**QR Code Tutorial**

**Format and Version Information**

The final step to creating a QR Code is to create the format and version strings, then place them in the correct locations in the QR code. This page explains how to generate the format and version strings, and illustrates where they are placed in the QR code.

**Format String**

The format information string encodes which error correction level and which mask pattern is in use in the current QR code. Since there are four possible error correction levels (L, M, Q, and H) and eight possible mask patterns, there are 32 (4 times 8) possible format information strings. The next section explains how to generate these format strings. For a complete list of the 32 format strings, please refer to the format string table.

**Generate the Format String**

The format string is always 15 bits long. To create the string, you first create a five bit string that encodes the error correction level and the mask pattern in use in this QR code. Then you use those five bits to generate ten error correction bits. The resulting fifteen bits are XORed with the mask pattern 101010000010010. This process is explained in detail below.

**The Error Correction Bits**

The first step to creating the format string is to get the two bits that specify the error correction level in use in the QR code. The following table shows the bit sequences for each error correction level.

|  |  |  |
| --- | --- | --- |
| **Error Correction Level** | **Bits** | **Integer Equivalent** |
| L | 01 | 1 |
| M | 00 | 0 |
| Q | 11 | 3 |
| H | 10 | 2 |

**Format and Version String Tables**

A QR code uses error correction encoding and mask patterns. The QR code's size is represented by a number, called a version number. To ensure that a QR code scanner accurately decodes what it scans, the QR code specification requires that each code include a format information string, which tells the QR code scanner which error correction level and mask pattern the QR code is using. In addition, for version 7 and larger, the QR code specification requires that each code include a version information string, which tells the QR code scanner which version the code is. This page lists all the possible format and version strings.

**About Format Information Strings**

QR Codes use error correction encoding. This is a way of generating redundant data that QR code scanners can use to detect and fix errors in the scanned code.

QR codes also use mask patterns. A mask pattern is an algorithm for changing the color (dark to light or light to dark) of a certain pattern of pixels in the code in order to make it easier for scanners to read accurately.

QR codes must include a format string that contains the information about which level of error correction coding and which mask pattern are in use in the code. This page lists all 32 possible format strings.

List of all Format Information Strings

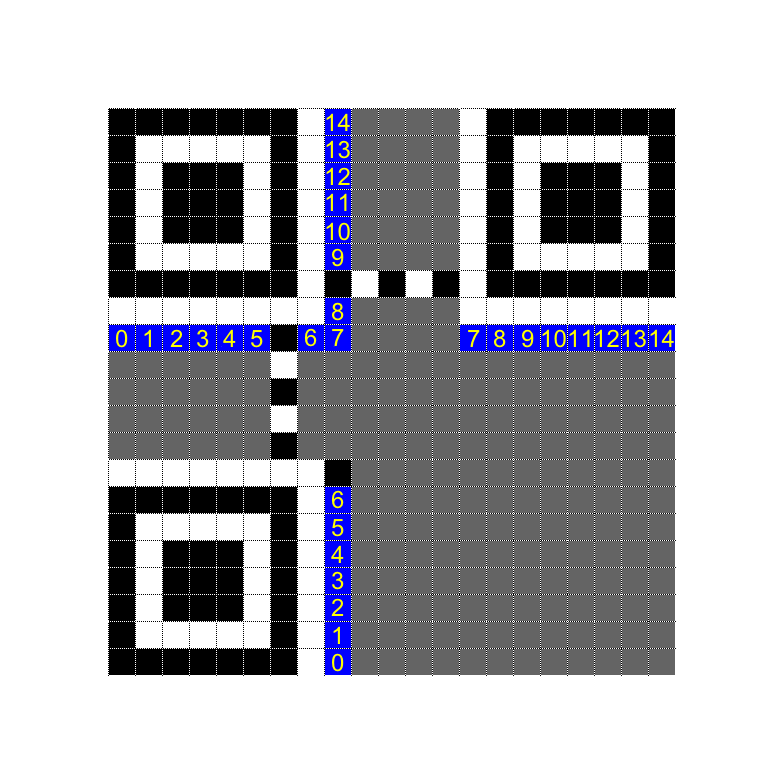
|  |  |  |
| --- | --- | --- |
| **ECC Level** | **Mask Pattern** | **Type Information Bits** |
| L | 0 | 111011111000100 |
| L | 1 | 111001011110011 |
| L | 2 | 111110110101010 |
| L | 3 | 111100010011101 |
| L | 4 | 110011000101111 |
| L | 5 | 110001100011000 |
| L | 6 | 110110001000001 |
| L | 7 | 110100101110110 |
| M | 0 | 101010000010010 |
| M | 1 | 101000100100101 |
| M | 2 | 101111001111100 |
| M | 3 | 101101101001011 |
| M | 4 | 100010111111001 |
| M | 5 | 100000011001110 |
| M | 6 | 100111110010111 |
| M | 7 | 100101010100000 |
| Q | 0 | 011010101011111 |
| Q | 1 | 011000001101000 |
| Q | 2 | 011111100110001 |
| Q | 3 | 011101000000110 |
| Q | 4 | 010010010110100 |
| Q | 5 | 010000110000011 |
| Q | 6 | 010111011011010 |
| Q | 7 | 010101111101101 |
| H | 0 | 001011010001001 |
| H | 1 | 001001110111110 |
| H | 2 | 001110011100111 |
| H | 3 | 001100111010000 |
| H | 4 | 000011101100010 |
| H | 5 | 000001001010101 |
| H | 6 | 000110100001100 |
| H | 7 | 000100000111011 |

**About Version Information Strings**

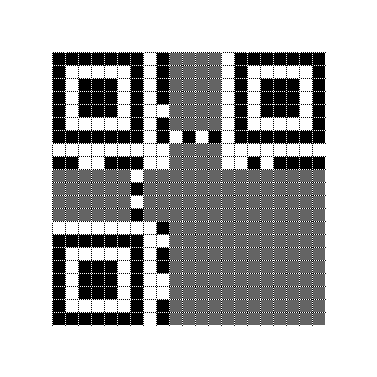
The size of a QR code is represented by a number called the version number. Codes that are version 7 and larger must include two 6x3 rectangular blocks that contain the version information string.

|  |  |
| --- | --- |
| **Version** | **Version Information String** |
| 7 | 000111110010010100 |
| 8 | 001000010110111100 |
| 9 | 001001101010011001 |
| 10 | 001010010011010011 |
| 11 | 001011101111110110 |
| 12 | 001100011101100010 |
| 13 | 001101100001000111 |
| 14 | 001110011000001101 |
| 15 | 001111100100101000 |
| 16 | 010000101101111000 |
| 17 | 010001010001011101 |
| 18 | 010010101000010111 |
| 19 | 010011010100110010 |
| 20 | 010100100110100110 |
| 21 | 010101011010000011 |
| 22 | 010110100011001001 |
| 23 | 010111011111101100 |
| 24 | 011000111011000100 |
| 25 | 011001000111100001 |
| 26 | 011010111110101011 |
| 27 | 011011000010001110 |
| 28 | 011100110000011010 |
| 29 | 011101001100111111 |
| 30 | 011110110101110101 |
| 31 | 011111001001010000 |
| 32 | 100000100111010101 |
| 33 | 100001011011110000 |
| 34 | 100010100010111010 |
| 35 | 100011011110011111 |
| 36 | 100100101100001011 |
| 37 | 100101010000101110 |
| 38 | 100110101001100100 |
| 39 | 100111010101000001 |
| 40 | 101000110001101001 |

The order of placement of format and version information string



The image below shows the example format string, 110011000101111, in a version 1 QR code.



For more information, visit https://www.thonky.com/qr-code-tutorial/