How to handle individual bits using QBitArray in Qt

This code snippet demonstrates how to handle individual bits in Qt using QBitArray.

`QBitArray` is an array of the bits, each of which can be handled separately. It also supports bitwise AND, OR, NOT, XOR operation similar to standard C++.

Preconditions

- Download and install the Qt SDK

Various Function

- Replaces len bytes from index position pos with the byte array after, and returns a reference to this byte array.

```
QByteArray x("Say yes!");
QByteArray y("no");
x.replace(4, 3, y);
```

- Sets the byte array to the printed value of n in base base (10 by default) and returns a reference to the byte array. The base can be any value between 2 and 36.

```
QByteArray ba;
int n = 63;
ba.setNum(n);
ba.setNum(n, 16);
```

- Returns the number of bytes in this byte array.

```
QByteArray ba("Hello");
int n = ba.size();
ba.data()[0];
ba.data()[4];
ba.data()[5];
```

Source File

More About QBitArray visit:http://pepper.troll.no/s60prereleases/doc/qbytearray.html

```
#include <QApplication>
#include <QBitArray>
#include <QWidget>
int main(int argc, char *argv[])
{
 QApplication app(argc, argv);

 QWidget *win = new QWidget;
 QBitArray x(15,true); // Create a bit arrray x of the size 15 and set all the bits. Second argument is optional.
 // Second way to initialize a bit array
 QBitArray a;
 a.resize(2);
 a[0] = false;
 a[1] = true;
 // Third way to initialize a bit array
 QBitArray a(2);
  a.setBit(0, false);
  a.setBit(1, true);
 QBitArray x(4);
  x.setBit(2, true);
  // x: [ 0, 0, 1, 0 ]

  QBitArray y(4);
  y.setBit(3, true);
  // y: [ 0, 0, 0, 1 ]

  x |= y;//OR operation
  // x: [ 0, 0, 1, 1 ]

 x &= y; // AND operation
 // x: [ 0, 0, 0, 1 ]

 x ^= y; // XOR operation
 // x: [ 0, 0, 0, 0 ]

 x =  ~x; // NOT operation
 // x: [ 1, 1, 1, 1 ]

}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.