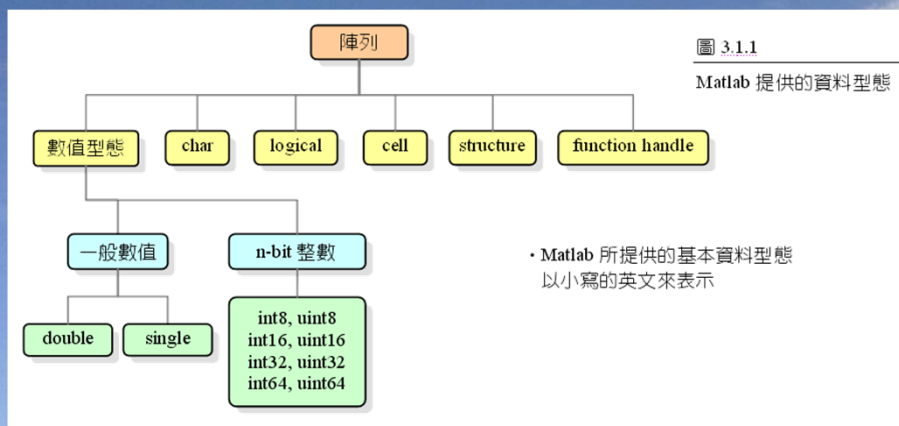


第三章 資料型態與輸出控制

本章學習目標

- 認識Matlab的基本資料型態
- 練習資料型態的轉換
- 學習如何控制Matlab的輸出格式
- 學習使用fprintf() 函數來控制輸出

Matlab的資料型態



一般數值型態

- 「一般數值」可分為single與double兩種型態

表 3.1.1 單精度與倍精度型態

資料型態	說明	位元組	最大的正數	最小的正數
single	單精度	4	3.4028×10^{38}	1.1755×10^{-38}
double	倍精度	8	1.7977×10^{308}	2.2251×10^{-308}

```
>> a=12.4
a =
    12.4000
```

```
>> whos a
Name Size Bytes Class
a      1x1      8 double
```

```
>> b=single(3.8)
b =
    3.8000
```

```
>> whos b
Name Size Bytes Class
b      1x1      4 single
```

3

n-bit整數型態

- n-bit整數可分為有號（signed）與無號（unsigned）兩種

表 3.1.2 n-bit 整數型態

資料型態	說明	位元組	最小值	最大值
int8	8-bit 整數	1	-128	127
uint8	8-bit 無號整數	1	0	255
int16	16-bit 整數	2	-32768	32767
uint16	16-bit 無號整數	2	0	65535
int32	32-bit 整數	4	-2147483648	2147483647
uint32	32-bit 無號整數	4	0	4294967295
int64	64-bit 整數	8	-9223372036854775808	9223372036854775807
uint64	64-bit 無號整數	8	0	18446744073709551615

4

n-bit整數的範例

```
>> m=[13 120 30; 36 42 112]
m =
    13    120    30
    36    42   112
```

```
>> m1=uint8(m)
m1 =
    13    120    30
    36    42   112
```

```
>> uint8([12 300 -250])
ans =
    12   255     0
```

```
>> int8(120)+int16(250)
Error using +
Integers can only be combined with integers
of the same class, or scalar doubles.
```

```
>> uint8(12)+uint8(64)
ans =
    76
```

```
>> uint8(12)^3
ans =
    255
```

```
>> 2^60
ans =
  1.1529e+18
```

```
>> int64(2)^60
ans =
  1152921504606846976
```

查詢數值資料型態的範圍

表 3.1.3 查詢數值資料型態所能表示的範圍的函數

函 數	說 明
<code>realmax('data type')</code>	查詢所指定之一般數值資料型態的最大值，其中 <i>data type</i> 可為 <code>single</code> 或 <code>double</code>
<code>realmin('data type')</code>	同 <code>realmax</code> ，不過是查詢最小的正數
<code>intmax('data type')</code>	查詢所指定之整數資料型態的最大正整數，其中 <i>data type</i> 可為 <code>int8</code> 、 <code>int16</code> 、 <code>int32</code> 、 <code>int64</code> 、 <code>uint8</code> 、 <code>uint16</code> 、 <code>uint32</code> 或 <code>uint64</code> 任何一個
<code>intmin('data type')</code>	同 <code>intmax</code> ，不過是查詢最小的整數
<code>class(var)</code>	查詢變數 <i>var</i> 的資料型態

```
>> realmax('double')
ans =
  1.7977e+308
```

```
>> realmin('single')
ans =
  1.1755e-38
```

```
>> intmax('int64')
ans =
  9223372036854775807
```

字元資料型態

- 在Matlab裡，字元是以成對的單引號括起來
- 每一個字元佔了兩個bytes。

```
>> ch='A'
ch =
A
>> double(ch)
ans =
    65
>> char(65)
ans =
A
>> ch+1
ans =
    66
```

```
>> char(65:90)
ans =
ABCDEFGHIJKLMNOPQRSTUVWXYZ
>> str='a string'
str =
a string
>> asc=double(str)
ans =
    97    32   115   116   114   105   110   103
>> char(asc)
ans =
a string
```

7

邏輯資料型態

- Matlab以1代表運算結果為true，以0代表運算結果為false
- 邏輯資料型態的變數佔了一個位元組

```
>> t1=isprime(13)
t1 =
    1
>> t2=(3>6)
t2 =
    0
>> t3=0
t3 =
    0
```

```
>> t4=logical(t3)
t4 =
    0
>> logical(-7)
ans =
    1
>> logical([12 1 0 -9.4])
ans =
    1    1    0    1
```

8

控制Matlab的顯示方式

表 3.2.1 控制顯示或不顯示運算結果

敘述型式	說明
敘述 1, 敘述 2, 敘述 3;	執行敘述 1~3, 但敘述 3 的結果不顯示
敘述 1; 敘述 2; 敘述 3;	執行敘述 1~3, 且每一個結果均不顯示

```
>> a=3,b=4,c=5;
```

```
a =  
    3
```

```
b =  
    4
```

```
>> x=3;y=4;z=5
```

```
z =  
    5
```

```
>> p=primes(1000);
```

```
>> length(p)
```

```
ans =  
    168
```

```
>> max(p)
```

```
ans =  
    997
```

9

指令跨行的控制

- 敘述較長無法撰寫在同一行時，可利用跨行符號「...」

```
>> sin(1.4)-cos(3.14)*12+...
```

```
tan(0.2)
```

```
ans =  
    13.1881
```

```
>> sin(1.4)-cos(3.14)*12 ...
```

```
+tan(0.2)
```

```
ans =  
    13.1881
```

```
>> A=[200 300 500; 400 600 700;...
```

```
100 400 300]
```

```
A =  
    200    300    500  
    400    600    700  
    100    400    300
```

10

控制Matlab的輸出格式

表 3.2.2 控制 Matlab 的輸出格式

格式指令	說明
<code>format</code>	Matlab 的預設格式，數值的小數部分是以 4 個位數來顯示。當數值是整數時，若位數小於或等於 9，則以整數來顯示，否則以指數的型式來表示
<code>format short</code>	精簡格式，其格式同 <code>format</code>
<code>format short g</code>	若數值為整數，格式同 <code>format</code> ，若數值帶有小數，則以總共 5 個位數（整數加小數）來顯示數值部分
<code>format short e</code>	若數值為整數，格式同 <code>format</code> ，若數值帶有小數，以指數的型式來顯示
<code>format long</code>	完整格式，以 16 個位數來顯示數值。若數值大於 100 或小於 0.001，則以指數型式來表示
<code>format long g</code>	完整格式，以整數位數加小數位數，共 15 個位數來顯示數值
<code>format long e</code>	完整格式，以指數型式來顯示完整格式
<code>format compact</code>	簡潔格式，即在指令輸入與結果輸出之間不留任何空行
<code>format loose</code>	寬鬆格式，即在指令輸入與結果輸出之間空一行

格式化輸出 fprintf()

函數	說明
<code>Fprintf('str',e1,e2,...)</code>	<p>依格式字串str所記載的格式碼，依序將e1,e2,...入str中列印出來。下面列出了格式字串裡常用的格式碼</p> <p>%c 列印字元</p> <p>%s 列印字串</p> <p>%md 以m個欄位的寬度列印整數，若省略m，則以最精簡的格式來列印</p> <p>%m.nf 以n個小數位，總共m個欄位的寬度列印數值，若省略m,n，則以6個位數的小數來列印，自動在數值後補上0，使得小數位數湊滿</p> <p>%m.ne 同上，但以指數型式來列印數值</p> <p>%m.ng 以m個欄位，n個有效位數來列印數值。如果省略m,n，則以最精簡的格式來列印</p>

用於fprintf()的特殊字元

特殊字元	說明
\n	換行
\t	跳格
‘’	印出單引號
\\	印出反斜線
%%	印出百分比符號

13

fprintf() 的範例

```
a=22;
fprintf('a=%6.3f\n',a);

b=3.14159;
fprintf('b=%5.2f\n',b);
```

a	=	2	2	.	0	0	0	\n
---	---	---	---	---	---	---	---	----

%6.3f, 佔 6 個欄位, 小數點以下 3 位

b	=	3	.	1	4	\n
---	---	---	---	---	---	----

%5.2f, 佔 5 個欄位, 小數點以下 2 位

```
>> a=22; b=3.14159; c='@';
>> fprintf('a=%6.3f\n',a);
a=22.000
>> fprintf('b=%5.2f\n',b);
b= 3.14
```

```
>> fprintf('a=%5d, b=%7.4f\n',a,b);
a= 22, b= 3.1416
>> fprintf('c=%c\n',c);
c=@
>> fprintf('\n');
```

14

%g 格式碼

- %g 用於想列印的數可能是整數或帶有小數的浮點數時
- %g 格式碼不會在小數點之後補上0，可讓數字看起來簡潔

```
>> fprintf('%f\n',2.94)
2.940000
```

```
>> fprintf('%f\n',256)
256.000000
```

```
>> fprintf('%g\n',2.94)
2.94
```

```
>> fprintf('%g\n',256)
256
```

```
>> fprintf('%6.3g\n',3.14159)
3.14
```

```
>> fprintf('%6.4g\n',3.14159)
3.142
```

15

disp() 函數

- disp() 函數可用在要顯示的內容較為簡單時
- disp是display的縮寫，也就是顯示的意思

```
>> disp(a)
22
```

```
>> disp([a b])
22.0000    3.1416
```

```
>> disp('A sunny day')
A sunny day
```

16