

# IFT159

## Analyse et programmation

Commentaires avec Doxygen

Gabriel Girard et Benoît Fraikin

Département d'informatique



16 septembre 2009

## Doxygen

- 1 Qu'est-ce que Doxygen
- 2 Comment utiliser Doxygen
- 3 Comment documenter votre code
  - Commandes pour les blocs de documentation
  - Commandes pour l'information de base
  - Commandes pour les entrées et sorties
  - Commandes pour les paramètres des fonctions
  - Commandes de mise en forme supplémentaire

## Qu'est-ce que Doxygen

- Doxygen est un outil de documentation
- Il permet de créer des documentations techniques pour notamment le C et le C++
- Nous utilisons seulement quelques commandes de base

## Qu'est-ce que Doxygen

- Commande : `doxygen ift159.cfg`
- `ift159.cfg` se trouve sur la page WEB du cours
- La commande produit des fichiers de documentation en HTML
- Pour ouvrir la documentation dans le navigateur, ouvrir le fichier `index.html` contenu dans le répertoire `doc/html/`

## Bloc de documentation

- Un bloc de documentation est repris tel quel par Doxygen pour générer la documentation
- Deux formes
  - `/** ... */`
  - `///`
- *N.B. : il en existe d'autres mais nous ne les utiliserons pas dans le cadre de ce cours*

## Information de base

- `\file` : nom du fichier contenant le programme
- `\author` : nom de ou des auteurs
- `\version` : version du programme
- `\date` : date de création ou de modification
- `\brief` : brève description du programme

## Exemple 1

```
/**
 * \file main.cpp
 * \brief Programme de demonstration
 * \author Gabriel Girard
 * \version 0.1
 * \date 11 septembre 2009
 *
 * Description detaillee .....
 *
 */
```

## Exemple 2

```
/**
 * \file consommation.cpp
 * \brief Calcul de consommation d'essence d'une voiture.
 * \author Gerard Houdeville
 * \date 19 aout 2006 (derniere version)
 * \date 8 janvier 1996 (creation)
 * \version v1.4 : 19 aout 2006, mise a jour aux normes, Benoit Fraikin
 * \version v1.3 : 3 janvier 2004, ajout de documentation, Benoit Fraikin
 * \version v1.2 : 16 juillet 2003, modification legere, Gabriel Girard
 * \version v1.1 : 8 janvier 1996, modification legere, Gerard Houdeville
 */
```

## Documentation des entrées et sorties

- `\b Entrée ou Sortie`
- `\li (medium) nom : description (type)`
- `medium` = écran, clavier ou fichier (pour le cours)
- `type` = int, float, double, ...

## Exemple 3

```
/**
 \file consommation.cpp
 \brief Calcul de consommation d'essence d'une voiture.
 \author Gerard Houdeville

 \b Entrées :
 \li (clavier) distance parcourue (float - positif non nul)
 \li (clavier) volume utilise (float - positif non nul)

 \b Sorties
 \li (ecran) consommation (float - positif)
 */
```

## Documentation des paramètres de fonctions

- `\param[direction] nom description`
- `\return description`
- `direction` = [in], [out] ou [in, out]
- `type` = int, float, double, ...

## Exemple 4

```
/**
 \brief Cette fonction calcule l'aire d'un cercle

 Calcule l'aire d'un cercle dont le rayon est
 passe en parametre.
 L'aire vaut PI * (rayon)^2.
 La constante PI est approximee a 3,14159.

 \param[in] rayon le rayon du cercle
 \return l'aire du cercle de rayon "rayon"
 */
```

## Exemple 5

```
/**
 * \brief fonction qui calcule la somme et la moyenne
 *       d'un tableau de valeurs
 *
 * \param[in] valeurs un tableau de valeurs
 * \param[out] somme la somme des valeurs
 * \param[out] moyenne la moyenne des valeurs
 *
 * \return une valeur de verite indiquant s'il y a eu
 *         une erreur durant le calcul
 */
```

## Mise en forme

- \c suivi d'un mot : met le mot en « typewriter »
- \e suivi d'un mot : met le mot en « italique »
- \b suivi d'un mot : met le mot en « gras »

## Règles

- à utiliser avec parcimonie
- \c sert pour un mot du programme
- \e sert pour un mot important
- \b est à réserver pour des « titres »

## Retour exemple 4

```
/**
 * \brief Cette fonction calcule l'aire d'un cercle
 *
 * Calcule l'aire d'un cercle dont le rayon \c rayon est
 * passe en parametre.
 * L'aire vaut \c PI * (\c rayon)^2.
 * La constante \c PI est approximee a 3,14159.
 *
 * \param[in] rayon le rayon du cercle dont on veut
 *              obtenir l'aire
 * \return l'aire du cercle de rayon \c rayon
 */
```

## Production de l'exemple 4

```
float calcule_aire ( float rayon )
```

Cette fonction calcule l'aire d'un cercle.

Calcule l'aire d'un cercle dont le rayon est passe en parametre. L'aire vaut  $\pi * (\text{rayon})^2$ . La constante  $\pi$  est approximee a 3,14159.

**Paramètres:**

[in] *rayon* le rayon du cercle dont on veut obtenir l'aire

**Renvoie:**

l'aire du cercle de rayon *rayon*

Voir les exemples en ligne