

入学年度	学部	学科	組	番号	検	フリガナ	
						氏名	

QRコードで漢字をコード化する仕組みを詳しく知るために、自分の名前をQRコードにしてみよう。

- **型**：日本人の大抵の名前は漢字7文字以内なので、誤り訂正レベルが高めの「レベルQ」を選択しても1型に収まる。漢字4文字ならさらに訂正能力の高い1-H型を用いることが可能である。今年度は履修者の名前はすべて漢字4文字なので、1-H型を用いることにする。
- **モード指示子**：今回はQRコードで用いられる種々のモードのうち「漢字モード」を用いる。これを指示する最初の4bitは1000である。
- **文字数指示子**：つぎに文字数指示子は、名前の字数を2進法で表示する。漢字モードの場合、これを8bitで表す(英数字モードの場合は9bit)。例えば「草野美登莉」のように5文字なら00000101となる。
- **データのbit列化**：そして、いよいよ実際の漢字データをbit列になおす。欧文に使われる文字は様々なアクセント記号を含めても $2^8 = 256$ 以内に収まるので、1byte=8bitで表現できるが、漢字などの非欧文文字は2byteを用いないと表せない。このように様々な文字を2byteの数で表現する仕組みのことを「文字コード体系」と呼ぶ。QRコードでは漢字を表す際には日本独自の体系である“Shift_JISコード”と呼ばれる文字コードが用いられる。

Shift_JISコードの概略

0と1のみを扱うコンピュータ内部では、数は10進法ではなく2進法で表現した方が都合がよい。しかし、プログラミングの際に数を2進法で表すと桁数が大きくなりすぎて人間にとっては理解しづらくなってしまふ。そこで、プログラミングなどの際には16進法を用いて数を表すことが一般的である。1byte = $2^8 = 16^2$ なので、16進法で2桁までの数を1byteで表すことができ、2byteで16進法で4桁までの数を表現できる。

一方、2byteで表せる文字数は $2^{16} = 65,535$ 字であるが、常用漢字は約2,000字程度であり、さらに使用頻度の低いJIS第4水準漢字までをすべて含めても約11,000字程度であって、かなりの空き領域ができる。Shift_JISコードはJIS第2水準漢字までの漢字など6,879文字が含まれるが、これは $2^{13} = 8,192$ より少ないので、すべて表すには2byte = 16bitも必要はなく、13bitあれば十分である。そこで、QRコードでは無駄をなくすために、漢字1文字を16bitではなく13bitの2進数で表示することにより、情報を圧縮して格納している。

Mathematicaで“草”のShift_JISコードを求めるには“**ToCharacterCode**”というコマンドを用いるが、**ToCharacterCode["草","ShiftJIS"]**とすると、結果が10進法で表示されるので、16進法で表示するために次のようにする。

```
BaseForm[ToCharacterCode["草","ShiftJIS"],16] (Shift_Enter)
{9116, 9016}
```

Shift_JISコードの漢字は第1byteを縦軸に、第2byteを横軸にとった仮定の正方形の中で、2つの長方形の領域内に含まれている。第1の長方形は第1byteが 81_{16} から $9F_{16}$ 、第2byteが 40_{16} から FC_{16} 、第2の長方形は第1byteが $E0_{16}$ から EB_{16} 、第2byteが 40_{16} から FC_{16} の長方形である。

	第2byte				
第1byte	40_{16}	41_{16}	...	FC_{16}	
81_{16}		、	...		○
⋮			⋮		
$9F_{16}$	槩	藪	...		滌
			⋮		
$E0_{16}$	漾	瀉	...		琰
⋮			⋮		
EB_{16}	*	*	...		*

圧縮の原理は以下の通り。第1byteは 81_{16} を0番とし、 $9F_{16}$ までを連番で表すと、 $9F_{16} - 81_{16} = 1E_{16} = 30$ なので、 $9F_{16}$ は30番となる。漢字コードが $E040_{16}$ を超える場合、 $E0_{16}$ から $C1_{16}$ を引くと、 $E0_{16} - C1_{16} = 1F_{16} = 31$ となるので、 $E0_{16}$ が31番、 $E1_{16}$ が32番、...、 EB_{16} が42番となる。一方、第2byteは 40_{16} を0番とすると、 $FC_{16} - 40_{16} = BC_{16} = 188$ より、 FC_{16} は188番となる。そこで、漢字コードを次のようにして「192進法化」する。すなわち、

$$(\text{第1byteの通し番号}(0\sim 43)) \times 192 + (\text{第2byteの通し番号}(0\sim 188))$$

を漢字コードの番号とする。この計算はすべて16進法のまま行う。例えば、“草”なら“ 9190_{16} ”から“ 8140_{16} ”を引くと“ 1050_{16} ”となるが、この最初の1byteに $C0_{16} = 192$ を掛け、つぎのbyteを足す。すなわち、

$$10_{16} \times C0_{16} + 50_{16} = 0C50_{16}$$

こうして“草”は次のように変換される。

$$\begin{aligned} \text{“草”} &\Rightarrow 9190_{16} \\ &\Rightarrow 9190_{16} - 8140_{16} = 1050_{16} \\ &\Rightarrow 10_{16} \times C0_{16} + 50_{16} = 0C50_{16} \\ &\Rightarrow 0110001010000_2 \end{aligned}$$

では、“草”、“野”、“美”、“登”、“莉”を「192進法化」してみよう。まず、漢字を対応するShift_JIS漢字コードに直と、それぞれ、“ 9190_{16} ”、“ $96EC_{16}$ ”、“ $94FC_{16}$ ”、“ $936F_{16}$ ”、“ $E4BB_{16}$ ”となる。

漢字 4 文字の場合には下の表のようになる.

氏名	Shift_JIS コード	差	「192 進法化」	13bit 化
草	9190 ₁₆ - 8140 ₁₆ = 1050 ₁₆		0C50 ₁₆	0110001010000
野	96EC ₁₆ - 8140 ₁₆ = 15AC ₁₆		106C ₁₆	1000001101100
美	94FC ₁₆ - 8140 ₁₆ = 13BC ₁₆		0EFC ₁₆	0111011111100
登	936F ₁₆ - 8140 ₁₆ = 122F ₁₆		0DAF ₁₆	0110110101111
莉	E4BB ₁₆ - C140 ₁₆ = 237B ₁₆		1ABB ₁₆	1101010111011

これを自分の名前でやってみよう.

氏名	Shift_JIS コード	差	「192 進法化」	13bit 化

すべてを bit 化したら, モード指示子, 文字数, 漢字データを順に並べ, 最後に終端パターンとして 0000 を付加する.

$$\underbrace{1000}_{\text{漢字モード}} \underbrace{00000100}_{\text{文字数 4}} \underbrace{0110001010000}_{\text{“草”}} \underbrace{1000001101100}_{\text{“野”}} \dots \underbrace{0000}_{\text{終端パターン}}$$

こうして得られたデータを 8bit ごと (1byte ごと) に区切り直す. 最後のビット列が 8bit 未満の場合は 0 で埋める. また, 1-Q 型では RS(26, 13, 6) 符号を用いるので, 得られた byte 数が情報 byte 数である 13 に満たない場合は "11101100" および "00010001" という「埋め草パターン」を交互に付加する.

漢字モード	文字数	(第一文字)	(第二文字)
1	0 0 0 0 0 0 0 0	0 1 0 0	
		(第三文字)	(第四文字)
終端パターン	0 fill	埋め草パターン 1	埋め草パターン 2
0 0 0 0	0 0 0 0	1 1 1 0 1 1 0 0	0 0 0 1 0 0 0 1
			1 1 1 0 1 1 0 0
埋め草パターン 2			
0 0 0 1 0 0 0 1			

漢字 5 文字の場合には裏の表のようになる.

漢字モード	文字数	(第一文字)	(第二文字)
1	0 0 0 0 0 0 0 0	0 1 0 1	
		(第三文字)	(第四文字)
(第五文字)	終端パターン	0 fill	埋め草パターン 1
	0 0 0 0	0 0 0 0 0 0 0 0	1 1 1 0 1 1 0 0
埋め草パターン 2			
0 0 0 1 0 0 0 1			