

CS381: Numerical Computation & Softwares

Introduction to Julia programming (2)

Jakramate Bootkrajang

Department of Computer Science

Chiang Mai University

Rev.2 – 2019

Outline

- Character and String.
- Array, string and indexing.

Character and String

- อักขระ (Character) ใช้เรียกตัวอักษร ตัวเลข และสัญลักษณ์ต่างๆที่แสดงได้ในโปรแกรม
 - ▶ ตัวอย่างเช่น '1', '0'
 - ▶ ใน Julia เราสามารถเขียนอักขระโดยใช้เครื่องหมาย ' (single quote) ครอบอักขระ ตัวเลข หรือสัญลักษณ์ที่เราต้องการ
- สายอักขระ (String) ใช้เรียกกลุ่มของอักขระหลายตัวประกอบกัน
 - ▶ "10001"
 - ▶ ใน Julia เราสามารถสร้างสายอักขระโดยใช้เครื่องหมาย " (double quote) ครอบกลุ่มของอักขระที่เราต้องการ

Character and Number

เราสามารถใช้ฟังก์ชัน `typeof()` เพื่อเรียกดูชนิดของตัวแปร

```
In [18]: typeof(1)
```

```
Out[18]: Int64
```

```
In [19]: typeof('1')
```

```
Out[19]: Char
```

เราไม่สามารถกระทำการทางคณิตศาสตร์ทุกอย่างกับอักขระได้

```
In [27]: '1' + '9'
```

```
MethodError: no method matching +(::Char, ::Char)
Closest candidates are:
  +(::Any, ::Any, ::Any, ::Any...) at operators.jl:424
  +(::Char, ::Integer) at char.jl:40
  +(::Integer, ::Char) at char.jl:41
```

```
In [28]: '1' < '6'
```

```
Out[28]: true
```

Parsing number from character

หากต้องการคำนวณทางคณิตศาสตร์กับอักขระ จะต้องแปลงอักขระให้เป็นตัวเลขก่อน โดยใช้ฟังก์ชัน `parse()`

```
In [32]: # parse(type, string, base)
          parse(Int64, '1', 2)
```

```
Out[32]: 1
```

หมายเหตุ: ใน Julia Version 1.0.3 ฟังก์ชัน `parse` ถูกย้ายไปอยู่ใน Module `Base` เวลาเรียกใช้ให้ระบุชื่อโมดูลนำหน้าด้วย `Base.parse(Int64, '1', base=2)`

Parsing number from string

เราสามารถแปลงสายอักขระเป็นตัวเลขได้โดยใช้ฟังก์ชัน `parse()` เช่นเดียวกัน

```
In [33]: parse{Int64, "11011", 10}
```

```
Out[33]: 11011
```

```
In [34]: parse{Int64, "11011", 2}
```

```
Out[34]: 27
```

หมายเหตุ: ใน Julia Version 1.0.3 ฟังก์ชัน `parse` ถูกย้ายไปอยู่ใน Module `Base` เวลาเรียกใช้ให้ระบุชื่อโมดูลนำหน้าด้วย `Base.parse{Int64, '1', base=2}`

Array construction and indexing

การประกาศอะเรย์เพื่อใช้เก็บข้อมูลสามารถทำได้ง่ายๆโดยเลือกใช้ฟังก์ชัน `zeros(n,m)`

```
In [10]: a = zeros(1,10)
         println(a)
```

```
[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0]
```

```
In [8]: a[3] = 5
        println(a)
```

```
[0.0 0.0 5.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0]
```

ซึ่งจะสามารถสร้างอะเรย์ที่สมาชิกทุกตัวเป็น 0 จากนั้นเราจึงค่อยไปกำหนดค่าของสมาชิกแต่ละตัว

String is an array of characters

สายอักขระความจริงก็คืออะเรย์ที่ประกอบด้วยอักขระหลายตัวต่อกัน เราสามารถเข้าถึงอักขระแต่ละตัวได้โดยใช้หลักการของการเข้าถึงอะเรย์

```
In [5]: a = "10001111"  
  
println(a[1])  
println(a[1:5])  
println(a[5:8])  
println(a[5:end])  
println(a[0])
```

```
1  
10001  
1111  
1111
```

```
BoundsError: attempt to access "10001111"  
at index [0]
```

Must read

- Learn X in Y minutes for Julia

<https://learnxinyminutes.com/docs/julia/>