Strings

Τα strings μπορούν να αντιστοιχηθούν σε μεταβλητές δηλώνοντας τα με απόστροφο

```
>>s = ' Hi there ' ;
```

Αν πρέπει να θέσουμε και μια απόστροφο στο string τότε ο απόστροφος πρέπει να επαναληφθεί.

```
>>s = ' o ' ' clock ';
```

Με την εντολή input τα string μπορούν να εισαχθούν με δυο τρόπους ή με απόστροφους ή δηλώνοντας ένα όρισμα **'s'** στην εντολή input οπότε δεν πρέπει να χρησιμοποιούμε απόστροφους

```
>> name = input( ' Enter your surname: ' , ' s ' );
Enter your surname: kapoios
```

Τα string στο Matlab είναι διανύσματα

Ένα Matlab string είναι ένα διάνυσμα όπου κάθε στοιχείο παριστάνει έναν χαρακτήρα του string. Για παράδειγμα

```
>>s = ' Napoleon '
```

Η εντολή whos μας πληροφορεί ότι το s είναι 1-by-8. Η εντολή

```
>>s(8:-1:1)
```

θα μας δείξει το string 'Napoleon' αντίστροφα.

Υπάρχουν διάφοροι τρόποι για να ανακαλύψουμε την διανυσματική υφή των strings. Για παράδειγμα ο παρακάτω κώδικας σβήνει όλα τα κενά από ένα string.

```
>>s = ’Rock and Roll’;

>>nonblanks = s˜= ’ ’;

>>s(nonblanks)
```

Το noblanks είναι ένα λογικό διάνυσμα με το 0 να αντιστοιχεί στα στοιχεία του s που είναι κενά. Οπότε χρησιμοποιώντας το διάνυσμα noblanks στο s σαν υποδείκτη σβήνει τα κενά.

Επικόλληση strings

Επειδή τα strings είναι διανύσματα μπορούν να επικολληθούν με τετραγωνικές παρανθέσεις. Π.χ.

```
>>onoma = ’Maria’;

>>onoma = [Maria, ’ Magdalini’]
```

Κωδικοποίηση ASCII, οι εντολές double και char

Ένας χαρακτήρας στο Matlab παριστάνεται με μια 16-bit αριθμητική τιμή. Η κωδικοποίηση των χαρακτήρων ξεκινά από 1. Οι πρώτοι 127 κώδικες αντιστοιχούν στους κώδικες ASCII. Για παράδειγμα οι κώδικες ASCII των κεφαλαίων γραμμάτων του Αγγλικού αλφάβητου 'A' μέχρι 'Z' αντιστοιχούν στους διαδοχικούς ακεραίους από το 65 μέχρι το 90 ενώ των πεζών 'a' μέχρι 'z' από το 97 μέχρι το 122.

Μπορούμε την κωδικοποίηση ASCII ενός string με την εντολή double

```
>>double(’Napoleon’)

ans

78  97  112  111  108  101  111  110
```

Η κωδικοποίηση ASCII στο Matlab

Code	*Char*	*Code*	*Char*	*Code*	*Char*	*Code*	*Char*	*Code*	*Char*
0	(null)	26	→	52	4	78	N	104	h
1	☺	27	←	53	5	79	O	105	i
2	☻	28	∟	54	6	80	P	106	j
3	♡	29	↔	55	7	81	Q	107	k
4	◇	30	▲	56	8	82	R	108	l
5	♣	31	▼	57	9	83	S	109	m
6	♠	32	(blank)	58	:	84	T	110	n
7	•	33	!	59	;	85	U	111	o
8	◘	34	"	60	<	86	V	112	p
9	○	35	#	61	=	87	W	113	q
10	◙	36	$	62	>	88	X	114	r
11	♂	37	%	63	?	89	Y	115	s
12	♀	38	&	64	@	90	Z	116	t
13	♪	39	'	65	A	91	[	117	u
14	♫	40	(	66	B	92	\	118	v
15	☼	41	)	67	C	93	]	119	w
16	►	42	*	68	D	94	^	120	x
17	◄	43	+	69	E	95	-	121	y
18	↕	44	,	70	F	96	‘	122	z
19	!!	45	–	71	G	97	a	123	{
20	¶	46	.	72	H	98	b	124	|
21	§	47	/	73	I	99	c	125	}
22	▬	48	0	74	J	100	d	126	~
23	↨	49	1	75	K	101	e	127	Δ
24	↑	50	2	76	L	102	f		
25	↓	51	3	77	M	103	g		

Αντίστροφα η εντολή char μετατρέπει έναν κώδικα ASCII σε string

```
>>char(65:70)

ans

ABCDEF
```

Επίσης μπορούμε να χρησιμοποιήσουμε τις εντολές char και double για να δημιουργήσουμε πίνακες με ίδιους χαρακτήρες

```
>> x = char(ones(4,20)*double('#'))

x =

#######################

#######################

#######################

#######################
```

Αν ένας χαρακτήρας υπεισέρχεται σε μια αριθμητική πράξη το Matlab χρησιμοποιεί την τιμή της στον κώδικα ASCII για να εκτελέσει την πράξη

```
>>s = ’a’
```

Η πράξη s + 1 δίνει 98, αφού ο κώδικας για το ‘a’ είναι 97.

Οι πράξεις των διανυσμάτων μπορούν να εφαρμοσθούν στα strings. Για παράδειγμα αν s είναι ένα string γραμμάτων η έκφραση char(s+1) δίνει κάθε γράμμα στο s μετατοπισμένο κατά 1. Άλλο παράδειγμα είναι :

```
>>alpha = double('a'):double('z')
```

Σύγκριση string

Ας δοκιμάσουμε τις εντολές

```
>>s1 = ’ann’;

>>s2 = ’ban’;

>>s1 < s2

ans

1  0  0
```

Τι δηλώνει το αποτέλεσμα?

Η εντολή strcmp(s1, s2) συγκρίνει τα δυο string s1 και s2 και έχει ως αποτέλεσμα 1 αν είναι ταυτόσημα και 0 διαφορετικά. Τα string δεν είναι απαραίτητο να έχουν το ίδιο μήκος.

Συναρτήσεις για string

blanks Παράγει ένα string με κενά.

deblank Απομακρύνει κενά στο τέλος ενός string

int2str, num2str μετατρέπει τα αριθμητικά ορίσματα σε string.

ischar δίνει 1 αν το όρισμα είναι string, 0 διαφορετικά

lower, upper μετατρέπει string σε πεζά και κεφαλαία αντίστοιχα.

fprintf απεικονίζει στην οθόνη ένα string χρησιμοποιώντας το %s

fprintf(’%8sVIII\n’, ’Henry’)

sprintf δουλεύει σαν την fprintf

help matlab\strfun για πλήρη οδηγό βοήθειας σε συναρτήσεις που αφορούν string.

Δισδιάστατα string

Μπορούμε να δημιουργήσουμε δισδιάστατα strings όπως κάνοντας με πίνακες, εκτός από το ότι κάθε γραμμή πρέπει να έχει την ίδια διάσταση, αν είναι αναγκαίο πρέπει να βάλουμε κενά για παράδειγμα

```
>>nameAndAddress = [' Maria M ' ; ' 21 Akakia'; ' Kipos   ' ]
```

Ένας πιο εύκολος τρόπος για να δημιουργήσουμε πίνακες string είναι να χρησιμοποιήσουμε την εντολή char η οποία δημιουργεί αυτόματα τα απαραίτητα κενά

```
>> nameAndAddress = char('Maria','21 Akakia','Kipos')

nameAndAddress =

Maria

21 Akakia

Kipos

>> nameAndAddress(1:3)

ans =

M2K
```

eval και macros

Όταν ένα string έχει καταχωρηθεί σε μια μια μεταβλητή t, η συνάρτηση eval(t) κάνει την μεταβλητή t να υπολογισθεί. Αυτή η επιλογή λέγεται macro ευχέρεια

```
>>s = 'x = -b/(2*a);'

>>eval(s)

s =

x = -b/(2*a);
```

Μια άλλη χρήση της eval είναι ένας εύχρηστος τρόπος για να εισάγουμε μια συνάρτηση στο command window, πχ στο script

```
f = input( 'Enter function (of x) to be plotted: ', 's');

x = 0:0.01:10;

plot(x, eval(f)),grid

>> Enter function (of x) to be plotted:

exp(-0.5*x) .* sin(x)
```

- Η παράμετρος 's' δηλώνει ότι δεν χρειάζεται να βάλουμε απόστροφους