**Character Encoding**

We know how to encode numbers in binary – 15 in decimal is 00001111 in binary, or 0x0f in hexadecimal, and so on.

However, how do we encode a character, or a string of characters?

We use **character sets**. These are mappings between languages and a numeric value. We can then represent this value in binary or hexadecimal.

Two of the most well-known character sets are ASCII and Unicode.

There are different variants, i.e. UTF-8, UTF-16 and UTF-32 are both Unicode variants that encode characters using 8 bits and 16 bits respectively. 16 bits are required to encode a wider range of characters including foreign languages (Unicode has around 109,000 characters at the last count). 8 bits can encode a maximum of 255 different characters.

ASCII is the older standard, is 8-bit, and encodes 128 different characters including the English alphabet.

Different applications use different standards. There are also other standards which you may see referred to as **code pages** – essentially these are the same as character sets.

**Exercises:**

1. Find an ASCII character map using Google.  
   Encode the following message, including punctuation and spaces, into hexadecimal:

“Stalybridge, Hyde, U.K.”

1. What is the Unicode reference number for the symbol that denotes pi? (Use Google to help)
2. Use the ASCII chart to decode the following message, encoded in decimal.

82 69 86 73 83 69 32 72 65 82 68 44 32 69 88 65 77 32 83 79 79 78 33