第 1 章
データの表現・テキスト処理

2024 年 9 月 19 日

学習目標

(1) データの表現方法について理解する．
(2) テキストの表現方法について理解する．
(3) 画像・音声・動画の表現方法について理解する．

本章は，専修大学商学部の高萩栄一郎の著作である．

1　データの表現方法

本節では，データ（数値，文字，画像等）の表現方法について学修します．コンピュータのデータは，デジタルデータと呼ばれ，0 と 1 の組み合わせで表現されます．1 桁の 0 または 1 をビット (bit), 通常，8 ビットのまとまりを 1 バイト (byte) といいます．1 ビットでは，0 か 1 の 2 種類の情報を表現でき，8 ビットでは，$2^8 = 256$ 種類の情報を表現できます．

情報入門 2 のホームページにある dexp.xlsx を利用します．オレンジのセルが，値を入力してみるセル，青のセルが皆さんで計算式を入力するセルになっています．

1.1　数値の表現方法

1.1.1　整数型の表現

整数は，2 進数で表現します．10 進数が，$0, 1, \ldots, 8, 9, 10, 11, \ldots$ のように，9 の次，10 になると繰り上がって 10(イチ ゼロ) となる数と同じように，2 進数は，$0, 1, 10, 11, 100, 101, 110, 111, 1000, \ldots$ のように，2 になると繰り上がる数です．

図 1 の (1) 整数型は，B4 に整数を入れると，C4 に 2 進数表現を文字列として表示するものです．Excel では，10 数数 (decimal) の数値を 2 進数 (binary) に変換する関数は，DEC2BIN です．

C4: =DEC2BIN(B4)

B4 に 0 から 10 くらいまで数値を 1 ずつ変化させ，2 進数表現の変化を確認しましょう．

負の数はどのようになるのでしょうか？ B4 に-1 を入れ見ましょう．「1111111111」と 1 が 10 桁表示された思います．これは，整数を 10 ビットで表現したとき，1 を加えると 0 になる数 (ただし，$1111111111 + 1 = 10000000000$ となるが，10 ビットとしたので，最初の 1 は保持されない（無視）) です．このような表現方法を 2 の補数表現と呼ばれます．

	A	B	C	D	E	F
1	(1) 整数型					
2	10進数から2進数					
3		整数	DEC2BIN			
4		9	1001			
5						
6	(2) 指数表現					
7		小数	指数表現			
8		28053	2.80530E+04			
9						
10		仮数	指数	小数		
11		2.8053	2	280.53		
12	※　コンピュータ内部は，10進数ではなく2進数					
13						
14	(3)　計算誤差					
15		整数1	整数2	整数1-整数2	10倍	10を引く
16		53	52	1	10	0
17						
18		小数1	小数2	小数1-小数2	10倍	1を引く
19		5.3	5.2	0.1	1	-3.55271E-15

図 1　数値の表現（dexp.xlsx のシート「数値」）

1.1.2　浮動小数点型の表現

図 1 の (2) 指数表現は，数値を指数表現で表現し直したものです．

- C8: =B8
- C8 を右クリックして，セルの書式設定
- 表示形式のタブで，分類を指数，小数点以下の桁数を 5 と設定します．

B8 に 280.53 と入力すると，「2.80530E+02」と表示されます．「E+02」の右は，10 の何乗倍かを表し，これは，$10^2 = 100$ 倍を表します．「2.80530E+02」は，$2.80530 * 10^2 = 280.530$ を表します．

このような表現をしたとき，2.8053 の部分を仮数，+02 を指数と呼びます．B11 に仮数，C11 に指数を入力したとき，D11 にその数値を表示させましょう．

D11: `=B11*10^C11`

このような表現をすることにより，大きな数（例えば，987654321012.99) や 0 に近い数 (0.0000000000123) などを表現できます（B8 に，大きな数や 0 に近い数を入力すると，表示形式が標準のため，自動的に指数形式で表示されます)．

小数はコンピュータ内部では，2 進数で同様な方法で表現されます．仮数とその符号，指数部の整数の 3 つの部分で表現されます．このような表現方法を浮動小数点型と呼ばれています．

1.1.3 計算誤差

2 進数では正確には，10 進数を表現できません．これは，10 進数で 1/3 を有限の桁の小数で表現できないのと同様の理由です．小数の計算を行うと誤差が生じることがあります．図 1 の (3) 計算誤差で，誤差を生じさせてみます．

整数 1 − 整数 2 = 1 となる数値を B16,C17 に入力し，整数 1 − 整数 2(D16) を 10 倍し (E16),10 を引きます (F16)．整数の場合，誤差なく 0 になります．

小数 1 − 小数 2 = 0.1 となる数値を B19,C19 に入力し，小数 1 − 小数 2(D19) を 10 倍し (E19),1 を引きます (F19)．整数の場合，誤差なく 0 になります．図のように小数 1 に 5.3, 小数 2 に 5.2 を入力したときは，-3.55271E-15 となり 0 に近いで

すが誤差が生じます．（入力した小数により，0 となることもあります）．

このように，小数の計算では誤差が生じることがあるので，会計などの事務計算では注意しましょう．

動画 (復習用教材):計算誤差（音声付）

動画 (復習用教材):計算誤差（音声なし）

1.2 文字の表現

文字には番号が振られており，その番号は文字コードと呼ばれています．英数字小文字は，7 ビットで表現される ASCII コードが使われます．漢字やハングル文字などは，全世界の文字を基本的に 16 ビット (2 バイト) の整数値で表現する ユニコード (Unicode) が使われることが多いです．実際の文字コード (符号化方式) は，UTF-8 と呼ばれる方式で表現されることが普及しています．

文字コードは，4 ビットで 1 桁を表す 16 進数 (hexadecimal) で表示します．ASCII 文字は 8 ビットに納まるので，16 進数 2 桁，日本語文字のユニコードの整数値は，2 バイトなので 16 進数 4 桁で表示します．

文字コード（数値，10 進数）を，16 進数の文字列に変換する関数は DEC2HEX で，16 進数の文字列を文字コードに変換する関数は HEX2DEC です．

図 2 の (1)ASCII コードは，文字コード (32〜126）を入力するとそのコードの文字を表示する関数 char を利用したものと ASCII 文字を入力するとその文字コードを表示する関数 code を利用したものです．

- C3: `=DEC2HEX(B3)`
- D3: `=CHAR(B3)`
- C4: `=CODE(B6)`
- D4: `=DEC2HEX(C6)`

	A	B	C	D	E
1	(1) ASCII コード				
2		文字コード(32～126)	文字コード(16進数)	文字	
3		65	41	A	
4					
5		文字(ASCII)	文字コード(10進数)	文字コード(16進数)	
6		A	65	41	
7					
8	(2) Unicode 文字				
9		日本語文字	Unicode(10進数)	Unicode(16進数)	
10		漢	28450	6F22	
11					
12		Unicode(16進数)	Unicode(10進数)	日本語文字	
13		6F22	28450	漢	
14					
15	(3) UTF-8				
16		日本語文字	URL encode	UTF-8	
17		漢	%E6%BC%A2	E6BCA2	
18					

図 2　数値の表現（dexp.xlsx のシート「文字」）

図 2 の (2)Unicode は，漢字などの文字を入力するとその Unicode の整数値を表示する関数 UNICODE 利用したものと Unicode の整数値を入力すると，その文字を表示する関数 =UNICHAR(C13) を利用したものです.

- C10: `=UNICODE(B10)`
- D10: `=DEC2HEX(C10)`

- C13: =HEX2DEC(B13)
- D13: =UNICHAR(C13)

いろいろな文字コードや文字で試してみましょう.

16進数	0	1	2	3	4	5	6	7
0				0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A			*	:	J	Z	j	z
B			+	;	K	[	k	{
C			,	<	L	¥	l	|
D			-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

図 3　ASCII コード表 (文字のみ)

図 2 の (3)UTF-8(参考) は，日本語文字を UTF-8 に変換するものです．UTF-8 で日本語文字は，16 進数 3 桁 (3 バイト) に変換されます.

図 3(ワークシート「文字コード表」) は，ASCII コードの文字コード（文字のみ）表です.

表の見方:「M」は，上方 (H1) の 4 が 16 進数の 1 桁目，左方 (B16) の D が 16 進数の 2 桁目になり，「M」は 16 進数で 4D となります.

1.3　画像・音・動画の表現

画像，音声，動画も，数値や文字と同様に，コンピュータ内部ではデジタル表現（2 進数での表現）で格納されています．それぞれどのように，表現されるのか説明します．

1.3.1　画像の表現

画像の表現は，大きく分けると 2 種類あります．

■ラスタ型　ラスタ型は，格子状のマス目を作り，各マス目をピクセル（画素）と呼び，各ピクセルの色番号で表現します．ピクセル数が多ければ，高画質になります．例えば，最近のスマートフォンのカメラのピクセル数は 4800 万ピクセルのものがあります (2023 年 8 月)．拡大していくと，ピクセルが見えるように，拡大の限界があります．また，ファイルサイズが大きくなる傾向にあります．

写真やスキャナーで取り込んだ画像などが当てはまります．ファイル形式では，JPEG, GIF, PNG などがあり，JPEG などでは，ファイルサイズを抑えるため，圧縮が行うことができます．

■ベクター型　ベクター型は，座標を使って，直線や円弧などの曲線や塗りつぶしなどで表現します．
次の囲みの中は，ベクター型の表現方法の 1 つである SVG(Scalable Vector Graphics) での円弧を描く表現です．

これは，「座標 (150,150) を中心に半径 100 の円を太さ 1 の黒線で描き，中を白で塗りつぶせ」という命令の SVG です．

```
<svg width="300" height="300">
  <circle cx="150" cy="150" r="100" stroke="black" stroke-width="1" fill="white" />
</svg>
```

ベクター型は，拡大に強く，拡大しても曲線が保たれます．また，ファイルサイズは比較的小さくなります．しかし，ラスター型のファイルをベクター形式に変換するのは，かなり困難な処理と言われています．

ベクター形式は，線や図形を組み合わせて作成されたイラストや設計図などで使われます．ファイル形式には，SVG, EPS, PDF などがあります．

■比較　maru.bmp は，ラスター形式のファイルで円を描いたものです．2000% くらいまで拡大すると，ピクセルの矩形が見えてきます．

maru_svg.html は，上記 SVG を html ファイルに組み込んだものです．拡大していっても，ピクセルの矩形が見えず，円弧の曲線が保たれているのが確認できます．

動画 (復習用教材):ラスタ型ベクター型比較（音声付）

動画 (復習用教材):ラスタ型ベクター型比較（音声なし）

1.3.2　音の表現

音は，空気の振動の波として表されるので，波の大きさを一定間隔で測定（サンプリング）して，その大きさをデジタル表現します．1 秒間に何回，サンプリングをするのかをサンプリング周波数と呼んでいます．たとえば，44.1kHz（CD のサンプリング周波数）は，1 秒間に 44100 回サンプリングをすることを意味します．

1.3.3　動画の表現

動画は，パラパラ漫画のように画像（フレーム）を短い間隔で切り換えて表示することにより表現しています．1 秒間に何回，フレームを切り換えるのかをフレームレートと呼ばれています．

1.3.4　圧縮

ラスタ型の画像や音，動画は容量が大きくなります．そのままですと，保存や通信に不向きになります．そのとき，圧縮する技術を使って容量を小さくして保存，通信をします．圧縮には，可逆圧縮と非可逆圧縮の 2 種類あり，可逆圧縮は圧縮前のデータに戻せる圧縮方法で，非可逆圧縮は圧縮前のデータに戻せない圧縮方法で，一般に非可逆圧縮のほうが容量は小さくなります．

1.4　構造化データ，非構造化データ

1.4.1　構造化データ

構造化データは，決められた構造で表現されたデータです．例として，表 1 のような売り上げデータが挙げられます．

表 1　構造化データの例 (売り上げデータ)

売り上げ NO	日付	顧客 NO	商品 NO	個数
A001	2023/09/10	C987654	SU6128	7
A002	2023/09/11	C123456	SU6128	20
A002	2023/09/11	C123456	RT7588	3

日付の列には，日付が入り，個数には，非負の整数が入るなどの決まり・制約があり，それに従って，データが入れられています．このように表現すことにより，検索・抽出（たとえば，顧客番号 C123456 の一覧）や計算・集計 (たとえば，SU6128 の個数の合計) をしたりすることが容易に行えます．構造化データは，表計算ソフトウエアでも管理できますが，より大規模な場合は，リレーショナル（関係）データベース (RDB) として管理されます．

1.4.2　非構造化データ

非構造化データは，表の形で表現することが困難データで，文書（テキスト）データ，画像データ，音声データなどがあります．分析などのデータ利用が，構造化データと比べて困難なことが多いです．しかし，さまざまな統計，数理，AI などを利用した分析・解析ツールが登場しています．

非構造化データの分析例として，第 5 章のアンケート分析で自由記述欄（テキスト）の分析を体験します．

1.5　半構造化データ

構造化データのように決まった構造ではなく，柔軟な構造を与えたものです．SVG の例で，タグ (`<circle>`）は，円を描くことを示しています．SVG では円以外にも線，矩形，文字などさまざまなタグで図形をえがくことができ，順番や個数など柔軟に表現できます．

また，web のページは，html で表現されており，右クリックして，ページのソースを表示 で HTML 表現を見ることができます．さまざなタグ，`<HTML>`や`<li>`，`<script>`などは，表示情報の種類を表していますが，テキスト同様に，順番や回数は自由です．

このように，半構造化データは，構造化データと非構造化データの中間の性質を持っています．