

# Bits and Bytes

At the smallest scale in the computer, information is stored as bits and bytes. In this section, we'll look at how that works.

## Bit

- Bit -- like an atom -- the smallest unit of storage
- A bit stores just a 0 or 1
- "Computer is all 0's and 1's" ... bits
- Anything with two separate states can store 1 bit --Nick's tennis racket example
- Chip uses electricity 0/1 states
- Hard drive uses spots North/South magnetism 0/1 states
- Bit too small to be much use
- Group 8 bits into a byte

Everything in a computer is 0's and 1's ... what does that mean? The **bit** stores just a 0 or 1 .. it's the smallest building block of storage.

## Byte

- One byte = grouping of 8 bits
- e.g. 0 1 0 1 1 0 1 0
- One byte can store one letter, e.g. 'b' or 'x'

## How much exactly can one byte hold?

- How many distinct/separate patterns can be made with 1, 2, or 3 bits?
- (Not hold you responsible for details, go through derivation)

Number of bits	Distinct Patterns
1	0 1
2	00 01 10 11
3	000 001 010 011 100 101 110 111

- 3 Bits
- Consider just the leftmost bit
- It can only be 0 or 1
- Lefmost bit is 0, append 2-bit patterns
- Leftmost bit is 1, append 2-bit patterns again
- Result ... 3-bits has twice as many patterns as 2-bits

Number of bits	Distinct Patterns
1	0 1
2	00 01 10 11
3	000 001 010 011 100 101 110 111

- In general: add 1 bit, double the number of patterns
- 1 bit - 2 patterns
- 2 bits - 4
- 3 bits - 8
- 4 bits - 16
- 5 bits - 32
- 6 bits - 64
- 7 bits - 128
- 8 bits - 256
- Mathematically: n bits yields 2<sup>n</sup> patterns (2 to the nth power)

## One Byte - 256 Patterns

- Need to know:
- 1 byte is group of 8 bits
- 8 bits can make 256 different patterns
- How to use the 256 patterns?
- How to store a number in a byte?
- Start with 0, go up, one pattern per number, until run out of patterns (256)
- 0, 1, 2, 3, 4, 5, ... 254, 255
- One byte holds a number 0..255
- Code: pixel.setRed(n) took a number 0..255. Why?
- Red value stored in **one byte**