Builtin Functions & Data types

by Ahmet Sacan

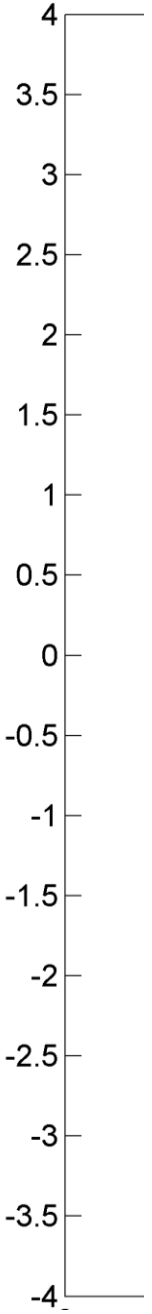
Built-in functions and help

- doc mod
- help mod
- Google

- a = abs(-4)
  - *call, argument, return*
- mod(13,5) vs. mod(5,13)
- abs, sign, floor, ceil, round, fix

Rounding Functions

- floor, ceil, round, fix
- floor(3.5)
- ceil(3.5)
- fix(3.5)
- round(3.4)
- round(2.5)
- floor(-3.5)
- ceil(-3.5)
- fix(-3.5)
- round(-3.4)
- round(-2.5)



Constants

- pi()
- i()
- j
- inf
- nan
- Exercise: find the result of:
  - 10 / 0
  - 0 / 10
  - 10 / 0 * 0
  - inf * 0
  - 0 / 0
  - inf + 1
  - inf + inf
  - inf - inf

Bits, Bytes, Binary, Hexadecimal

- bit=0/1
- byte=8bits (in other programming languages, byte=char)
- a=17 (decimal)
- a=010001 (binary)
- a=0x11 (hexadecimal)
- B=0x0A : 10
- B=0x0F : 15
- B= 0xF1 : 241

Data Types

- http://www.mathworks.com/help/techdoc/matlab_prog/f2-43934.html
- Numeric
  - Integer: uint8, uint16, uint32, uint64, int8, int16, int32, int64, intmax('int8'), intmin('int8'), isinteger()
  - Floating-Point: double, single, realmax(), realmin(), zeros(), ones(), rand(), randi(), eye()
  - Complex: complex(), real(), imag()
- Logical: true, false
- char
- class(), whos(), isa(x,'uint32')
- Data Conversion: implicit, explicit

Exercises

- Why would you use integer type instead of double?
- intmin('int8') =?
- intmax('int8') =?
- int8(200) =?
- int8(-130) =?

Floating Point Representation & Anomalies

- $significant\ bits\ * base^{exponent}$
- Special Values
  - inf(), isinf()
  - NaN(), isnan()
- Round-Off Errors:
  - 1 - 3*(4/3 - 1)
  - 0.1+0.1+0.1 == 0.3
  - (2^53 + 1) - 2^53
  - sin(pi)

Characters & Encoding

- int8('a')
- 'a'+1
- char('a'+1)
- double('abcd')
- Shifting by 1:
- char('abcd')+1

- Note: when parentheses is omitted from a function call, the arguments (if any) are assumed to be text.
  - int8 abcd

Dec	Name	Char	Dec	Char	Dec	Char	Dec	Char
0	Null	NUL	32	Space	64	@	96	`
1	Start of heading	SOH	33	!	65	A	97	a
2	Start of text	STX	34	"	66	B	98	b
3	End of text	ETX	35	#	67	C	99	c
4	End of xmit	EOT	36	$	68	D	100	d
5	Enquiry	ENQ	37	%	69	E	101	e
6	Acknowledge	ACK	38	&	70	F	102	f
7	Bell	BEL	39	'	71	G	103	g
8	Backspace	BS	40	(	72	H	104	h
9	Horizontal tab	HT	41	)	73	I	105	i
10	Line feed	LF	42	*	74	J	106	j
11	Vertical tab	VT	43	+	75	K	107	k
12	Form feed	FF	44	,	76	L	108	l
13	Carriage feed	CR	45	-	77	M	109	m
14	Shift out	SO	46	.	78	N	110	n
15	Shift in	SI	47	/	79	O	111	o
16	Data line escape	DLE	48	0	80	P	112	p
17	Device control 1	DC1	49	1	81	Q	113	q
18	Device control 2	DC2	50	2	82	R	114	r
19	Device control 3	DC3	51	3	83	S	115	s
20	Device control 4	DC4	52	4	84	T	116	t
21	Neg acknowledge	NAK	53	5	85	U	117	u
22	Synchronous idle	SYN	54	6	86	V	118	v
23	End of xmit block	ETB	55	7	87	W	119	w
24	Cancel	CAN	56	8	88	X	120	x
25	End of medium	EM	57	9	89	Y	121	y
26	Substitute	SUB	58	:	90	Z	122	z
27	Escape	ESC	59	;	91	[	123	{
28	File separator	FS	60	<	92	\	124	|
29	Group separator	GS	61	=	93	]	125	}
30	Record separator	RS	62	>	94	^	126	~
31	Unit separator	US	63	?	95	_	127	DEL

adapted from:
https://www.commfront.com/pages/ascii-chart

Random Numbers

- rand()
- rng('shuffle')
- Exercise:
  - Generate a random number between 0 and 10.
  - Generate a random number between *low* and *high*. (e.g., test for low=7, high=10).
  - Generate a random integer between 1 and 10.