■ 画像ファイルのフォーマット

画像ファイルをペイントで作成した場合、そのファイルには .bmp という拡張子が付けられる。拡張子とファイルの関係については以前の実習で説明したが、これはファイルの中身がどのようなものか、また、どのようなソフトウェアによって作られたかを示している。

ひとくちに画像ファイルと言っても、そのフォーマットは幾つもあり、拡張子も異なったものが付けられる。以下に簡単な説明をつけておく。

拡張子	形式の名前	特徴
jpg	JPEG	写真など多くの色を扱う画像を、小さなファイルサイズで自然に見えるように圧縮できる。規格が公開されており、多くのソフトウェアで扱える。 もちろんより強く圧縮すると見た目が汚くなる傾向にある。大きな絵（画素数の多い画像）を圧縮するのには向くが、小さな絵（アイコンなどの画素数の少ない画像）を扱うと却ってファイルサイズが大きくなる。
bmp	ビットマップ	Microsoft 社が開発したフォーマット。ペイントなど、Windows に標準的に付随してくるソフトウェアが対応している。 圧縮されていないようで、ファイルサイズは大きい。ブラウザによっては表示できない場合がある。
gif	GIF	Compuserv 社が開発したフォーマット。小さなサイズの画像を効率よく圧縮するために作られ、規格が公開されていたため、初期の WWW 普及時に利用が広まり、今に至る。imode 電話機で標準的に採用されている画像フォーマットでもある。 写真のような色数が多く、画素数の多い画像を扱うとファイルサイズが大きくなる傾向がある。UNISYS パテント問題(*1)から、徐々に対応ソフトが減りつつある。
png	PNG	UNISYS パテント問題に対応するために作成された規格。機能的にはほぼ GIF に等しく、それより優れている部分もある。利用が啓蒙されているが、まだ対応ソフトウェアも少なく、普及は今ひとつ。しかし今後は積極的に利用するべきであろう。

*1 UNISYS パテント問題
GIF が利用している圧縮機構には UNISYS 社がパテントをもっている部分があり、UNISYS 社がライセンスしていないソフトウェアが作った GIF 画像を WWW に利用すると、数十万円のライセンス料を請求すると宣言されている(http://www.unisys.com/unisys/lzw/lzw-license_j.asp など)。有名なところでは Adobe 社等はライセンス料を払っており、これらのソフトウェアを利用して GIF ファイルを作成しなければならない。

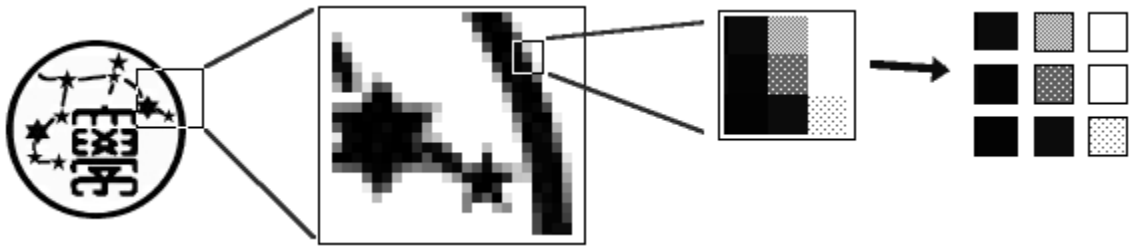
□デジタルデータの表現方法

一般にデジタルデータにはフォーマット（書式）が存在する。書式とは「どのように書かれているか」を意味し、つまりそのデータが、どのようにして（どのようなルールで）データ化されたかがわかっていることがデジタルデータ利用の必須要件である。
画像データにも既に示したように多様なフォーマットがあり、フォーマットがわからないと表示したりすることはできない。
Netscape Communicator はビットマップ形式のデータ化ルールを知らないため、ビットマップ形式の画像を画面上に表示することが出来ない。Internet Explorer はルールを知っているため、それができる。

以下に、何故デジタルデータにとってデータ化ルール（データの書かれ方＝フォーマット）が重要なのかを説明する。

□画像のデータ化

デジタルな表現とは、数値記録を意味する。つまり絵なら絵を、音なら音を、すべて数値で表現しなおしたもので記録され、処理されている。
例として下記のような画像を部分的に数値化してみる。



現在の実習で扱っているような種類のデジタル画像を拡大していけば、ある種の点の集まりとして認識できるだろう。最終的に最も小さな構成要素となる、このような点のことを「画素」と呼ぶ。（この用語はもともと英語で呼ばれていた Pixel（Picture Element の略、ピクセル）の直訳と思われる。）

上の画像はおおよそ 90 x 90 画素の画像だったが、その一部分を取り出し、3 x 3 画素部分を数値化（しなおす）することにする。手順としてはこのようになる。

1. ■▓▒▒□ の 5 種類の色（濃さ）の画素が存在するため、これに 1,2,3,4,5 の番号を振る。
2. 左上から右、次に真ん中の行の左から右、というように色番号を数えることにする。
3. 結果、1-3-5, 1-2-5, 1-1-4 がこの 3 x 3 画素のデジタル表現となる。

90 x 90 の画像全体に存在する色の種類が 40 段階だったとしても、この数え方でデータ化可能であることがわかるだろう。

□ 数え方の違い＝フォーマットの違い

ところで、上の数え方とはまた異なる数え方が出来ることがわかるだろう。

・■▓▒▒□に対する番号づけを薄い順から行い、5,4,3,2,1 とする。
・左上からではなく右上から数える、または縦方向（下）に数えていく。

このように、同じ対象物でも、数え方（ルール）を違えると、まったく違う数字列になることがわかる。つまり異なるデータ化ルールで表現された画像を表示させると、ぐちゃぐちゃの絵が表示されたりする。（実際にはそれ以前に「これはおかしい」とソフトウェア自身がチェックして「表示できない」とエラーメッセージを代わりに出す。）

つまりデータ化のルールは非常に重要であり、データとデータ化ルールはセットでなければならない。こうしたデータ化のことをコード化（符号化）、エンコーディング（encoding）とも言う。逆にコード化されたデータから元の画像に復元することをデコード（decode）と呼ぶ。

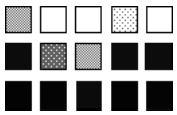
フォーマットの違いはエンコードルールの違いであり、どのようなルールを用いているかを知らない限り、デコードすることはできない。

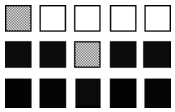
ファイルに付けられた拡張子は、エンコードルールを示している。 

など、それぞれ拡張子に合わせて違う、妥当なデコード処理が行われて表示される。

□ 圧縮

データを MO などに記録する場合、その数値情報（コード）の量に応じて MO 上の領域を占領していく。同じ画像を記録するのなら、各ファイルサイズが小さく、一枚の MO などに多くの画像が記録できるほうが便利であるため、ファイルサイズが小さくなる工夫をこらす場合がある。サイズを縮めるために行われる工夫なので一般に圧縮と呼ばれる。

簡単な圧縮のアイディアを示す。例えば右のような画素をエンコードすると、

3-5-5-4-5, 1-2-3-1-1, 1-1-1-1-1
となるが、最下行が 1 の連続を「1 の 5 連続」として表現すると、
3-5-5-4-5, 1-2-3-1-1, 1x5
とでき、データ量が少なくなっていることがわかる。背景が真っ白な画像などを対象にすれば効果は大きい。デコードの際には「1x5」のルールが判っておれば「1-1-1-1-1」に戻して表現できるため、圧縮する前と全く同じ画像が再現できる。（可逆圧縮と呼ぶ）

今度は、どうせ 5 段階の色のうち、2,4 の中間色は目立たないから、共に黒と白にして、右のように 3 段階に変更した場合を考える。

3-5x4, 1-1-3-1-1-1, 1x5
とさらに圧縮できた。この場合、見た目はよく似ているだろうが、圧縮前とは厳密には異なる画像しか再現できない。（不可逆圧縮と呼ぶ） JPEG は不可逆圧縮であり、圧縮率を高めると品質が落ちる（見た目が汚くなる）のはこのためである。

■文字コード

文字も絵と同様にデジタルデータとして扱われており、そこにはコード化ルールがある。
例えばアルファベットの A から Z までを 1-26、a から z を 27-52、数字の 0 から 9 を 53-63 というように数値化した場合、
12Monkeys
は、54,55,13,41,40,37,31,51,45 と表現できる。
漢字も同様に数千文字にそれぞれ番号が割り振られている。

ところが絵と同様に、漢字を含めた文字にはコード化ルールが複数ある。日本でよく使われている文字コード体系には以下の三つがある。

・ISO-2022-JP （いわゆる JIS コード）
・日本語版 EUC（単に EUC とだけ呼ばれる場合もあるが他の言語版 EUC が幾つかある）
・Shift-JIS（SJIS とも呼ばれる）

テキストエディタ を利用して HTML ファイルを書いている場合、保存するときにどの漢字コードで保存するか指定することが出来る。どれでも構わないが講師は EUC または JIS を多く利用している。Windows は標準的には Shift-JIS を利用しているので、例えばメモ帳などを利用して作成した HTML ファイルは Shift-JIS コードである。

絵と同様、例えば EUC-JP で書かれた HTML ファイルを JIS コードだと思って表示するなど、コード化ルールを間違えて再生すると、右図のように読めない文字で表示されたりする。（いわゆる文字化け）

この場合は、Internet Explorer なら「表示」メニューの「エンコード」から、Netscape Communicator なら「表示メニュー」の「文字コードセット」から、適当と思えるエンコード方式を選択してやることで正しく表示できる。

逆に HTML ファイルを作成する側でも、どの漢字コードを利用しているかを明示できるため、これで文字化けをある程度防ぐことが出来る。具体的には下記のような<META>タグを、冒頭、<HEAD> と </HEAD>の間に記入する。

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;charset=ISO-2022-JP">
<TITLE>実験ページ</TITLE>
</HEAD>
<BODY>
……（以下略）
```

上は HTML ファイルの作成時に指定した漢字コードが JIS コードの場合で、これが EUC であった場合は charset=EUC-JP 、S-JIS の場合は charset=Shift_JIS と書く。

文字コード	charset の記述
JIS	ISO-2022-JP
EUC	EUC-JP
S-JIS	Shift_JIS

□ 参考：ASCII コード

実際に一般的に利用されている ASCII と呼ばれるコード化ルールでのアルファベット、数字、記号などの番号づけは以下のようなものである。

A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73
J 74	K 75	L 76	M 77	N 78	O 79	P 80	Q 81	R 82
S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	
a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105
j 106	k 107	l 108	m 109	n 110	o 111	p 112	q 113	r 114
s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	
0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56
9 57								
(空白) 32								
+ 43	- 45	* 42	/ 47					
(40	) 41	< 60	> 62	[91	] 93	{ 123	} 125	
: 58	; 59	! 33	@ 64	# 35	36	% 37	& 38	. 46
, 44	? 63							