

# 基礎コンピュータ工学

## 第2章 情報の表現

### (パート5：小数や文字の表現と補助単位)

<https://github.com/tctsigemura/TecTextBook>

本スライドの入手：



## 2進数による小数の表現

固定小数点方式（小数点の位置を約束する.）

$$00.00_2 = 0.0_{10}$$

$$00.01_2 = 0.25_{10}$$

$$00.10_2 = 0.5_{10}$$

$$00.11_2 = 0.75_{10}$$

$$01.00_2 = 1.0_{10}$$

$$01.01_2 = 1.25_{10}$$

$$01.10_2 = 1.5_{10}$$

$$01.11_2 = 1.75_{10}$$

$$10.00_2 = 2.0_{10}$$

...

$$11.11_2 = 3.75_{10}$$

桁の重み

- 小数点から左に進むと 2 倍

$$001.0000_2 = 1.0_{10}$$

$$010.0000_2 = 2.0_{10}$$

$$100.0000_2 = 4.0_{10}$$

- 小数点から右に進むと  $1/2$  倍

$$000.1000_2 = 0.5_{10}$$

$$000.0100_2 = 0.25_{10}$$

$$000.0010_2 = 0.125_{10}$$

$$000.0001_2 = 0.0625_{10}$$

## 固定小数点方式 2 進数 → 10 進数

桁の重みを合計する.

$$\begin{aligned} 10.01_2 &= 1 \times 2 + 0 \times 1 + 0 \times 1/2 + 1 \times 1/4 \\ &= 2 + 0 + 0 + 1/4 \\ &= 2 + 0.25 \\ &= 2.25 \end{aligned}$$

問題 1 2 : 2 進数を 10 進数に変換しなさい.

1)  $0101.1010_2$

2)  $0011.0011_2$

3)  $0100.0101_2$

4)  $1010.1111_2$

## 10進数 → 固定小数点方式 2進数

2進数	×2は	10進数
0.101 <sub>2</sub>	左シフトと同じ	0.625
✓		×
1.010 <sub>2</sub>		<u>1.250</u>

2進数	×2は	10進数
0.010 <sub>2</sub>	左シフトと同じ	0.250
✓		×
0.100 <sub>2</sub>		<u>0.500</u>

2進数	×2は	10進数
0.100 <sub>2</sub>	左シフトと同じ	0.500
✓		×
1.000 <sub>2</sub>		<u>1.000</u>

10進数で計算したとき、小数点を横切って整数部に出てきた数を小数点の右に順番に並べると 0.101<sub>2</sub> になる。

## 10進数 → 固定小数点方式 2進数

問題 1 3 : 10進数を 2進数に変換しなさい.  
但し 2進数は, 小数点以下 4 桁, 全体で 6 桁とする.

1)  $0.75_{10}$

2)  $0.5625_{10}$

3)  $2.5_{10}$

4)  $1.1875_{10}$

# 文字の表現 (ASCII コード)

ASCII コード (*American Standard Code for Information Interchange*)  
1963 年にアメリカ規格協会 (ANSI) が定めた情報交換用の文字コード.

(上位 3 ビット)

(下位 4 ビット)		0	1	2	3	4	5	6	7
	0	NUL	DLE	(SP)	0	@	P	`	p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	"	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	\$	4	D	T	d	t
	5	ENQ	NAK	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	'	7	G	W	g	w
	8	BS	CAN	(	8	H	X	h	x
	9	HT	EM	)	9	I	Y	i	y
	A	LF	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[	k	{
	C	FF	FS	,	<	L	\	l	
	D	CR	GS	-	=	M	]	m	}
	E	SO	RS	.	>	N	^	n	~
	F	SI	US	/	?	O	_	o	DEL

# 文字の表現 (JIS 8ビットコード)

JIS (Japan Industrial Standard : 日本工業規格) 8ビットコード

JIS 8ビットコードは、ASCIIコードに半角カタカナを追加したもの。記号、数字、英字の部分は、ほぼ、同じ並びになっている。

(上位4ビット)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE		0	@	P	`	p				一	タ	ミ		
1	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
2	STX	DC2	"	2	B	R	b	r			「	イ	ツ	メ		
3	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
4	EOT	DC4	\$	4	D	T	d	t			、	エ	ト	ヤ		
5	ENQ	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL	ETB	'	7	G	W	g	w			デ	キ	ヌ	ラ		
8	BS	CAN	(	8	H	X	h	x			イ	ク	ネ	リ		
9	HT	EM	)	9	I	Y	i	y			ウ	ケ	ノ	ル		
A	LF	SUB	*	:	J	Z	j	z			エ	コ	ハ	レ		
B	VT	ESC	+	;	K	[	k	{			オ	サ	セ	ロ		
C	FF	FS	,	<	L	¥	l				ヤ	シ	フ	ワ		
D	CR	GS	-	=	M	]	m	}			ユ	ス	ヘ	シ		
E	SO	RS	.	>	N	^	n	~			ヨ	セ	ホ	。		
F	SI	US	/	?	O	_	o	DEL			ヅ	ゾ	マ	。		

(下位4ビット)

はASCIIコード表と異なる部分

# 文字の表現 (Unicode)

*Unicode* : 最も用いられている業界標準規格  
世界中で使用される全ての文字を収録しようと現在も拡張が続いている.

1. 全ての文字に 21 ビットのコードを割り振る.  
最大で約 111 万文字にコードを割り振ることができる.  
現在, 約 15 万文字が登録されている.
2. 000000H~00007FH の範囲は ASCII コードと同じ
3. 000370H~0003FFH の範囲にギリシャ文字
4. その後に, キリル文字, アルメニア文字, ヘブライ文字...
5. 002E80H~009FFFH の範囲に漢字・かな・カナ・その他
6. 00F900H~00FAFFH の範囲は漢字
7. 01F600H~01F64FH の範囲に顔文字
8. 020000H~0323AFH の範囲も漢字

参考 : <https://ja.wikipedia.org/wiki/Unicode>



# 文字の表現 (UTF-8)

Unicode のエンコーディング (符号化) 方式の一つ

21 ビットの Unicode をそのままファイル等へ書き込むと、00H のバイトだらけになり容量がもったいないし、通信すると時間がかかる。

よく使う文字 (ASCII) を短く符号化する。 (8 ビット可変長)

1. 000000H～00007FH の範囲は 1 バイトに変換  
(00H～7FH, 第 1 バイトが 0 で始まる)
2. 000080H～0007FFH の範囲は 2 バイトに変換  
(C2H,80H～DFH,BFH, 第 1 バイトが 110 で始まる)
3. 000800H～00FFFFH の範囲は 3 バイトに変換  
(E0H,A0H,80H～EFH,BFH,BFH, 第 1 バイトが 1110 で始まる)
4. 010000H～10FFFFH の範囲は 4 バイトに変換  
(F0H,90H,80H,80H～F7H,BFH,BFH,BFH, 第 1 バイトが 11110 で始まる)

第 2 バイト以降は、10 で始まっている。

参考：<https://ja.wikipedia.org/wiki/UTF-8>

# 補助単位

1,000m を 1km, 1,000g を 1kg, 0.001l を 1ml, 0.001m を 1mm  
ここで, k や m は補助単位と呼ばれる.

一般的に			記憶容量		
値	記号	読み方	値	記号	読み方
$10^3$	<i>k</i>	キロ	$2^{10}$	<i>Ki</i>	キビ
$10^6$	<i>M</i>	メガ	$2^{20}$	<i>Mi</i>	メビ
$10^9$	<i>G</i>	ギガ	$2^{30}$	<i>Gi</i>	ギビ
$10^{12}$	<i>T</i>	テラ	$2^{40}$	<i>Ti</i>	テビ

- 通常は  $10^3$  毎に補助単位がある.
- コンピュータの記憶容量では  $2^{10}$  毎に補助単位がある.  
 $2^{10} = 1,024 = 1Ki$   
 $10^3 = 1,000 = 1k$