Digital Information

INFO/CSE 100, Autumn 2004
Fluency in Information Technology

http://www.cs.washington.edu/100

Readings and References

- Reading
  - » *Fluency with Information Technology*
    - Chapter 11, Representing Multimedia Digitally
- Wikipedia - The Free Encyclopedia
  - » Arabic numerals, ASCII
    - http://en.wikipedia.org/wiki/Arabic_numerals
    - http://en.wikipedia.org/wiki/Ascii
- Cyrillic Text
    - http://www.dimka.com/ru/cyrillic/

Recall: What do number positions represent?

			2×2×2×2	2×2×2	2×2	2	1	
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	base 10
1	0	0	0	1	0	1	0	base 2

$$1 \cdot 128 + 1 \cdot 8 + 1 \cdot 2 = 138_{10}$$

Each position represents one more multiplication by the base value.

For binary numbers, the base value is 2, so each new column represents a multiplication by 2.

Use the base, Luke

- Each position represents one more multiplication by the base value
  - » The base value can be **2** - *binary numbers*
    - Two symbols: 0 and 1
    - Each column represents a multiplication by two
  - » The base value can be **10** - *decimal numbers*
    - Ten symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    - Each column represents a multiplication by ten

10×10×10	10×10	10	1	
$10^3 = 1000$	$10^2 = 100$	$10^1 = 10$	$10^0 = 1$	base 10
0	1	3	8	base10

$$1 \cdot 100 + 3 \cdot 10 + 8 \cdot 1 = 138_{10}$$

text figure imageis from Wikipedia

Base 16 Hexadecimal

- The base value can be **16** - *hexadecimal numbers*
  - » Sixteen symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - » Each column represents a multiplication by sixteen
  - » Hex is easier to use than binary because the numbers are shorter even though *they represent the same value*

16×16×16 $16^3 = \mathbf{4096}$	16×16 $16^2 = \mathbf{256}$	16 $16^1 = \mathbf{16}$	1 $16^0 = \mathbf{1}$	base 10
0	0	8	A	base 16

$$8 \cdot 16 + 10 \cdot 1 = 138_{10}$$

Four binary bits ⇔ One hex digit

binary base 2		hexdecimal base 16		decimal base 10
0 0 0 0	⇔	0	⇔	0
0 0 0 1	⇔	1	⇔	1
0 0 1 0	⇔	2	⇔	2
0 0 1 1	⇔	3	⇔	3
0 1 0 0	⇔	4	⇔	4
0 1 0 1	⇔	5	⇔	5
0 1 1 0	⇔	6	⇔	6
0 1 1 1	⇔	7	⇔	7

binary base 2		hexdecimal base 16		decimal base 10
1 0 0 0	⇔	8	⇔	8
1 0 0 1	⇔	9	⇔	9
1 0 1 0	⇔	A	⇔	10
1 0 1 1	⇔	B	⇔	11
1 1 0 0	⇔	C	⇔	12
1 1 0 1	⇔	D	⇔	13
1 1 1 0	⇔	E	⇔	14
1 1 1 1	⇔	F	⇔	15

Binary to Hex examples

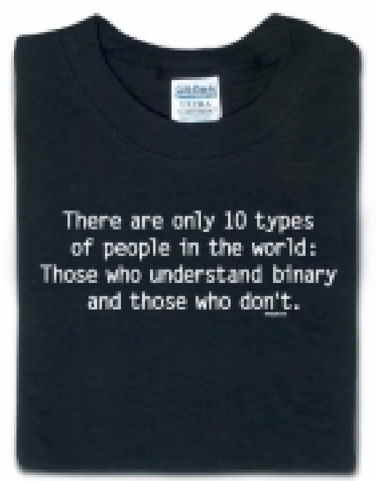
1000	0010	0000	0111	1010	0001	0000	1111	base 2
8	2	0	7	A	1	0	F	base 16

$$10000010000000111101000010000111 1_2 = 8207A10F_{16}$$

1000	0011	0100	0101	0110	1001	1011	1110	base 2

$$10000011010001010110100110111110_2 = \underline{\qquad\qquad}_{16}$$

Whew! We are now official geeks ...



ThinkGeek
stuff for smart masses

http://www.thinkgeek.com/tshirts/frustrations/5aa9/

Recall: The hardware is binary

- How many numbers can we represent with 0 and 1?
  - » As many as we want, it just takes a little more space to get a bigger range
- So what can we represent with these numbers?
  - » Anything that has a numeric value or can be associated with a numeric value
  - » Number of people, index into a list, account balance, ...
  - » Alphabetic characters, punctuation marks, display tags
  - » Any signal that can be converted into numeric values
    - colors, sounds, water level, blood pressure, temperature
  - » Computer instructions

Represent numbers

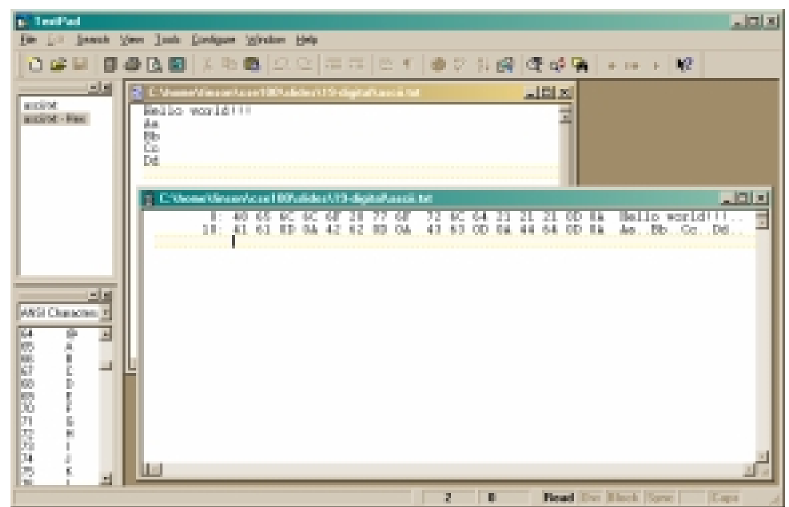
- How many bit positions to allocate?
  - » Depends on the desired range
  - » 8 bits → 0 to 255
    - or -128 to +127
  - » 16 bits → 0 to 65535
    - or -32768 to +32767
  - » 32 bits → 0 to 4294967296
    - or -2B to +2B

Represent Text - ASCII

- Assign a unique number to each character
  - » 7-bit ASCII
    - Range is 0 to 127 giving 128 possible values
    - There are 95 printable characters
    - There are 33 control codes like tab and carriage return

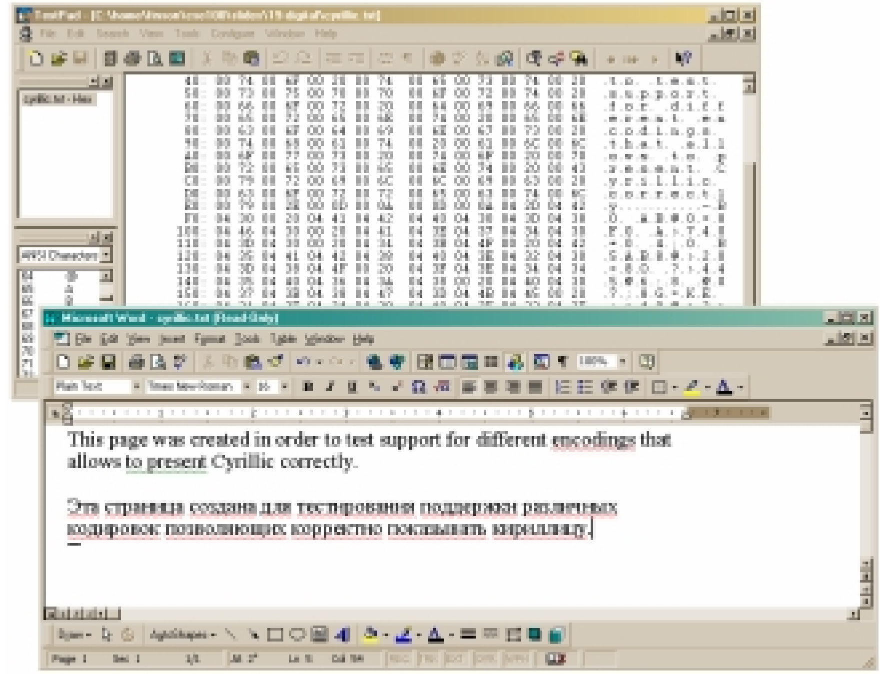
imageis from Wikipedia

ASCII text



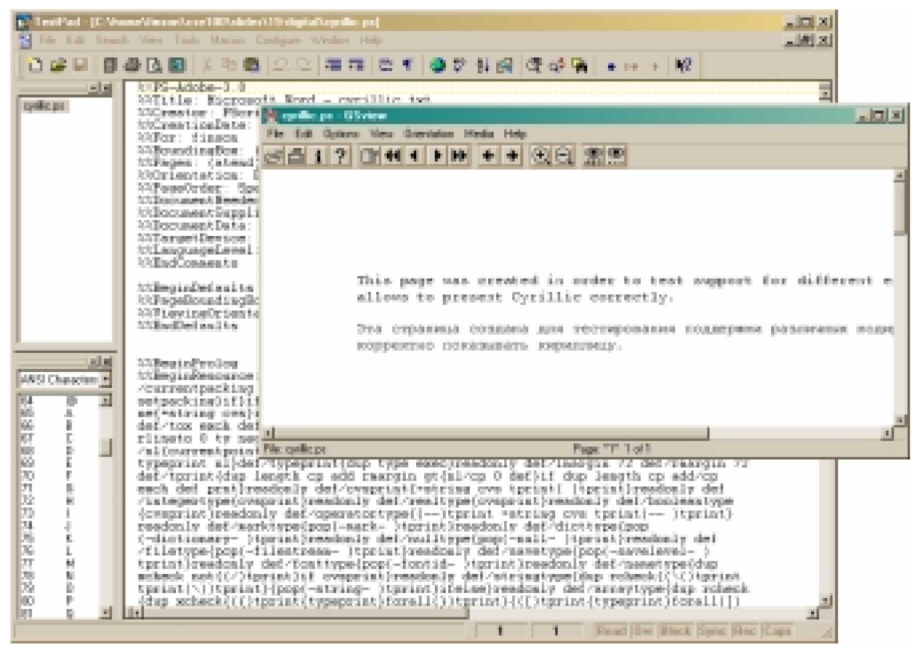
Represent Text - Unicode

- The goal of Unicode is to provide the means to encode the text of every document people want to store in computers
- Unicode aims to provide a unique number for each letter, without regard to typographic variations used by printers
- Unicode encodes each character in a number
  - » the number can be 7, 8, 16, or 32 bits long
  - » 16-bit encoding is common today



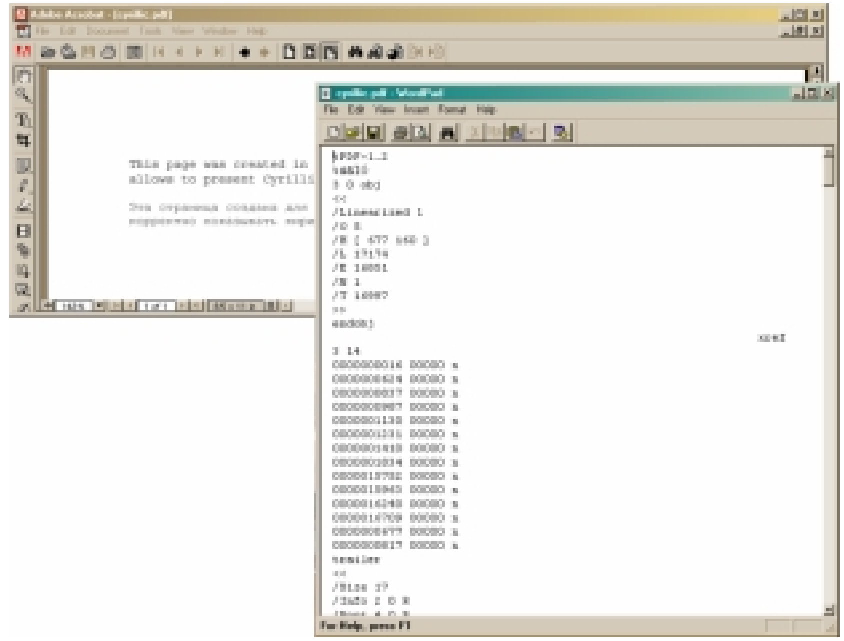
Represent Text - Postscript

- Postscript is a page description language somewhat like HTML
  - » The file is mostly text and can be looked at with a regular text editor
  - » programs that know what it is can interpret the embedded commands
  - » Programs *and printers* that understand Postscript format can display complex text and graphical images in a standard fashion



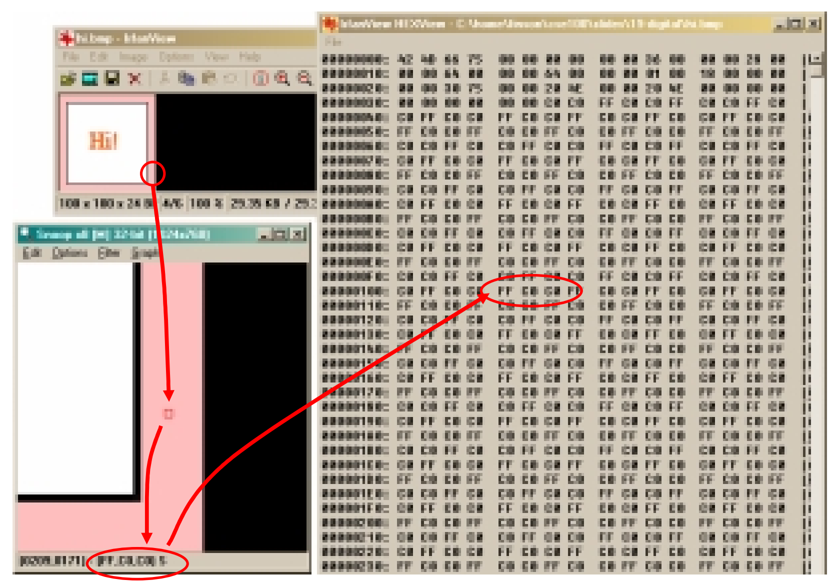
Represent Text - PDF

- PDF is another page description language based on Postscript
- The file is mostly text
  - » can be looked at with a regular text editor
  - » programs that know what it is can interpret the embedded commands
  - » just like Postscript and HTML in that respect



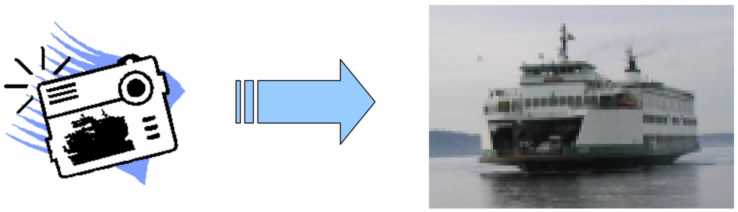
Represent Color - Bit Map

- Numbers can represent anything we want
- Recall that we can represent colors with three values
  - » Red, Green, Blue brightness values
- There are *numerous* formats for image files
  - » All of them store some sort of numeric representation of the brightness of each color at each pixel of the image
  - » commonly use 0 to 255 range (or 0 to FF_{16})

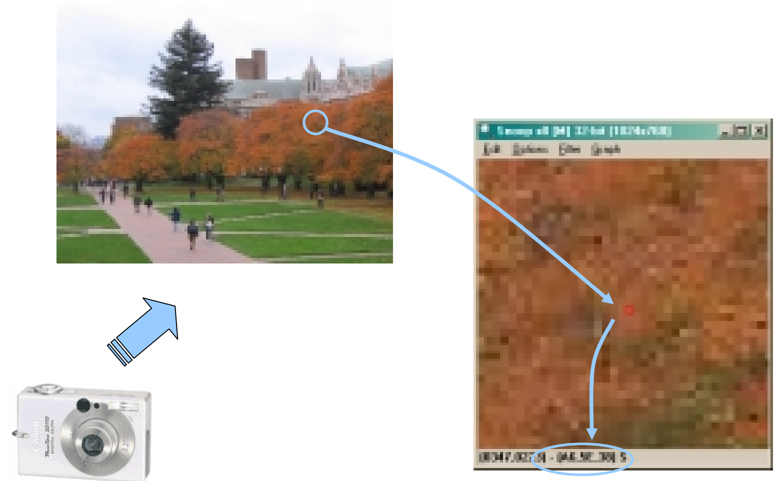


What about "continuous" signals?

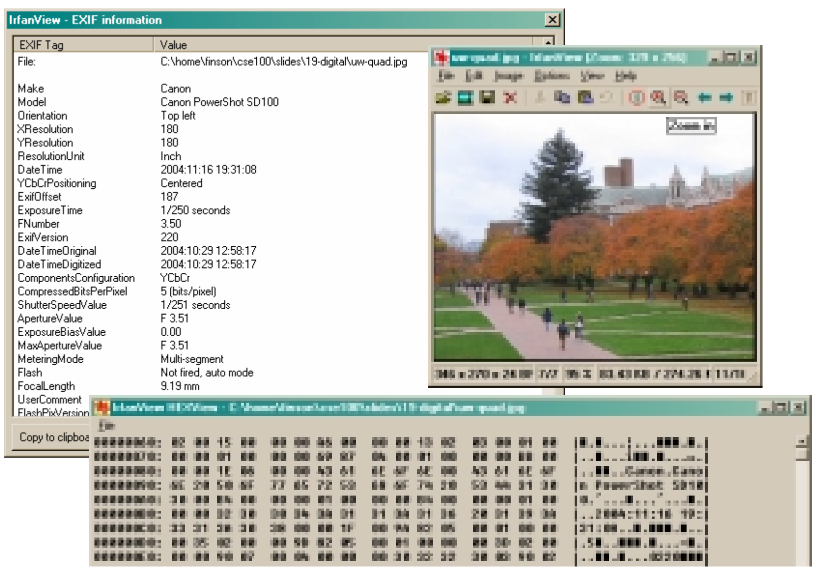
- Color and sound are natural quantities that don't come in nice discrete numeric quantities
- But we can “make it so!”



Digitized image contains color data



And much, much more!



Summary

- Bits can represent any information
  - » Discrete information is directly encoded using binary
  - » Continuous information is made discrete
- We can look at the bits in different ways
  - » The format guides us in how to interpret it
  - » Different interpretations let us work with the data in different ways