

Nouveaux membres

- Le signaux et slots sont des méthodes d'une classe. Pour pouvoir créer son propre signal ou slot dans une classe, il faut que celle-ci dérive directement ou indirectement de QObject.
- Dès que l'on doit créer un signal ou un slot personnalisé, il est nécessaire de définir une macro dans le header de la classe. Cette macro porte le nom de Q_OBJECT (tout en majuscules) et doit être placée tout au début de la déclaration de la classe :

Création de slots

- Exemple d'un slot changerLargeur qui change la largeur de la fenêtre sur la valeur d'un widget QSlider
- Définition :

```
class MaFenetre : public QWidget
{
    Q_OBJECT
    public:
        explicit MaFenetre(QWidget *parent = nullptr);

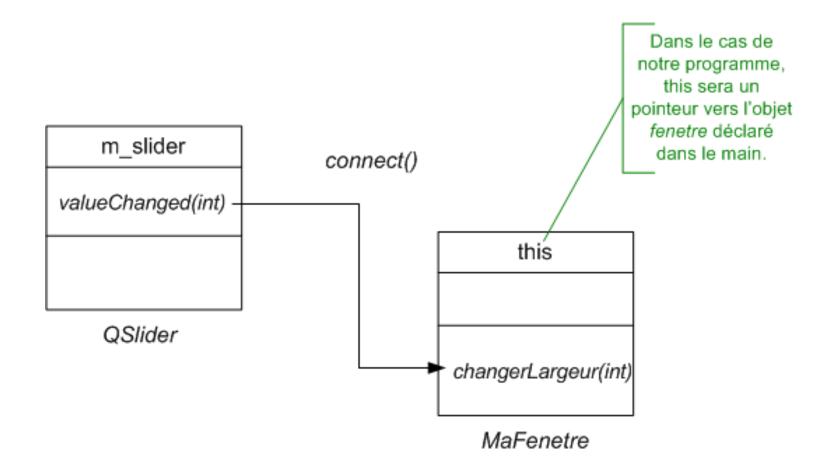
public slots:
    void changerLargeur(int largeur);

private:
    QSlider *m_slider;
};
```

• Implémentation :

```
void MaFenetre::changerLargeur(int largeur) {
    setFixedSize(largeur, 100);
}
```

• Connexion:



Création de signals

• Exemple : si le slider arrive à sa valeur maximale (600), alors on émet un signal agrandissementMax avec emit. Notre fenêtre doit pouvoir émettre l'information indiquant qu'elle est agrandie au maximum.

• Définition :

```
7 ▼ class MaFenetre: public QWidget
    {
 9
         Q OBJECT
         public:
10
             explicit MaFenetre(QWidget *parent = nullptr);
11
         public slots:
12
             void changerLargeur(int largeur);
13
         signals:
14
             void agrandissementMax();
15
16
         private:
             QSlider *m slider;
17
    };
18
```

Implémentation

```
void MaFenetre::changerLargeur(int largeur) {
    setFixedSize(largeur, height());
    if (largeur == 600) {
        emit agrandissementMax();
    }
}
```

Un signal peut passer un ou plusieurs paramètres emit Un signal doit toujours renvoyer void

Connexion

Pour connecter le signal au slot voulu, dans le constructeur de MaFenetre, on ajoute :

```
QObject::connect(this, SIGNAL(agrandissementMax()),
qApp, SLOT(quit()));
```

