Python: 6th lesson – Strings & Dictionaries

Strings:

* One place where the Python language really shines is in the manipulation of strings. This section will cover some of Python's built-in string methods and formatting operations. Such string manipulation patterns come up often in the context of data science work.
* String syntax:

You've already seen plenty of strings in examples during the previous lessons, but just to recap, strings in Python can be defined using either single or double quotations. They are functionally equivalent.

x = 'Pluto is a planet'

y = "Pluto is a planet"

x == y

True

Double quotes are convenient if your string contains a single quote character (e.g. representing an apostrophe). Similarly, it's easy to create a string that contains double-quotes if you wrap it in single quotes:

print("Pluto's a planet!")

print('My dog is named "Pluto"')

Pluto's a planet!

My dog is named "Pluto"

If we try to put a single quote character inside a single-quoted string, Python gets confused:

'Pluto's a planet!'

File "/tmp/ipykernel\_19/1561186517.py", line 1

'Pluto's a planet!'

^

SyntaxError: invalid syntax

We can fix this by "escaping" the single quote with a backslash.

'Pluto\'s a planet!'

"Pluto's a planet!"

* Strings as sequences:

Strings can be thought of as sequences of characters. Almost everything we've seen that we can do to a list, we can also do to a string.

# Indexing

planet = 'Pluto'

planet[0]

'P'

# Slicing

planet[-3:]

'uto'

# How long is this string?

len(planet)

5

# Yes, we can even loop over them

[char+'! ' for char in planet]

['P! ', 'l! ', 'u! ', 't! ', 'o! ']

* Strings methods:

Like list, the type str has lots of very useful methods. We'll show just a few examples here.

# ALL CAPS

claim = "Pluto is a planet!"

claim.upper()

'PLUTO IS A PLANET!'

# all lowercase

claim.lower()

'pluto is a planet!'

# Searching for the first index of a substring

claim.index('plan')

11

claim.startswith(planet)

True

# false because of missing exclamation mark

claim.endswith('planet')

False

Dictionaries:

* Dictionaries are a built-in Python data structure for mapping keys to values.

numbers = {'one':1, 'two':2, 'three':3}

In this case 'one', 'two', and 'three' are the keys, and 1, 2 and 3 are their corresponding values. Values are accessed via square bracket syntax similar to indexing into lists and strings.

numbers['one']

1

We can use the same syntax to add another key, value pair.

numbers['eleven'] = 11

numbers

{'one': 1, 'two': 2, 'three': 3, 'eleven': 11}

Or to change the value associated with an existing key.

numbers['one'] = 'Pluto'

numbers

{'one': 'Pluto', 'two': 2, 'three': 3, 'eleven': 11}

* Python has dictionary comprehensions with a syntax similar to the list comprehensions we saw in the previous tutorial.

planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']

planet\_to\_initial = {planet: planet[0] for planet in planets}

planet\_to\_initial

{'Mercury': 'M',

'Venus': 'V',

'Earth': 'E',

'Mars': 'M',

'Jupiter': 'J',

'Saturn': 'S',

'Uranus': 'U',

'Neptune': 'N'}

* The in operator tells us whether something is a key in the dictionary.

'Saturn' in planet\_to\_initial

True

'Betelgeuse' in planet\_to\_initial

False

* A for loop over a dictionary will loop over its keys.

for k in numbers:

print("{} = {}".format(k, numbers[k]))

one = Pluto

two = 2

three = 3

eleven = 11

* We can access a collection of all the keys or all the values with dict.keys() and dict.values(), respectively.

# Get all the initials, sort them alphabetically, and put them in a space-separated string.

' '.join(sorted(planet\_to\_initial.values()))

'E J M M N S U V'

* The very useful dict.items() method lets us iterate over the keys and values of a dictionary simultaneously (in Python jargon, an item refers to a key, value pair).

for planet, initial in planet\_to\_initial.items():

print("{} begins with \"{}\"".format(planet.rjust(10), initial))

Mercury begins with "M"

Venus begins with "V"

Earth begins with "E"

Mars begins with "M"

Jupiter begins with "J"

Saturn begins with "S"

Uranus begins with "U"

Neptune begins with "N"