The digital circuits used in computers are fundamentally **binary**( base-2) in nature. Only the digits 1 and 0 are found in binary numbers **.** Computer data at the lowest level consists of binary digits, or **bits**, with a value of **1 (ON)** or **0 (OFF)**. Binary data representations are difficult for humans and they take up too many columns on a printed page. Number systems using other bases save space.

A set of 8 bits is called a **byte**. A coding system called the American Standard Code for Information Interchange (**ASCII**) uses bytes to enable humans to better understand computer data. In fact, ASCII is the "**Rosetta Stone**" that translates between computers and humans. In extended ASCII, each 8-bit byte is assigned a character or letter of the alphabet. We humans put characters together to make words and sentences. Inside a computer, these are just a series of 1s and 0s. Google "ASCII" and you will find code charts that display the ASCII code characters and their equivalent values in binary and **decimal** (base-10) numbers. The best ASCII charts also display **hexadecimal** (base-16) numeric values for each ASCII character.

Computer professionals should understand **both** the binary and hexadecimal number systems because they turn up in many parts of computing. Skill with binary numbers is a must for understanding ASCII codes and **IP addresses** on the Internet. Hexadecimal numbers are used to specify **colors** on HTML pages and the hardware addresses on Network Interface Cards (**NICs**). Although binary and hexadecimal numbers can be perplexing at first, they can be mastered with study and practice.

The 8 bits of a byte are numbered from 7 on the left down to 0 on the right. **Think of each bit as its power of the base 2**. **Add up the powers of the ON bits and you can convert binary to decimal**. For Byte 1 below, bits 6 and 0 are ON. That's 64 + 1 = **65** as a decimal number. In ASCII, this is upper case **A**.

| Bit # | 7 | **6** | 5 | 4 | 3 | 2 | 1 | **0** | Base-10 | ASCII Code |
|---|---|---|---|---|---|---|---|---|---|---|
| Power of 2 | 128 | **64** | 32 | 16 | 8 | 4 | 2 | **1** | | |
| Example Byte 1 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | **1** | **65** | **A** |
| Example Byte 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 66 | B |
| Example Byte 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 97 | a |
| Example Byte 4 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 100 | d |

On a PC, you can verify that in **NotePad** or **Word** by pressing **Alt + 6 + 5** (using the **Numeric keypad**). Try it. You will see an **A** appear. Try **Alt + 9 + 7** and you will get **a**.

So when you type **A** into a document, you set a byte to **01000001** in computer memory.