數位資料表示法

到底數位是什麼呢？

- 數位在電學上是指不連續變化的數量表示法。
- 何謂不連續變化？
  - 實數是連續變化的數量表示法，因為任兩數之間還可以找到第三個數介於它們之間，而且到最後是沒有空隙的。
  - 整數是不連續變化的數量表示法，例如整數1和整數2之間，我們再也找不到任何整數是介於它們之間的。

到底數位是什麼呢？

- 針對不連續變化的數量，可以用位元(binary digit；bit)的組合來計數。
- 位元是數位資訊的基本粒子，也是電腦儲存或傳遞資料的最小單位，常用0或1來表示。
- 電腦會採用位元表示資料，主要是因為電子元件的穩定狀態有兩種，單一的0或1稱為位元(bit)。
  - 「開」(通常用來表示“1”)
  - 「關」(通常用來表示“0”)

到底數位是什麼呢？

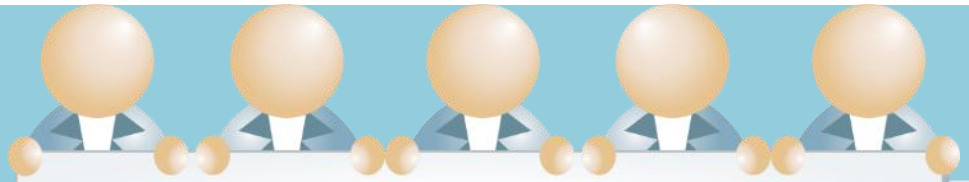
- 早期電腦以8個位元為存取單位，因此8個位元稱為位元組(byte)。
- 兩個位元可以有 2的2次方 共4種組合(00, 01, 10, 11)。
- 每增加一個位元，組合數就加倍。
- n個位元可以有 2^n 種不同的組合，就可用來表示 2^n 種不同物件。

到底數位是什麼呢？

- 8個位元可以有 2的8次方 共256種組合，足以表示每一個英文字母(大小寫共52個)、數字(0到9共10個)和標點符號。ASCII就是這類型組合的公定標準。
- 16個位元可以有 2的16次方 共65,536種組合，遠超過常用的中文字數目，因此16個位元可表示中文字。

到底數位是什麼呢？

- 為避免各國文字的位元表示方式有所衝突，萬國碼(Unicode)依實現方式不同，而以不同位元個數的組合來公定各國文字。
- 由於電腦的存取機制以位元組為基本單位，所以表示資料所需的位元數，通常是8、16及32等。



TIPS!

關於資料容量的單位，常見的有KB、MB、GB及TB四種。

「B」代表的是Byte(位元組)，不是Bit(位元)。

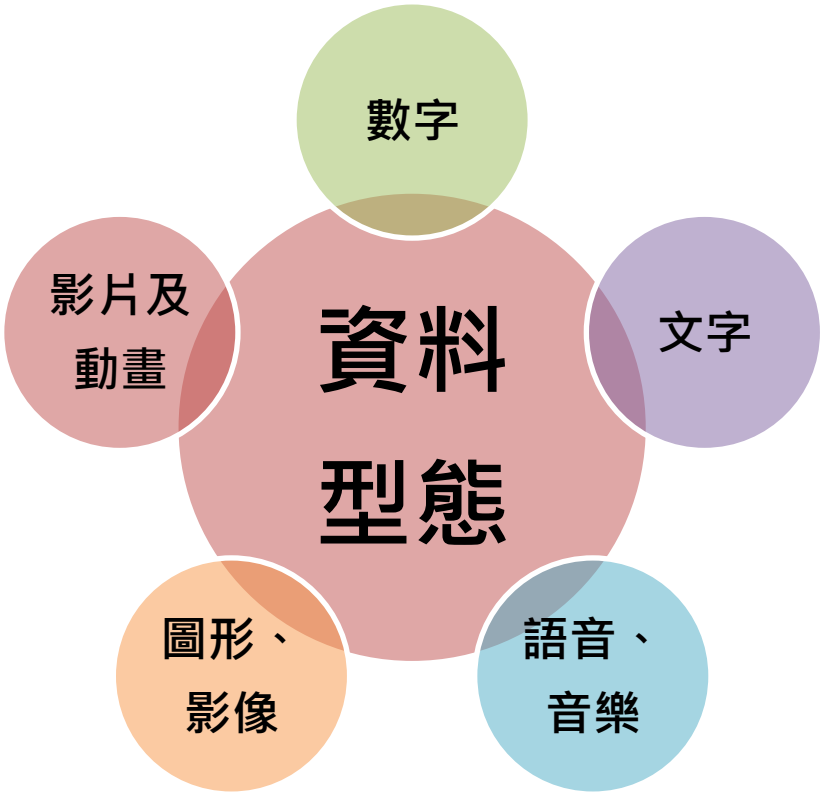
「K」代表了2^{10}，為1,024，大約是一千左右。

「M」是$2^{20} = 2^{10} \times 2^{10} = 1{,}048{,}576$，大約是百萬左右。

對於2^x的估算，我們常以2^{10}為簡化的捷徑，因為它和10^3(也就是1000)非常接近。

2-1 資料型態

- 電腦需要處理的**資料型態**(data type)有：數字、文字、語音、音樂、圖形、影像、影片及動畫等，會編碼成位元字串儲存在電腦裡，等到顯示或列印時，再解碼成原來的資料格式。



2-1 資料型態

- 影像數位化
  - 以黑白照片為例，照片的一小部分記錄每個方格的灰度(0~255)，每個方格可用八位元來表示(八個0與1可以有256種組合)。可依同樣道理將彩色圖片數位化。
- 聲音數位化
  - CD唱片上的取樣是每秒約四萬四千次，每一次取樣的聲波，都可轉化成相對應的位元。

2-1 資料型態

- 數位化的資訊方便編輯、處理、儲存、傳輸及播放，以便更有效精確地表達意念。
- 可用電腦編輯及整合不同的數位化資訊，精確安排各種複雜媒體出現的順序、時間及播放設備。
- 可利用電腦強大的處理及搜尋功能，提供多媒體的互動方式，加強虛擬實境的真實感。

2-2 二進位表示法

- 古巴比倫人所用的數字系統是六十進位法，逢「六十」進一，現在除了每分鐘六十秒及每小時六十分外，此法已不多見。
- 現今公制是以十為基數，採用十進位法，滿「十」進一。

2-2 二進位表示法

- 一個數字在不同的位置上所表示的數值也就不同。

523		
5 在百位上則表示5個百	2 在十位上就表示2個十	3 在個位上表示3個一

- $523 = 5\times10^2 + 2\times10^1 + 3$。

2-2 二進位表示法

- 電腦電子元件最穩定簡單的狀態為「開(1)」與「關(0)」，故目前通行電腦用二進位符號來儲存資料。
- 因為一個位元組有八個位元，可切成兩個十六進位數，因此電腦系統也常使用十六進位數來顯示資料。
- 十六進位系統的數字0到15，分別以阿拉伯數字的0~9及A~F表示。
- 二位元字串 11010011 可表示成 $D3_{16}$ 或 0xD3 (x起頭，代表該數為十六進位數)。

2-3 各種進位表示法的轉換

十六進位的數字符號及其所對應的十進位及二進位

十進位	二進位	十六進位	十進位	二進位	十六進位
0	0	0	8	1000	8
1	1	1	9	1001	9
2	10	2	10	1010	A
3	11	3	11	1011	B
4	100	4	12	1100	C
5	101	5	13	1101	D
6	110	6	14	1110	E
7	111	7	15	1111	F

範例

10110101.1101_2所對應的十進位數為181.8125

1	0	1	1	0	1	0	1	.	1	1	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}

$$2^7 + 2^5 + 2^4 + 2^2 + 2^0 + 2^{-1} + 2^{-2} + 2^{-4}$$

$$= 128 + 32 + 16 + 4 + 1 + 0.5 + 0.25 + 0.0625$$

$$= 181.8125$$

範例

十進位181所對應的二進位數為10110101_2

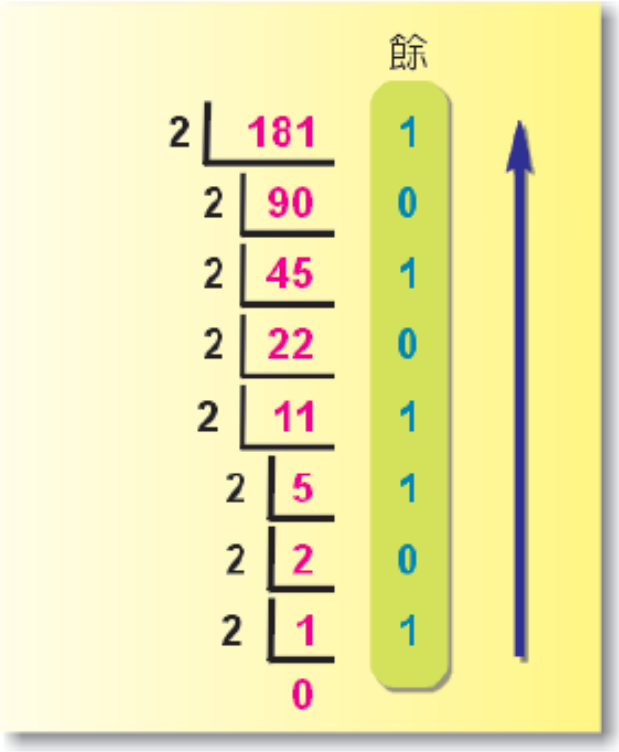
181÷2得商數90，餘數1

→d_1為 1

90÷2得商數45，餘數0

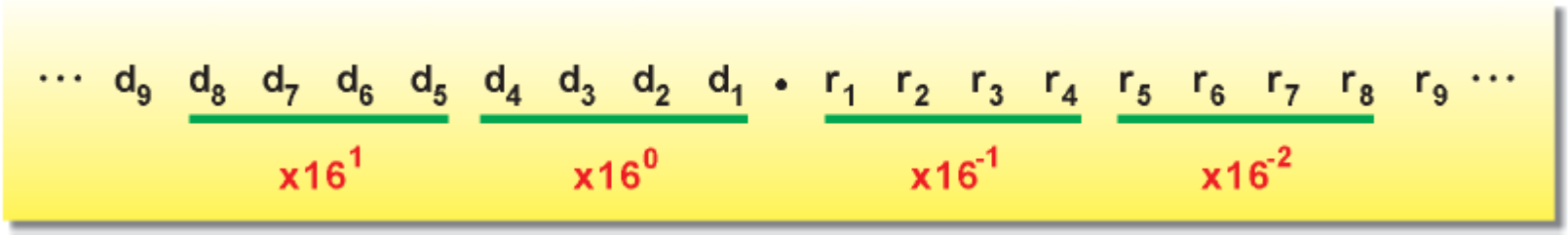
→d_2為 0

...以此類推。



二進位數與十六進位數的互換

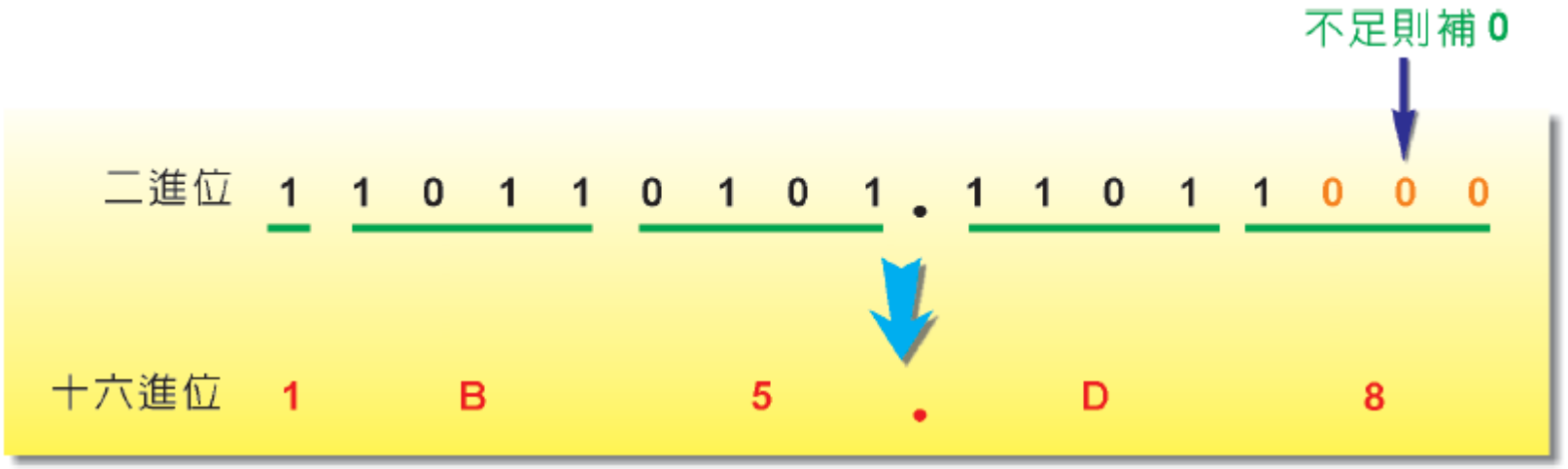
- 因為16為2的整數次方，所以二進位數和十六進位數可說是系出同門。
- 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- 那十進制呢?? Ans: $23_{10}=15_{16}$



二進位數換成十六進位數時，每四個位數合成一項

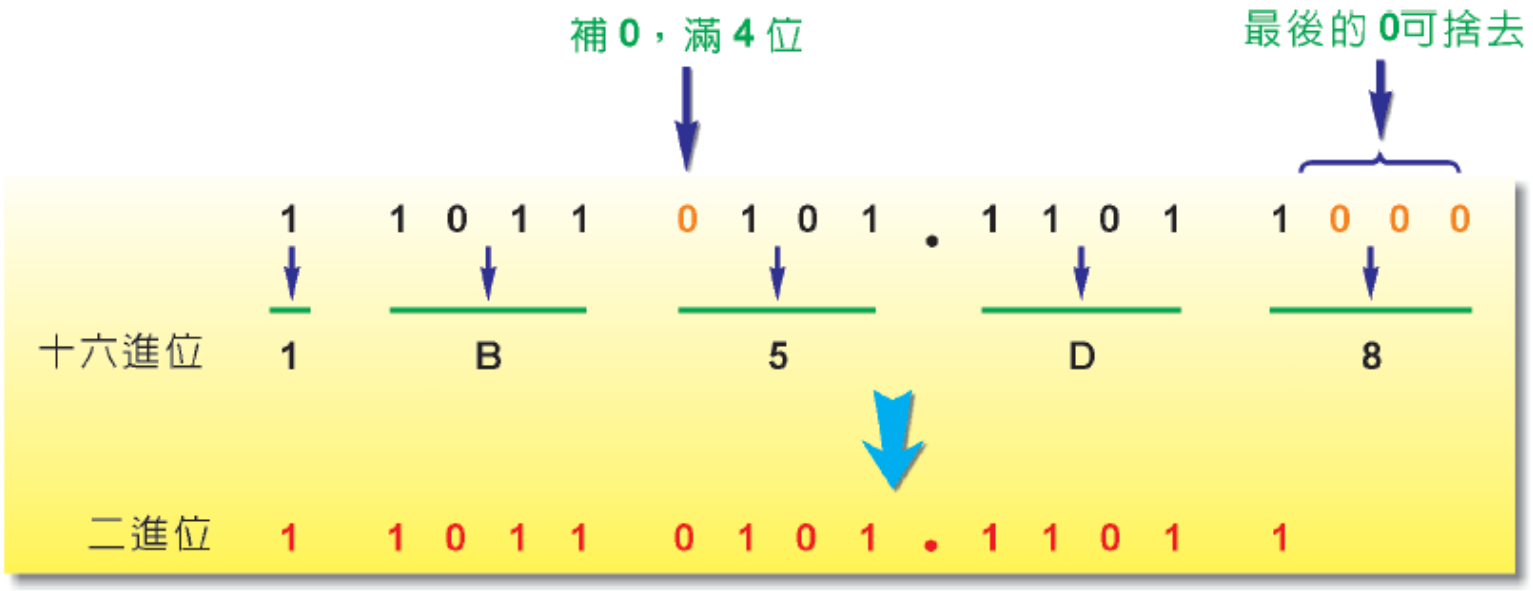
範例

110110101.11011_2 的十六進位表示法為 $1B5.D8_{16}$



範例

$1B5.D8_{16}$ 的二進位表示法為 110110101.11011_2



2-4 整數表示法

- 只表示非負的整數，只要將最小的位元字串(亦即全為0的字串)給0，依序表示到最大的數即可。
- n 個位元就可表示 2^n 個數，所表示的整數範圍為 $0 \sim 2^n-1$。
- 例如：使用8個位元，可表示$0 \sim 2^8-1$間的所有整數，也就是從0~255的所有整數。

無正負符號的整數

- 位元字串與十進位數的對應表

以8位元所表示的「無正負符號的整數」

位元字串	十進位數
00000000	0
00000001	1
00000010	2
⋮	⋮
11111110	254
11111111	255

帶正負符號大小表示法

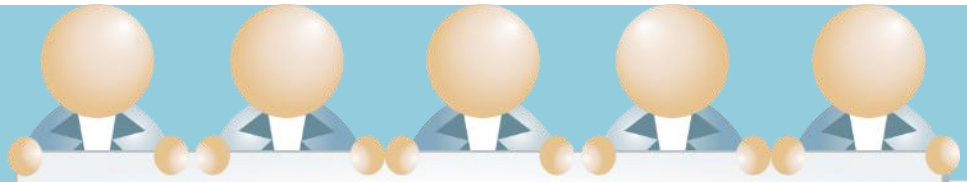
- 若要同時表示正數和負數，最直接的作法是採用「帶正負符號大小表示法」。
- 位元字串的最左邊位元當作符號位元(0為正數；1為負數)，剩下的n-1個位元用來表示數的大小。
  - 以位元0開頭的整數範圍為 $0 \sim 2^{n-1}-1$
  - 以位元1開頭的整數範圍為 $0 \sim -(2^{n-1}-1)$

帶正負符號大小表示法

- 若使用8個位元，則可表示 $-(2^7-1) \sim 2^7-1$ 間的所有整數(-127 ~ 127)。
- 此法的潛在問題：
  - 有兩個0，+0(000...00)和-0(100...00)。
  - 正數和負數的運算(例如加和減)並不直接。

以8位元所表示的「帶正負符號大小表示法」

位元字串	十進位數
00000000	0
00000001	1
⋮	⋮
01111111	127
10000000	-0
10000001	-1
⋮	⋮
11111111	-127



TIPS!

1. 次方與位值
   a. 位值(例：0.1、1、10、100、...)可用 10 的次方來表示(例：10^{-1}、100、101、102、...)。
   b. 以十進位表示法表示的數字也可用 10 的次方的形式來表示。
   例：$2346.531 = 2\times10^3+3\times10^2+4\times10^1+6\times10^0+5\times10^{-1}+3\times10^{-2}+1\times10^{-3}$ 。

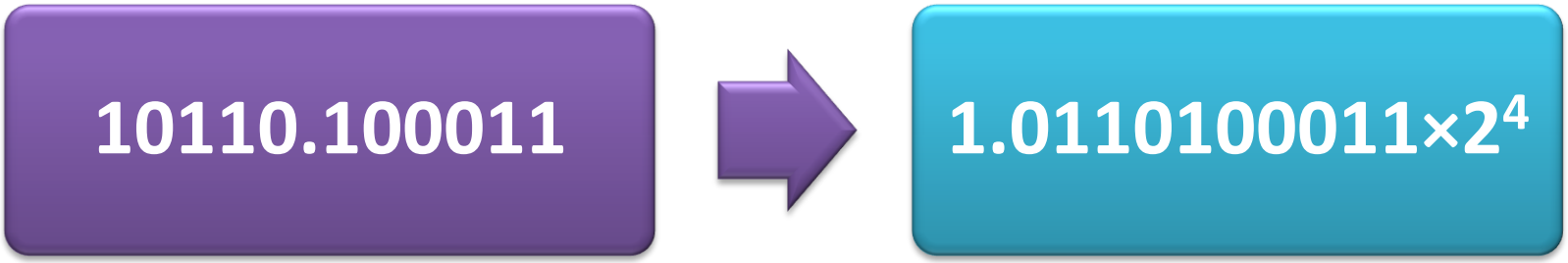
2. 科學記號表示法
   a. 把一個正數寫成 $a\times10^n$ 的形式，其中 a 大於或等於 1， 但是小於 10，且 n 為整數， 則 $a\times10^n$ 就是這個數的科學記號表示法。
   例：$6600000000 = 6.6\times10^9$，$0.000000071 = 7.1\times10^{-8}$ 。
   b. 要比較以科學記號表示的數時，要先把 10 的指數部分化為相同的數字，再比較前面所乘數字的大小。例：若 $a = 1.23\times10^{-6}$，$b = 23.1\times10^{-7}$，則 $a = 1.23\times10^{-6}$，$b = 2.31\times10^{-6}$，因為 $2.31 > 1.23$，所以 $b > a$ 。

2-5 浮點數表示法

- 浮點數表示法是電腦表示實數最常用的方式。
- 「536.87」表示成科學記號為「5.3687×10^2」，浮點數表示法的運作原理亦同，會移動小數點，使其「浮動」到標準的位置。
- 在有限位元數的情況下，浮動小數點所能表示的數值範圍比固定小數點位置的方式大許多。

2-5 浮點數表示法

- 科學記號標準化動作：



  - 小數點左邊的數值一定是1。
  - 小數點右邊的0110100011稱為**尾數**(mantissa)，而**指數**(exponent)為4。

2-5 浮點數表示法

- 目前所採用的浮點數表示法以IEEE 754標準為主，主要有三部分：

2-5 浮點數表示法

- **單倍精準數**：以1個位元表示符號；8個位元表示指數；23個位元表示尾數部分。
- **雙倍精準數**：以1個位元表示符號；11個位元表示指數；52個位元表示尾數部分。

單倍精準數所能表示的數字範圍

- 最小正數為00000000100000000000000000000000，其數值為 $+2^{-126}$
- 最大正數為01111111011111111111111111111111，其數值為 $(2-2^{-23})\times 2^{127}$。
- 最大負數為10000000100000000000000000000000，其數值為 -2^{-126}
- 最小負數為11111111011111111111111111111111，其數值為 $-(2-2^{-23})\times 2^{127}$。

2-6 ASCII及Unicode

- 美國國家標準局在1963年時發表的ASCII(唸成Asskey；美國國家資訊交換標準碼)是當今最普及的公定標準。
- 標準ASCII以7個位元儲存一字符，共有$2^7=128$種組合。電腦的儲存常用的位元組為8個位元，多出來的位元用來儲存錯誤檢驗位元(parity bit)。
- 擴充型ASCII用8個位元儲存一字符，有$2^8=256$種組合，可儲存非英文符號、圖形符號及數學符號等。

代碼解釋：Dec：10進制 Hx：16進制 Oct：8進制 Char：字元

16進制表示法：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F、10 ...

Ctrl	Dec	Hex	Char	Code
^@	0	00		NUL
^A	1	01		SOH
^B	2	02		STX
^C	3	03		ETX
^D	4	04		EOT
^E	5	05		ENQ
^F	6	06		ACK
^G	7	07		BEL
^H	8	08		BS
^I	9	09		HT
^J	10	0A		LF
^K	11	0B		VT
^L	12	0C		FF
^M	13	0D		CR
^N	14	0E		SO
^O	15	0F		SI
^P	16	10		DLE
^Q	17	11		DC1
^R	18	12		DC2
^S	19	13		DC3
^T	20	14		DC4
^U	21	15		NAK
^V	22	16		SYN
^W	23	17		ETB
^X	24	18		CAN
^Y	25	19		EM
^Z	26	1A		SUB
^[	27	1B		ESC
^\	28	1C		FS
^]	29	1D		GS
^^	30	1E	▲	RS
^-	31	1F	▼	US

Dec	Hex	Char
32	20	
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Char
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	⌂*

擴充字元集(Extended Character Set)

128	Ç	144	É	160	á	176	░	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	|	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	Θ	249	·
137	ë	153	Ö	169	_	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	▉	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	_
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	ϕ	253	²
141	ì	158	_	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

Unicode

- 美國萬國碼制訂委員會於1988-1991年間訂定的Unicode (萬國碼) 字符編碼標準，已成為ISO認證之標準(ISO10646)。
- Unicode發展出下列多種編碼方式：
  - UTF-8 在全球資訊網最通行。
  - UTF-16 為JAVA及Windows所採用。
  - UTF-32 則為一些UNIX系統使用。

Unicode

- Unicode前面128個符號為ASCII字符，其餘則為英、中、日、韓文以及其他非英語系國家之常用文字。
- Unicode中最大宗的分類是CJK，主要是中文、日文及韓文之漢字集。

在http://www.unicode.org/charts/ 網址裡，提供了各種不同類別字符的對照表。

Unicode符號對照表

範　圍	代表的字符群
0000-007F	基本拉丁字符（與ASCII相同）
0080-024F	擴充的拉丁字符
0370-03FF	希臘字符
0E00-0E7F	泰文
0E80-0EFF	寮文
2200-22FF	數學符號
2500-25FF	方塊圖形及幾何圖形
3040-30FF	平假名及片假名
4000-9FFF	CJK；中文、日文及韓文之漢字

EBCDIC

- 除了ASCII和Unicode外，IBM的EBCDIC也是某些機型上常用的編碼方式。
- 國際標準局(ISO)用四個位元組(也就是32位元)制定一種編碼方式，可以有2^{32}種組合，可表示多達4,294,967,296種字符。

Big5 / GB

- 以正體字而言，大五碼(Big5；約一萬六千字)是廣受歡迎的一種編碼方式，盛行於台灣及香港。
- 以簡體字而言，國標(GB；約八千字)是廣受歡迎的編碼方式，盛行於大陸地區。
- 這些字體已逐步被包含於Unicode的CJK字集中，未來的整合一致化指日可待。

Homework

- 請寫出 2^4 的所有變化
- 請寫出11110101_2，01101010_2 的十進位
- 請寫出85，152的二進位數值
- 請寫出19_{16}，95_{16}的十進位數值
- 1010110111_2，95213的十六進位數值
- 請寫出8547的十進位科學記號表示法
- 請寫出ASCII代碼中ThankYou! 的十六進位數值


Thank You
For Your Atten

單倍精準數

- 符號位元：1個位元，以0表示正數；以1表示負數。
- 指數部分：8個位元，以過剩127(Excess 127：將位元數值減去127所得的值，才是真正所儲存的值)方式表示。8個位元所存的數值可從0～255，共有2^8種變化。
- 尾數部分：23個位元，從標準化的小數點後開始存起，不夠的位元部分補0。

範例

01000010100101000110000000000000 所儲存的數值為多少？

第一步

位元符號為0，所以是正數，指數部分是10000101＝十進位133，再減去127，得6。

第二步

01000010100101000110000000000000所儲存的數值為1.0010100011×2^{6}，也就是1001010.0011。

範例

10000010100101000110000000000000 所儲存的數值為多少？

第一步

位元符號為1，所以是負數，指數部分是00000101＝十進位5，再減去127，得-122。

第二步

10000010100101000110000000000000所儲存的數值為$-1.0010100011\times2^{-122}$。