

```
squares = [x**2 for x in range(10)]
even_squares = [x**2 for x in range(10) if x % 2 == 0]
even_odd = ['Even' if i % 2 == 0 else 'Odd' for i in range(10)]
pairs = [[(i,j) for j in range(10)] for i in range(10)]
```

OntarioTech UNIVERSITY

Wrap-up

- Strings
- Lists

OntarioTech UNIVERSITY

Coming Up

- Tuples
- Dictionaries

OntarioTech UNIVERSITY