

# Business Programming (using Python)

Zhaohu (Jonathan) Fan

Information Technology Management

Scheller College of Business

Georgia Institute of Technology

September 26, 2023

# Main topics

- Data Structures
  - Strings & Regular Expressions (III)
  - use the `find()` to 'find' the occurrence of 'Google' within each line. (Q#3)
  - Use the `split()` to parse each line (Q#4 and Q#5)
  - Use the `split()` to parse the entered expression (Q#4 and Q#5).
  - Exercises

# Bonus question

- Current Pattern: `^\w{3,}\s(\w\s)?\w{3,}`
  - **Matches:** 'John Doe', 'John M Doe', 'John Deere'
  - **Doesn't Match:** 'John Mr Doe'
  - Useful tool for regular expression testing
- Objective:
  - **Match:** 'John Doe', 'John Deere', 'John Mr Doe'
  - **Don't Match:** 'John M Doe'
- Question:
  - How can we adjust the **regex pattern** to meet the new matching criteria using Python?
  - **Charles McCartney** will present his answer to this question on Thursday.

# Data Structures - Strings & Regular Expressions (III)

# Business context question: Analyzing tech impact through text

## Input

"As I reflect on our progress this past year,  
I first want to say thank you for your commitment and investment Google."

"We are living at Google a crucial time in history where the impact of technology  
on every part of our daily life and work and every aspect of our society and economy  
is more acute than ever before."

# Business context question: Analyzing tech impact through text

- **Objective:**

- Determine the frequency of the mention of 'Google' in our recent report to gauge our engagement level.

- **Approach:**

- Use a simple program to count the occurrences of the word 'Google' in the report.

- **Action:**

- Extract text from the report.
- Run the program to count occurrences.

# Step 1

- We first define the `report_lines` list containing the two lines of text from the report.

## Python

```
# Define the report lines
```

```
report_lines = [
```

```
    "As I reflect on our progress this past year,
```

```
    I first want to say thank you for your commitment and investment Google."
```

```
    "We are living at Google a crucial time in history where the impact of technology  
    every part of our daily life and work and every aspect of our society and  
    economy is more acute than ever before."]
```

# Step 2

- We initialize a counter `google_line_count` to 0.

## Python

```
# Initialize a counter for lines containing 'Google'  
google_line_count = 0
```

# Step 3

- We iterate through each `line` in `report_lines`, using the
  - `find()` method to check for the occurrence of the word 'Google'. If
  - `find()` returns a value other than -1, it indicates that 'Google' was found in the line, and we increment `google_line_count` by 1.

## Python

```
# Check each line for 'Google'
for line in report_lines:
    if line.find('Google') != -1:
        google_line_count += 1
```

# Step 4

- Finally, we print the result using the `print()` function.
  - `print()` Function: is used to output text to the console. f-string `(f' ... ')` :
  - The f at the beginning indicates that this is a formatted string, which allows you to embed expressions inside string literals, using curly braces `{}` .
  - Since the string is enclosed in single quotes, you need to escape the single quote around the word Google using a backslash `\` to prevent it from ending the string prematurely.
  - Embedding a Variable `{google_line_count}` : Within the f-string, `{google_line_count}` is a placeholder that gets replaced by the value of the `google_line_count` variable.
  - Result: When this line is executed, it will print a message like Number of lines containing the word 'Google': 2 (assuming `google_line_count` is 2) to the console.

## Python

```
print(f'Number of lines containing the word \'Google\': {google_line_count}')
```

# Complete Python code

## Python

```
# Define the report lines
report_lines = [
    "As I reflect on our progress this past year, I first want to say thank you for yo
    "we are living at Google a crucial time in history where the impact of technology
]

# Initialize a counter for lines containing 'Google'
google_line_count = 0

# Check each line for 'Google'
for line in report_lines:
    if line.find('Google') != -1:
        google_line_count += 1

# Print the result
print(f'Number of lines containing the word \'Google\': {google_line_count}')
```

## Output

```
Number of lines containing the word 'Google': 2
```

# Cheat sheet for metaCharacters

## Regular Expression Basics

.	Any character except newline
a	The character a
ab	The string ab
a b	a or b
a*	0 or more a's
\	Escapes a special character

## Regular Expression Quantifiers

*	0 or more
+	1 or more
?	0 or 1
{2}	Exactly 2
{2, 5}	Between 2 and 5
{2,}	2 or more

Default is greedy. Append ? for reluctant.

## Regular Expression Groups

(...)	Capturing group
(?:...)	Non-capturing group
\Y	Match the Y'th captured group

## Regular Expression Character Classes

[ab-d]	One character of: a, b, c, d
[^ab-d]	One character except: a, b, c, d
[b]	Backspace character
\d	One digit
\D	One non-digit
\s	One whitespace
\S	One non-whitespace
\w	One word character
\W	One non-word character

## Regular Expression Assertions

^	Start of string
\$	End of string
\b	Word boundary
\B	Non-word boundary
(?=...)	Positive lookahead
(?!...)	Negative lookahead

## Regular Expression Flags

g	Global Match
i	Ignore case
m	^ and \$ match start and end of line

## Regular Expression Special Characters

\n	Newline
\r	Carriage return
\t	Tab
\0	Null character
\YYY	Octal character YYY
\xYY	Hexadecimal character YY
\uYYYY	Hexadecimal character YYYY
\cY	Control character Y

## Regular Expression Replacement

\$\$	Inserts \$
\$&	Insert entire match
\$`	Insert preceding string
\$'	Insert following string
\$Y	Insert Y'th captured group

# Exercises

- Please click on the link provided below.
  - [In-Class Exercise](#)