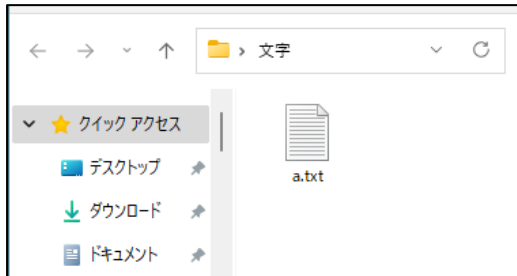


## 文字のデジタル表現 演習

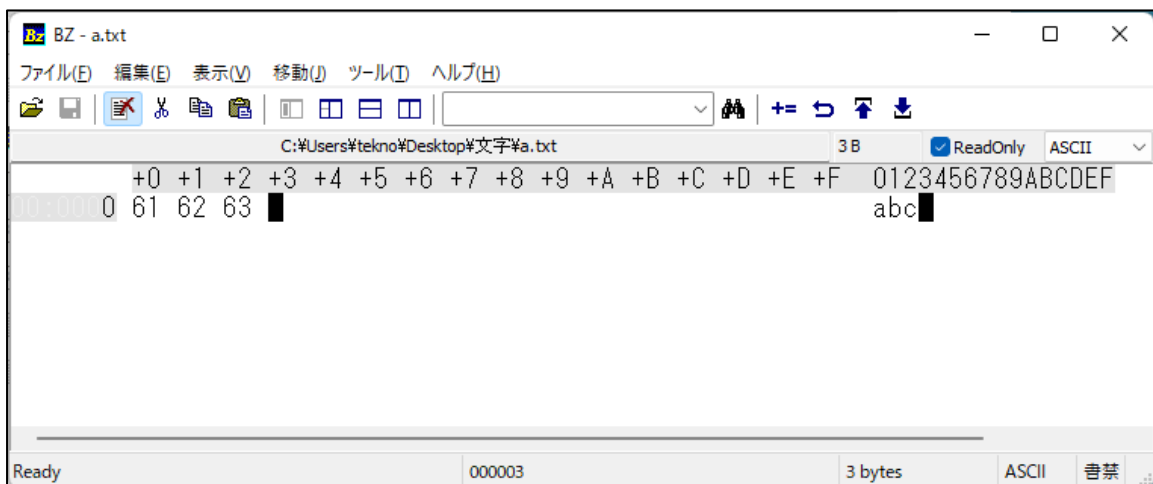
1. 任意の場所にテキストファイルを作成する。(ここでは「a.txt」)



2. そのファイルをテキストエディタ「メモ帳」で開き、文字の入力モードを半角英数モードにして「abc」と入力して、上書き保存する。



3. そのファイルをバイナリエディタ「BZ Editor」で開く。(BZ Editor を起動して、ウィンドウの中にテキストファイルをドラッグする)



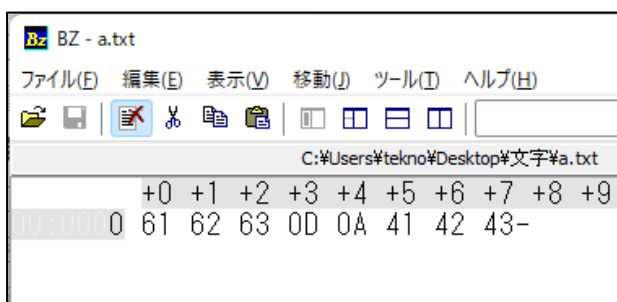
文字コード表どおりになっているか確認する。

JIS X 0201 (半角英数・半角カナ)																			
		上位4ビット																	
		2進数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
下 位 4 ビ ット	2進数	16進数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	0000	0	NUL	DLE	SP	0	@	P	`	p				ー	タ	ミ			
	0001	1	SOH	DC1	!	1	A	Q	a	q				。	ア	チ	ム		
	0010	2	STX	DC2	"	2	B	R	b	r				「	イ	ツ	メ		
	0011	3	ETX	DC3	#	3	C	S	c	s				」	ウ	テ	モ		
	0100	4	EOT	DC4	\$	4	D	T	d	t				、	エ	ト	ヤ		
	0101	5	ENQ	NAK	%	5	E	U	e	u				・	オ	ナ	ユ		
	0110	6	ACK	SYN	&	6	F	V	f	v				ヲ	カ	ニ	ヨ		
	0111	7	BEL	ETB	'	7	G	W	g	w				ア	キ	ヌ	ラ		
	1000	8	BS	CAN	(	8	H	X	h	x				イ	ク	ネ	リ		
	1001	9	HT	EM	)	9	I	Y	i	y				ウ	ケ	ノ	ル		
	1010	A	LF	SUB	*	:	J	Z	j	z				エ	コ	ハ	レ		
	1011	B	VT	ESC	+	;	K	[	k	{				オ	サ	ヒ	ロ		
	1100	C	FF	IS4	,	<	L	¥	l					ヤ	シ	フ	ワ		
	1101	D	CR	IS3	-	=	M	]	m	}				ユ	ス	ヘ	ン		
	1110	E	SO	IS2	.	>	N	^	n	~				ヨ	セ	ホ	、		
1111	F	SI	IS1	/	?	O	_	o	DEL				ッ	ソ	マ	・			

#### 4. メモ帳で改行をして、「ABC」と入力する



#### 5. バイナリエディタ「BZ Editor」で新規作成した後、ウィンドウにテキストファイルをドラッグしてもう一度開く。



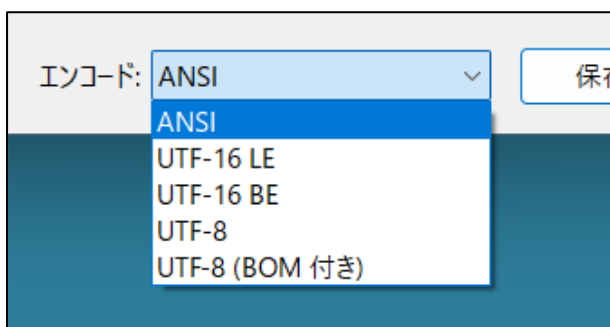
文字コード表通りか確認し、

「abc」と「ABC」の間のコード(0D と 0A)は何を意味しているか考える。

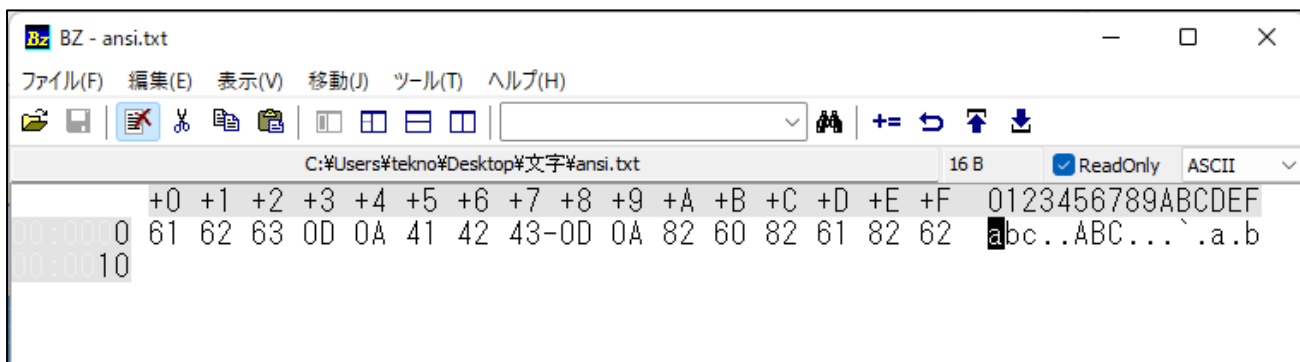
6. 改行して、入力モードをひらがなにしてい「ABC」と入力する



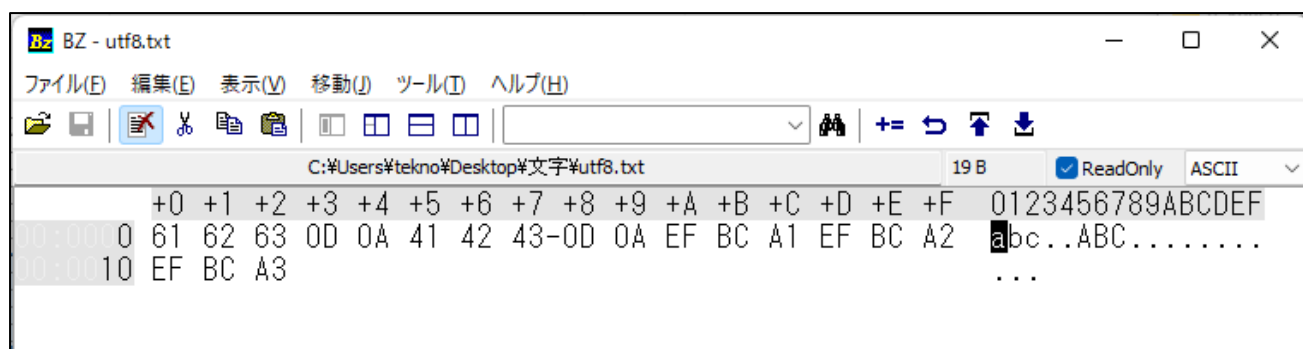
名前を付けて保存をして、エンコードを ANSI にしたものを「ansi.txt」、UTF-8 にしたものを「utf8.txt」とする



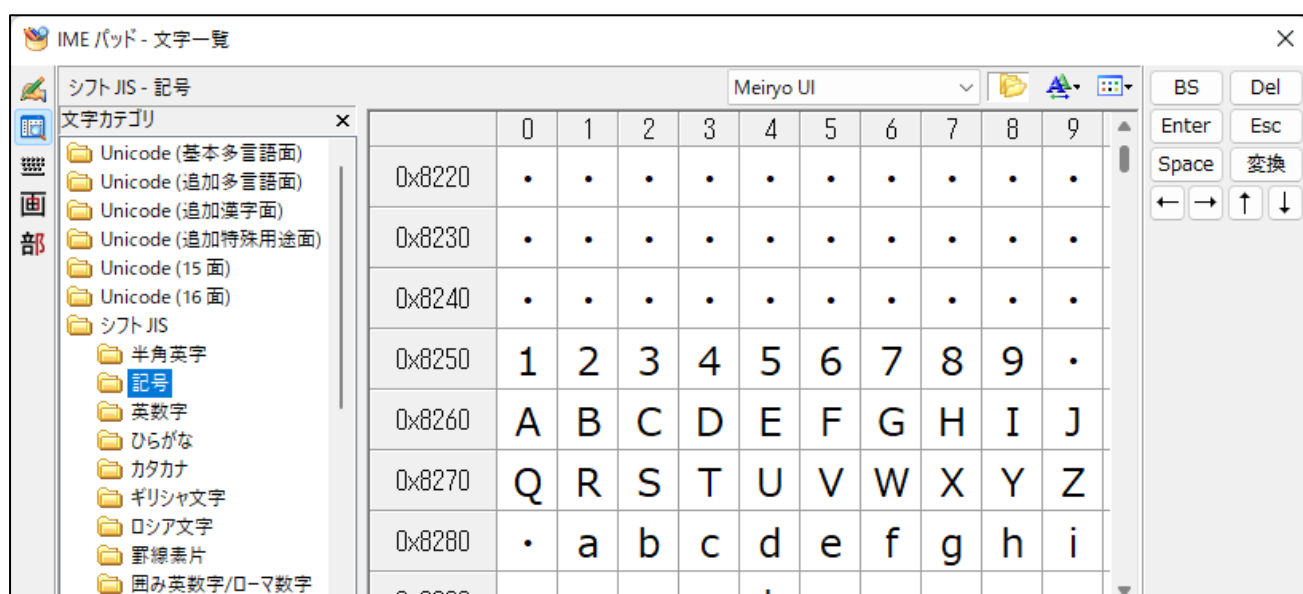
7. バイナリエディタで文字コードを確認する。  
(ANSI)



(UTF-8)

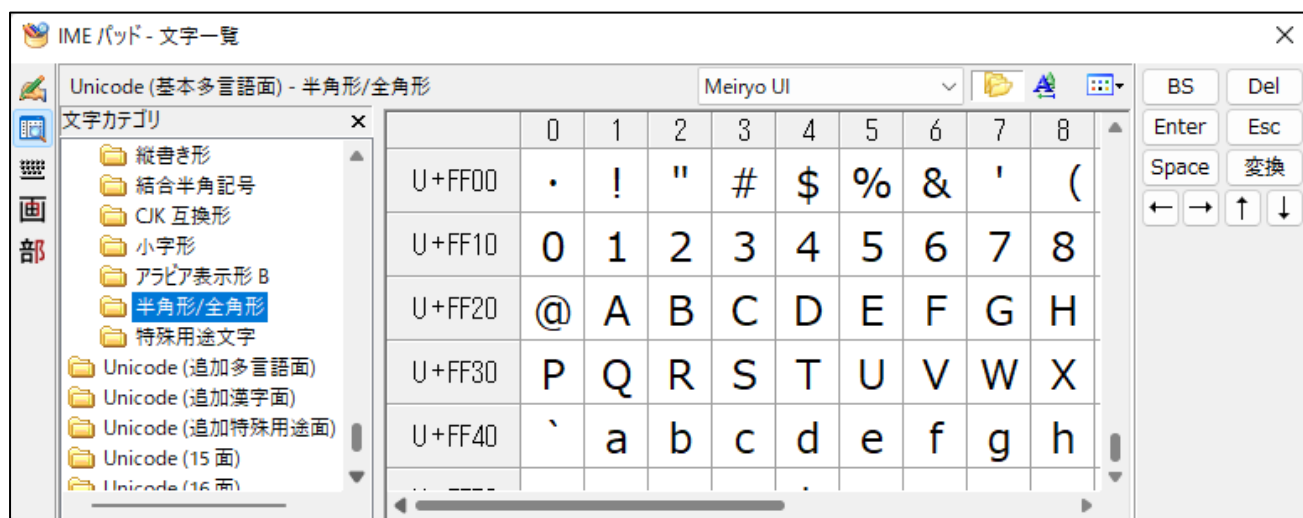


8. 文字コード表通りになっているか確認する。  
(ANSI) IME パッドの文字一覧の「シフト JIS」を見る。



※日本語 Windows で ANSI と呼ばれている Shift\_JIS は、Shift\_JIS に Microsoft が独自拡張を加えたもの。本来は CP932 という文字コード。

(UTF-8)



Unicode の「A」の場所を示すスカラ値は、「FF21」となっているが、  
バイナリエディタを見ると、「EFBCA1」と 3 バイトで記録されている。

理由は場所を示すスカラ値を、「UTF-8」の形にして保存してあるからで、  
A のスカラ値の「FF21」は

二進数に直すと 1111 1111 0010 0001

となり、

これを UTF-8 の形にすると、1110 1111 1011 1100 1010 0001

E F B C A 1

16進数で表記すると、「EFBCA1」となる。

(参考:<https://seiai.ed.jp/sys/text/csd/cf14/c14a070.html>)