Fondamenti di Informatica (Elettronici)

THINK JULIA – Capitolo 8 (seconda parte)

9 novembre 2020

8. Stringhe (seconda parte)[1]

1. Interpolazione di stringhe
2. Ricerca
3. Loop e conteggio
4. Libreria per le stringhe
5. L'operatore $\in$
6. Confronto di stringhe
7. Debug
8. Glossario
9. Esercizi

[1] Tratto da https://benlauwens.github.io/ThinkJulia.jl/latest/book.html, disponibile sotto Licenza 'Creative Commons Attribution-NonCommercial 3.0 Unported'.

Section 1

Interpolazione di stringhe

Interpola qualsiasi espressione in una stringa

La costruzione di stringhe usando la concatenazione può diventare un po complicata. Per ridurre la necessità di queste chiamate dettagliate a stringhe o moltiplicazioni ripetute, Julia consente l'interpolazione di stringhe utilizzando il carattere $:

```
julia> greet = "Hello"
"Hello"
julia> whom = "World"
"World"
julia> "$greet, $(whom)!"
```

Interpola qualsiasi espressione in una stringa

La costruzione di stringhe usando la concatenazione può diventare un po complicata. Per ridurre la necessità di queste chiamate dettagliate a stringhe o moltiplicazioni ripetute, Julia consente l'interpolazione di stringhe utilizzando il carattere $:

```
julia> greet = "Hello"
"Hello"
julia> whom = "World"
"World"
julia> "$greet, $(whom)!"
```

È più leggibile e conveniente della concatenazione di stringhe:
salutare "," chi "!"

L'espressione completa più breve dopo $ viene considerata come l'espressione il cui valore deve essere interpolato come stringa. Pertanto, puoi interpolare qualsiasi espressione in una stringa usando le parentesi:

```
julia> "1 + 2 = $(1 + 2)"
```

Section 2

Ricerca

Pattern di ricerca 1/2

Cosa fa la seguente funzione?

```
function find(word, letter)
    index = firstindex(word)
    while index <= sizeof(word)
        if word[index] == letter
            return index
        end
        index = nextind(word, index)
    end
    -1
end
```

In un certo senso, "find" è l'inverso dell'operatore "[]". Invece di prendere un indice ed estrarre il carattere corrispondente, prende un carattere e trova l'indice in cui appare quel carattere. Se il carattere non viene trovato, la funzione restituisce "-1".

Pattern di ricerca 2/2

Questo è il primo esempio che abbiamo visto di un'istruzione return all'interno di un ciclo. Se "word [index] == letter", la funzione esce dal ciclo e ritorna immediatamente. Se il carattere non compare nella stringa, il programma esce normalmente dal ciclo e restituisce "-1".

Questo modello (pattern) di calcolo — attraversare una sequenza e tornare quando troviamo quello che stiamo cercando — è chiamato ricerca.

Esercizio 8-4

Modificare `find` in modo che abbia un terzo parametro, l'indice nella parola in cui dovrebbe iniziare a cercare.

Section 3

Loop e conteggio

Pattern di contatore

Il seguente programma conta il numero di volte in cui la lettera a appare in una stringa:

```
word = "banana"
counter = 0
for letter in word
    if letter == 'a'
        global counter = counter + 1
    end
end
println(counter)
3
```

Questo programma mostra un altro modello di calcolo chiamato contatore.

La variabile `counter` viene inizializzata a 0 e quindi incrementata ogni volta che viene trovata una a .
Quando il ciclo termina, `counter` contiene il risultato — il numero totale di a

Section 4

Libreria per le stringhe

Funzione uppercase

Julia fornisce funzioni che eseguono una serie di operazioni utili sulle stringhe

Ad esempio, la funzione `uppercase` accetta una stringa e restituisce una nuova stringa con tutte le lettere maiuscole.

```
julia> uppercase("Hello, World!")
"HELLO, WORLD!"
```

A quanto pare, esiste una funzione chiamata `findfirst` che è notevolmente simile alla funzione `find` che abbiamo scritto:

```
julia> findfirst("a", "banana")
2:2
```

Funzione `findfirst`

In realtà, la funzione `findfirst` è più generale della nostra funzione; può trovare sottostringhe, non solo caratteri:

```
julia> findfirst("na", "banana")
3:4
```

Per impostazione predefinita, `findfirst` inizia all'inizio della stringa, ma la funzione `findnext` accetta un terzo argomento, `index` dove dovrebbe iniziare:

```
julia> findnext("na", "banana", 4)
5:6
```

Section 5

L'operatore ∈

Operatore ∈

L'operatore ∈ (`\in TAB`) è un operatore booleano che accetta un carattere e una stringa e restituisce "`true`" se il primo appare nel secondo:

```
julia> 'a' in "banana" # 'a' in "banana"
true
```

Ad esempio, la seguente funzione stampa tutte le lettere di "word1" che appaiono anche in "word2":

```
function inboth(word1, word2)
    for letter in word1
        if letter in word2
            print(letter, " ")
        end
    end
end
```

Esempio

Con nomi di variabili ben scelti, Julia a volte si legge come l'inglese.

Potresti leggere questo ciclo: "per (ogni) lettera nella (prima) parola, se (la) lettera è un elemento della (seconda) parola, scrivi (la) lettera."

Ecco cosa ottieni se confronti "`apples`" e "`oranges`":

```
julia> inboth("apples", "oranges")
a e s
```

Section 6

Confronto di stringhe

Gli operatori relazionali lavorano sulle stringhe

Gli operatori relazionali lavorano sulle stringhe. Per vedere se due stringhe sono uguali:

```
word = "Pineapple"
if word == "banana"
    println("All right, bananas.")
end
```

Altre operazioni relazionali sono utili per mettere le parole in ordine alfabetico:

```
if word < "banana"
    println("Your word, $word, comes before banana.")
elseif word > "banana"
    println("Your word, $word, comes after banana.")
else
    println("All right, bananas.")
end
```

Avviso sull'ordinamento delle lettere

Julia non gestisce le lettere "maiuscole" e "minuscole" allo stesso modo delle persone.

Tutte le lettere maiuscole vengono prima di tutte le lettere minuscole, quindi:

La parola "Pineapple" viene prima di "banana".

Avviso sull'ordinamento delle lettere

Julia non gestisce le lettere "maiuscole" e "minuscole" allo stesso modo delle persone.

Tutte le lettere maiuscole vengono prima di tutte le lettere minuscole, quindi:

La parola "Pineapple" viene prima di "banana".

Avviso sull'ordinamento delle lettere

Julia non gestisce le lettere "maiuscole" e "minuscole" allo stesso modo delle persone.

Tutte le lettere maiuscole vengono prima di tutte le lettere minuscole, quindi:

La parola "Pineapple" viene prima di "banana".

SUGGERIMENTO

Un modo comune per risolvere questo problema è convertire le stringhe in un formato standard, come tutte le lettere minuscole, prima di eseguire il confronto.

Section 7

Debug

aaaa

aaaa

Section 8

Glossario

Glossario I

sequenza Una raccolta ordinata di valori in cui ogni valore è identificato da un indice intero.

Glossario I

sequenza Una raccolta ordinata di valori in cui ogni valore è identificato da un indice intero.

standard **ASCII Uno standard di codifica dei caratteri per la comunicazione elettronica che specifica 128 caratteri.

Glossario I

sequenza Una raccolta ordinata di valori in cui ogni valore è identificato da un indice intero.

standard **ASCII Uno standard di codifica dei caratteri per la comunicazione elettronica che specifica 128 caratteri.

standard Unicode Uno standard del settore informatico per la codifica, la rappresentazione e la gestione coerenti del testo espresso nella maggior parte dei sistemi di scrittura del mondo.

Glossario I

sequenza Una raccolta ordinata di valori in cui ogni valore è identificato da un indice intero.

standard **ASCII Uno standard di codifica dei caratteri per la comunicazione elettronica che specifica 128 caratteri.

standard Unicode Uno standard del settore informatico per la codifica, la rappresentazione e la gestione coerenti del testo espresso nella maggior parte dei sistemi di scrittura del mondo.

indice Un valore intero utilizzato per selezionare un elemento in una sequenza, ad esempio un carattere in una stringa. In Julia gli indici partono da 1.

Glossario I

sequenza Una raccolta ordinata di valori in cui ogni valore è identificato da un indice intero.

standard **ASCII Uno standard di codifica dei caratteri per la comunicazione elettronica che specifica 128 caratteri.

standard Unicode Uno standard del settore informatico per la codifica, la rappresentazione e la gestione coerenti del testo espresso nella maggior parte dei sistemi di scrittura del mondo.

indice Un valore intero utilizzato per selezionare un elemento in una sequenza, ad esempio un carattere in una stringa. In Julia gli indici partono da 1.

codifica UTF-8 Una codifica di caratteri a larghezza variabile in grado di codificare tutti i 1.112.064 elementi di codice validi in Unicode utilizzando da uno a quattro byte a 8 bit.

Glossario I

sequenza Una raccolta ordinata di valori in cui ogni valore è identificato da un indice intero.

standard **ASCII Uno standard di codifica dei caratteri per la comunicazione elettronica che specifica 128 caratteri.

standard Unicode Uno standard del settore informatico per la codifica, la rappresentazione e la gestione coerenti del testo espresso nella maggior parte dei sistemi di scrittura del mondo.

indice Un valore intero utilizzato per selezionare un elemento in una sequenza, ad esempio un carattere in una stringa. In Julia gli indici partono da 1.

codifica UTF-8 Una codifica di caratteri a larghezza variabile in grado di codificare tutti i 1.112.064 elementi di codice validi in Unicode utilizzando da uno a quattro byte a 8 bit.

attraversare Per scorrere gli elementi in una sequenza, eseguendo un'operazione simile su ciascuno.

Glossario II

slice Una parte di una stringa specificata da un intervallo di indici.

Glossario II

slice Una parte di una stringa specificata da un intervallo di indici.

stringa vuota Una stringa senza caratteri e di lunghezza 0, rappresentata da due virgolette.

Glossario II

slice Una parte di una stringa specificata da un intervallo di indici.

stringa vuota Una stringa senza caratteri e di lunghezza 0, rappresentata da due virgolette.

immutabile La proprietà di una sequenza i cui elementi non possono essere modificati.

Glossario II

slice Una parte di una stringa specificata da un intervallo di indici.

stringa vuota Una stringa senza caratteri e di lunghezza 0, rappresentata da due virgolette.

immutabile La proprietà di una sequenza i cui elementi non possono essere modificati.

interpolazione di stringhe Il processo di valutazione di una stringa contenente uno o più segnaposto, che produce un risultato in cui i segnaposto vengono sostituiti con i valori corrispondenti.

Glossario II

slice Una parte di una stringa specificata da un intervallo di indici.

stringa vuota Una stringa senza caratteri e di lunghezza 0, rappresentata da due virgolette.

immutabile La proprietà di una sequenza i cui elementi non possono essere modificati.

interpolazione di stringhe Il processo di valutazione di una stringa contenente uno o più segnaposto, che produce un risultato in cui i segnaposto vengono sostituiti con i valori corrispondenti.

ricerca Un pattern di attraversamento che si interrompe quando trova ciò che sta cercando.

Glossario II

slice Una parte di una stringa specificata da un intervallo di indici.

stringa vuota Una stringa senza caratteri e di lunghezza 0, rappresentata da due virgolette.

immutabile La proprietà di una sequenza i cui elementi non possono essere modificati.

interpolazione di stringhe Il processo di valutazione di una stringa contenente uno o più segnaposto, che produce un risultato in cui i segnaposto vengono sostituiti con i valori corrispondenti.

ricerca Un pattern di attraversamento che si interrompe quando trova ciò che sta cercando.

contatore Una variabile usata per contare qualcosa, solitamente inizializzata a zero e poi incrementata.

Section 9

Esercizi

aaaa

aaaa