

qt_信号与槽的不同模式与执行顺序

在 Qt 中，`emit` 关键字用于发送信号。如果在发送信号的代码后面还有其他代码语句，执行的顺序会受到信号与槽连接类型的影响。下面是几种常见的情况：

1. 默认连接类型（Qt::AutoConnection）

`Qt::AutoConnection` 是默认的连接类型，Qt 会根据信号和槽所在的线程自动选择连接类型：

- 如果信号和槽在同一个线程中，`Qt::AutoConnection` 等效于 `Qt::DirectConnection`。
- 如果信号和槽在不同的线程中，`Qt::AutoConnection` 等效于 `Qt::QueuedConnection`。

2. 直接连接（Qt::DirectConnection）

当使用 `Qt::DirectConnection` 时，信号会立即调用连接的槽函数。这种情况下，槽函数会在发送信号的那一行立即执行，然后继续执行信号后面的代码。

```
1 void MyClass::myFunction() {
2     emit mySignal();
3     // 这里的其他代码会在 mySignal() 的所有直接连接的槽函数执行后再执行
4     doSomethingElse();
5 }
```

3. 队列连接（Qt::QueuedConnection）

当使用 `Qt::QueuedConnection` 时，信号会被放入接收对象所属线程的事件队列中，槽函数会在事件循环处理时被执行。这种情况下，信号后面的代码会立即执行，而槽函数会稍后执行。

```
1 void MyClass::myFunction() {
2     emit mySignal();
3     // 这里的其他代码会立即执行，不等待 mySignal() 的槽函数执行
4     doSomethingElse();
5 }
```

4. 独立线程连接 (Qt::BlockingQueuedConnection)

当使用 `Qt::BlockingQueuedConnection` 时，发送信号的线程会阻塞，直到槽函数执行完毕。这种连接类型通常用于需要确保信号和槽函数在不同线程中按顺序执行的情况。

```
1 void MyClass::myFunction() {
2     emit mySignal();
3     // 这里的其他代码会等待 mySignal() 的槽函数执行完毕后再执行
4     doSomethingElse();
5 }
```

5. 唯一连接 (Qt::UniqueConnection)

`Qt::UniqueConnection` 确保同一个信号与同一个槽之间只有一个连接，连接类型可以与上述任何一种结合使用。

示例代码

```
1 void MyClass::myFunction() {
2     emit mySignal();
3     qDebug() << "This will be printed immediately after emitting the
    signal.";
4 }
```

假设 `mySignal()` 连接到以下槽函数：

```
1 void MyClass::mySlot() {
2     qDebug() << "Slot function executed.";
3 }
```

- 如果 `mySignal` 和 `mySlot` 在同一个线程中且使用 `Qt::AutoConnection` 或 `Qt::DirectConnection`，输出将是：

```
1 Slot function executed.
2 This will be printed immediately after emitting the signal.
```

- 如果 `mySignal` 和 `mySlot` 在不同的线程中且使用 `Qt::AutoConnection` 或 `Qt::QueuedConnection`，输出将是：

- 1 | This will be printed immediately after emitting the signal.
- 2 | Slot function executed.

因此，信号 `emit` 之后的代码执行顺序取决于连接类型的选择。