

Decimal System

Digit Indices
Digit Value

3 2 1 0
1 9 8 7

$$1 \cdot (10^3) + 9 \cdot (10^2) + 8 \cdot (10^1) + 7 \cdot (10^0)$$
$$1000 + 900 + 80 + 7$$

Binary System

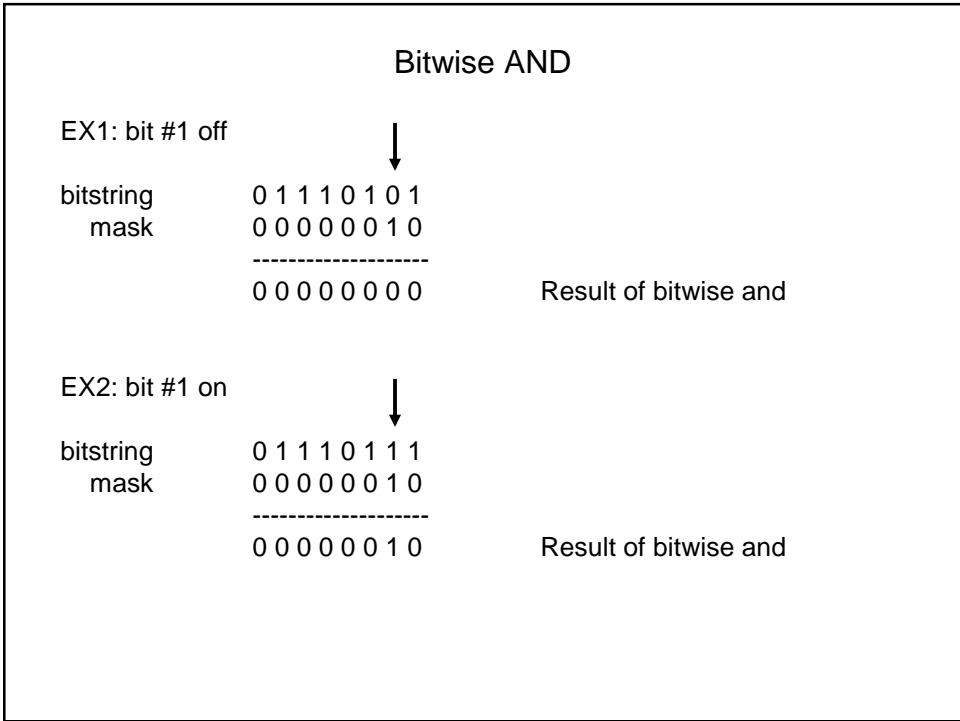
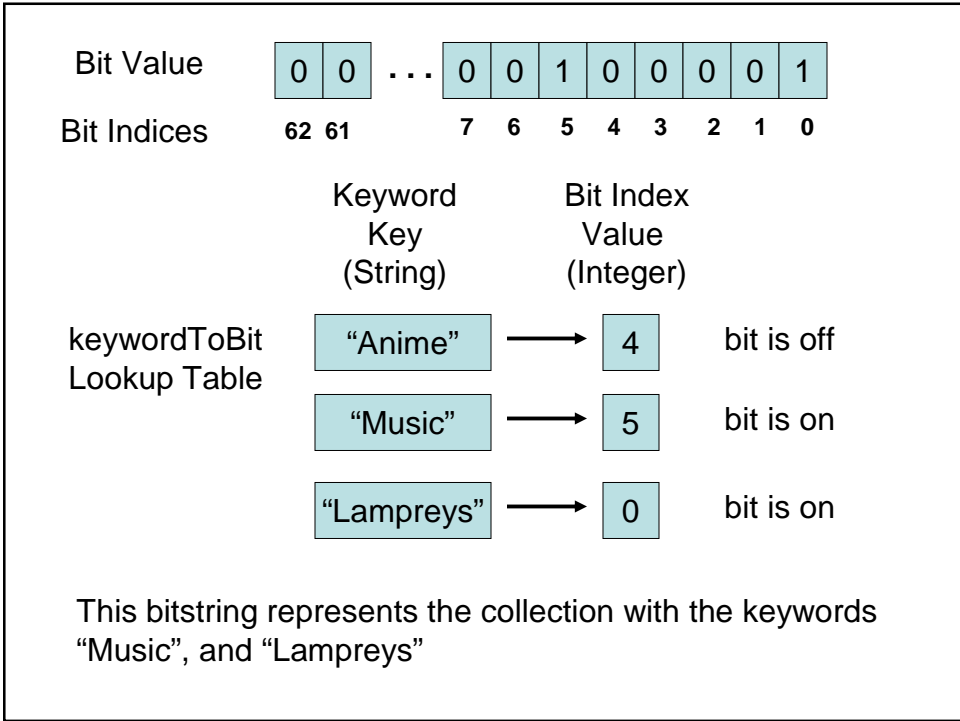
Binary 000...10001 = Decimal 33

Bit Value 0 0 ... 0 0 1 0 0 0 0 1
Bit Indices 62 61 ... 7 6 5 4 3 2 1 0

$$2^5 = 32 + 2^0 = 1$$

Bit 5 on Bit 1 on

Java long type is 64 bits: 1 for sign (+/-), we will use 63 bits, bits 62-0 inclusive.



Java Code

```
int bitIndex = 1;
int bitMask = Math.pow( 2, bitIndex );      // 2^1 = 2

// or you can use bit shifting
// int bitMask = 1 << bitIndex;           // The shift operator

int ex1Bitstring = 117;    // 0 1 1 1 0 1 0 1 in binary
int ex2Bitstring = 119;    // 0 1 1 1 0 1 1 1 in binary

if(( ex1Bitstring & mask ) == 0 ) {
    System.out.println( "ex1 has bit off" );
} else {
    System.out.println( "ex1 has bit on" );
}
if(( ex2Bitstring & mask ) == 0 ) {
    System.out.println( "ex2 has bit off" );
} else {
    System.out.println( "ex2 has bit on" );
}
```

You can have many Bitstrings (one for each KeywordCollection),
but only one keywordToBit lookup table.

(If you implement a bitToKeyword table, then only have one as well.)

