Business Programming (using Python)

Zhaohu (Jonathan) Fan

Information Technology Management

Scheller College of Business

Georgia Institute of Technology

September 19, 2023

Main topics

- Data Structures
  - Strings & Regular Expressions
  - Exercises

Data Structures - Strings & Regular Expressions

Business context question

- **In a text processing application for your e-commerce business, you are required to validate product codes.**
- **The product codes you are interested in must have exactly five characters, starting with an `a` and ending with an `s` .**
- **You decide to use Python's re module and come across the regular expression pattern ^a...s$.**
- **Question**:
  - Using this regular expression pattern `^a...s$` , can you determine whether the given product codes `"abs"` or `alias` or `"abyss"` are valid based on the criteria?

Example #1

- **Question**:
  - Using this regular expression pattern `^a ... s$` , can you determine whether the given product codes `"abs"` or `alias` or `"abyss"` are valid based on the criteria?
- **Quick answers**:

Expression	String	Matched?
	`abs`	No match
	`alias`	Match
`^a ... s$`	**`abyss`**	Match
	`Alias`	No match
	`An abacus`	No match

Example #1

- A **Reg**ular **Ex**pression (RegEx) is a sequence of characters that **defines a search pattern**.
  - A pattern defined using **RegEx** can be used to match against a string.

Expression	String	Matched?
	`abs`	No match
	`alias`	Match
`^a ... s$`	`abyss`	Match
	`Alias`	No match
	`An abacus`	No match

Regular Expressions

- For example, the below code defines a RegEx pattern.
    - The pattern is: **any five letter string starting with `a` and ending with `s`**.

Code

```
^a ... s$
```

Example #1

- Python has a module named `re` to work with RegEx.

Code

```
import re
pattern = '^a ... s$'
test_string = 'abyss'
result = re.match(pattern, test_string)
if result:
    print("Search successful.")
else:
    print("Search unsuccessful!")
```

Example #1

- Python has a module named `re` to work with RegEx.

Code

```
import re
pattern = '^a ... s$'
test_string = 'abyss'
result = re.match(pattern, test_string)
if result:
    print("Search successful.")
else:
    print("Search unsuccessful!")
```

Example #1

- The caret symbol `^` is used to check if a string **starts with** a certain character.
  - The dollar symbol `$` is used to check if a string **ends with** a certain character.
  - The period symbol `.` matches **any single character**.
  - **Together,** `^a...s` **matches any five-letter string starting with** `a` **and ending with** `s`.
  - `^`, `$`, and `.` are metacharacters.

Code

```
import re
  pattern = '^a...s$'
test_string = 'abyss'
result = re.match(pattern, test_string)
if result:
    print("Search successful.")
else:
    print("Search unsuccessful!")
```

MetaCharacters

- Metacharacters are characters that are interpreted in a special way by a RegEx engine.
    - Here's a list of metacharacters:
    - `[] . ^ $ * + ? {} () \ |`

Example #1

- `test_string` is a variable that holds the string "abyss" that you want to match against the pattern.

Code

```
import re
pattern = '^a...s$'
  test_string = 'abyss'
result = re.match(pattern, test_string)
if result:
    print("Search successful.")
else:
    print("Search unsuccessful!")
```

Example #1

- The `re.match()` function tries to match the pattern against the beginning of the test_string.
    - If it matches, `re.match()` returns a match object; otherwise, it returns `None`.

Code

```
import re
pattern = '^a ... s$'
 test_string = 'abyss'
  result = re.match(pattern, test_string)
if result:
    print("Search successful.")
else:
    print("Search unsuccessful!")
```

Example #1

- if result is not `None` (i.e., it found a match), it prints `"Search successful.`
    - Otherwise, it prints `"Search unsuccessful!"`

Code

```
import re
pattern = '^a ... s$'
 test_string = 'abyss'
 result = re.match(pattern, test_string)
if result:
    print("Search successful.")
  else:
    print("Search unsuccessful!")
```

Example #1

Output

```
Search successful.
```

Question(s)?

- Q: How do I use the .now() function to show the current time if I am in the EST time zone?

Example #2

- **How do I use the .now() function to show the current time if I am in the EST time zone?**
  - Import required modules
  - Initialize the Time Zone: Use `pytz.timezone('US/Eastern')` for EST
  - Use `datetime.datetime.now(pytz.utc)` to get the current UTC time and convert to EST
  - Use `.astimezone()` to convert the UTC time to EST, and display current time Format
  - Print the time.

Code

```
import datetime as dt
# install the pytz package if you haven't already:
# pip install pytz
import pytz
# Initialize the time zone
est = pytz.timezone('US/Eastern')
# Get the current time in UTC and then convert to EST
current_datetime_est = dt.datetime.now(pytz.utc).astimezone(est)
# Print the current date and time in EST
  print("Current date and time (EST):", current_datetime_est.strftime("%Y-%m-%d %H:%M:%S"))
```

Example #2

Output

```
Current date and time (PT): 2023-09-19 10:39:07
```

Question(s)?

- How do I use the .now() function to show the current time if I am in the Pacific Time (PT) zone?
  - For example, **what's the current time in California right now?**

Code

```
import datetime as dt
# Install the pytz package if you haven't already:
# pip install pytz
import pytz
# Initialize the time zone for Pacific Time (PT)
pt = pytz.timezone('America/Los_Angeles')
# Get the current time in UTC and then convert it to PT
current_datetime_pt = dt.datetime.now(pytz.utc).astimezone(pt)
# Print the current date and time in PT
print("Current date and time (PT):", current_datetime_pt.strftime("%Y-%m-%d %H:%M:%S"))
```

Example #2

Output

```
Current date and time (EST): 2023-09-19 13:39:09
```

What we learned

- "*New York time is 3 hours ahead of California time, but California time has not slowed down.*"
  - **New York Time**: Represents those who might seem ahead in the learning curve.
  - **California Time**: Represents those who might be feeling like they're behind.
  - **Not Slowed Down**: Even if it seems like you are behind others, it doesn't mean your progress is slower or less valuable.

Key takeaways

- "*New York time is 3 hours ahead of California time, but California time has not slowed down.*"
  - **Everyone has their own pace**
  - **Your journey is yours alone**
  - **Keep moving forward**

Exercises

- Please click on the link provided below.
  - In-Class Exercise