BinaryGround.com
ENJOY LEARNING BINARY & HEXADECIMAL

INTRODUCING BINARY AND HEXADECIMAL NUMBERS

YOU ALREADY KNOW THE DECIMAL SYSTEM

Let's start with the numeral system that we all know: **Decimal**. The decimal system uses all the **digits from 0 to 9**. The lowest and most right digit position in any decimal number has a value of 1. The next digit position to the left has a value of 10, the next 100 and so on, always increasing by a factor of 10.

Example

Number (decimal): **372**

Value (decimal): **3** x **100** + **7** x **10** + **2** x **1** = **300** + **70** + **2** = **372**

BINARY NUMBERS

In **Binary**, only the **digits 0 and 1** are used. There is no digit „2" in binary, so you write „10" instead. If you count in binary, it goes „0, 1, 10, 11, 100". The right most digit position again has the value 1, the next digit position to the left has a value of 2, the next 4 and so on, always increasing by a factor of 2.

Example

Number (binary): **1 0 1 1 0 1**

Value (decimal): **1** x **32** + **0** x **16** + **1** x **8** + **1** x **4** + **0** x **2** + **1** x **1** = **32** + **8** + **4** + **1** = **45**

HEXADECIMAL NUMBERS

Hexadecimal numbers use even more digits than 0-9. They include **digits with values up to 15**. Since there are no „normal" digits beyond 9, it is common to use the **letters A..F as digits with values of 10 .. 15**. The second position from the right has a value of 16, the third 256 and so on, always increasing by a factor of 16.

Example

Number (hexadecimal): **F7**

Value (decimal): (F=15) **15** x **16** + **7** x **1** = **240** + **7** = **247**

BinaryGround.com
ENJOY LEARNING BINARY & HEXADECIMAL

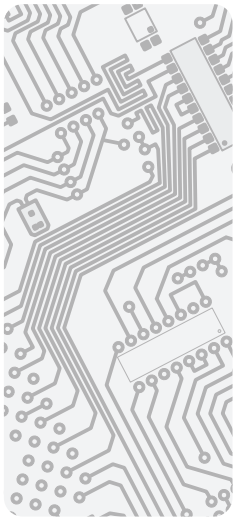
MORE ABOUT BINARY AND HEXADECIMAL NUMBERS

WHAT IS IT GOOD FOR?

Binary is the natural numeral system for **computers**, because they **can easily store and process binary numbers** with just „on“ and „off“. Early experiments with more complicated systems (e.g. ten different states to represent 0..9 per storage unit) were not successful.

Binary numbers are usually arranged into **bytes consisting of 8 bits** or as multiples of 8 bits (up to 64 bits). These are also units of processing and storage in computers. Here comes **Hexadecimal**: Each byte can be expressed with exactly two hexadecimal digits.

Hexadecimal numbers are a very compact way to write down or display the value of binary numbers. Would you use binary instead, you'd need eight digits for each byte, as opposed to just two hexadecimal digits.



MORE MATH BACKGROUND

We've explained that you assign a value to each position of a number. The value of the positions changes from right to left. It always starts with 1, independent of the numeral system. When you move to the left, the value is always multiplied by the **base** for each position **(base 10 for Decimal, 2 for Binary, 16 for Hexadecimal)**.

There is a mathematical way to express this: You assign a position number to each digit, starting from 0 for the right-most position, 1 for the next position to the left and so on. Now the value of each position for a given numeral system always equals **base^{position}**.

Example

Number (hexadecimal):	**2 F 7**
Position:	**2 1 0**
Value (decimal):	(F=15) **2 x 16^2 + 15 x 16^1 + 7 x 16^0**
	= 2 x 256 + 15 x 16 + 7 x 1
	= 512 + 240 + 7
	= 759