**3. Count Occurrences of a Substring**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func countOccurrences(str, subStr string) int {

return strings.Count(str, subStr)

}

func main() {

fmt.Println(countOccurrences("hello world, hello", "hello")) // Output: 2

}

**4. Remove Whitespace from a String**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func removeWhitespace(s string) string {

return strings.ReplaceAll(s, " ", "")

}

func main() {

fmt.Println(removeWhitespace(" hello world ")) // Output: "helloworld"

}

**5. Find the Longest Word in a Sentence**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func longestWord(sentence string) string {

words := strings.Fields(sentence)

longest := ""

for \_, word := range words {

if len(word) > len(longest) {

longest = word

}

}

return longest

}

func main() {

fmt.Println(longestWord("Go is an amazing programming language")) // Output: "programming"

}

**6. Check if Two Strings are Anagrams**

go

CopyEdit

package main

import (

"sort"

"strings"

"fmt"

)

func areAnagrams(str1, str2 string) bool {

if len(str1) != len(str2) {

return false

}

s1 := strings.Split(str1, "")

s2 := strings.Split(str2, "")

sort.Strings(s1)

sort.Strings(s2)

return strings.Join(s1, "") == strings.Join(s2, "")

}

func main() {

fmt.Println(areAnagrams("listen", "silent")) // Output: true

}

**7. Convert a String to a Slice of Runes**

go

CopyEdit

package main

import "fmt"

func stringToRunes(s string) []rune {

return []rune(s)

}

func main() {

runes := stringToRunes("hello")

fmt.Println(runes) // Output: [104 101 108 108 111]

}

**8. Remove All Duplicates from a String**

go

CopyEdit

package main

import "fmt"

func removeDuplicates(s string) string {

seen := map[rune]bool{}

result := []rune{}

for \_, r := range s {

if !seen[r] {

seen[r] = true

result = append(result, r)

}

}

return string(result)

}

func main() {

fmt.Println(removeDuplicates("google")) // Output: "gole"

}

**9. Capitalize the First Letter of Each Word**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func capitalizeFirstLetter(sentence string) string {

words := strings.Fields(sentence)

for i, word := range words {

words[i] = strings.Title(word)

}

return strings.Join(words, " ")

}

func main() {

fmt.Println(capitalizeFirstLetter("hello world from golang")) // Output: "Hello World From Golang"

}

**10. Count the Number of Vowels in a String**

go

CopyEdit

package main

import "fmt"

func countVowels(s string) int {

vowels := "aeiouAEIOU"

count := 0

for \_, char := range s {

if strings.ContainsRune(vowels, char) {

count++

}

}

return count

}

func main() {

fmt.Println(countVowels("hello world")) // Output: 3

}

**11. Check if a String Contains Only Digits**

go

CopyEdit

package main

import (

"unicode"

"fmt"

)

func isOnlyDigits(s string) bool {

for \_, char := range s {

if !unicode.IsDigit(char) {

return false

}

}

return true

}

func main() {

fmt.Println(isOnlyDigits("12345")) // Output: true

fmt.Println(isOnlyDigits("123a5")) // Output: false

}

**12. Split a String by a Delimiter**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func splitByDelimiter(s, delimiter string) []string {

return strings.Split(s, delimiter)

}

func main() {

fmt.Println(splitByDelimiter("apple,banana,cherry", ",")) // Output: ["apple", "banana", "cherry"]

}

**13. Find the Index of a Substring**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func indexOf(s, subStr string) int {

return strings.Index(s, subStr)

}

func main() {

fmt.Println(indexOf("hello world", "world")) // Output: 6

fmt.Println(indexOf("hello world", "golang")) // Output: -1

}

**14. Concatenate Multiple Strings**

go

CopyEdit

package main

import "fmt"

func concatenateStrings(strs []string) string {

result := ""

for \_, str := range strs {

result += str

}

return result

}

func main() {

strs := []string{"Go", "Lang", "is", "fun!"}

fmt.Println(concatenateStrings(strs)) // Output: "GoLangisfun!"

}

**15. Convert a String to Uppercase and Lowercase**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func toUpperAndLower(s string) (string, string) {

return strings.ToUpper(s), strings.ToLower(s)

}

func main() {

upper, lower := toUpperAndLower("Hello World")

fmt.Println(upper) // Output: "HELLO WORLD"

fmt.Println(lower) // Output: "hello world"

}

**Challenge Example: Replace Spaces with a Specific Character**

go

CopyEdit

package main

import (

"strings"

"fmt"

)

func replaceSpaces(s string, replacement rune) string {

return strings.ReplaceAll(s, " ", string(replacement))

}

func main() {

fmt.Println(replaceSpaces("hello world", '-')) // Output: "hello-world"

}

These Go functions address common string manipulation tasks and showcase how to use Go's standard library for string operations. Let me know if you need further explanation or more examples!

**You said:**

give input and output of question

**ChatGPT said:**

Sure! Here are the **inputs** and **outputs** for each of the functions mentioned above:

**3. Count Occurrences of a Substring**

**Input:**

* str: "hello world, hello"
* subStr: "hello"

**Output:**

* 2

**4. Remove Whitespace from a String**

**Input:**

* s: " hello world "

**Output:**

* "helloworld"

**5. Find the Longest Word in a Sentence**

**Input:**

* sentence: "Go is an amazing programming language"

**Output:**

* "programming"

**6. Check if Two Strings are Anagrams**

**Input:**

* str1: "listen"
* str2: "silent"

**Output:**

* true

**7. Convert a String to a Slice of Runes**

**Input:**

* s: "hello"

**Output:**

* [104 101 108 108 111]

**8. Remove All Duplicates from a String**

**Input:**

* s: "google"

**Output:**

* "gole"

**9. Capitalize the First Letter of Each Word**

**Input:**

* sentence: "hello world from golang"

**Output:**

* "Hello World From Golang"

**10. Count the Number of Vowels in a String**

**Input:**

* s: "hello world"

**Output:**

* 3

**11. Check if a String Contains Only Digits**

**Input:**

* s: "12345"

**Output:**

* true

**Input:**

* s: "123a5"

**Output:**

* false

**12. Split a String by a Delimiter**

**Input:**

* s: "apple,banana,cherry"
* delimiter: ","

**Output:**

* ["apple", "banana", "cherry"]

**13. Find the Index of a Substring**

**Input:**

* s: "hello world"
* subStr: "world"

**Output:**

* 6

**Input:**

* s: "hello world"
* subStr: "golang"

**Output:**

* -1

**14. Concatenate Multiple Strings**

**Input:**

* strs: ["Go", "Lang", "is", "fun!"]

**Output:**

* "GoLangisfun!"

**15. Convert a String to Uppercase and Lowercase**

**Input:**

* s: "Hello World"

**Output:**

* Uppercase: "HELLO WORLD"
* Lowercase: "hello world"

**Challenge Example: Replace Spaces with a Specific Character**

**Input:**

* s: "hello world"
* replacement: '-'

**Output:**

* "hello-world"

These are the inputs and outputs corresponding to each of the tasks and their respective functions. Each output is based on the behavior of the function when provided with the given input. Let me know if you need further clarification or additional examples!

**You said:**

what is substring?

**ChatGPT said:**

A **substring** is a contiguous sequence of characters within a string. In other words, it is any part of a string that you can extract by taking a subset of consecutive characters from that string.

For example, if you have the string "hello", some of its substrings are:

* "h"
* "he"
* "ell"
* "hello"
* "lo"

**Key Points:**

1. A substring can be as short as one character or as long as the entire string.
2. It must consist of consecutive characters (no skipping).
3. The string itself is considered a substring of itself.
4. A substring can be empty (i.e., a string with no characters), which is technically a valid substring.

**Example:**

If you have the string:

go

CopyEdit

s := "hello"

* "hello" is a substring (the entire string).
* "ell" is a substring.
* "lo" is a substring.
* "h" is a substring.
* "o" is a substring.
* An empty string "" is also considered a substring.

**Finding Substrings:**

In programming, you can find or check if a string contains a substring using various methods, such as strings.Contains, strings.Index, or even slicing in Go.

For example:

go

CopyEdit

s := "hello"

substr := "ell"

if strings.Contains(s, substr) {

fmt.Println("Substring found!")

}

In this case, "ell" is a substring of "hello".