# Master 2 LITL
# Programmation pour le TAL (SLT0905V)
# mot-clé `this`

Franck Sajous/CLLE-ERSS


`http://fsajous.free.fr/`

# this

Mot clé utilisé par un objet pour se désigner lui-même, e.g. pour passer son adresse comme argument d'une méthode (on y reviendra).

## Autre utilisation possible

Sous Eclipse :

```
class C {
  int i;
  String s;
  Token t;
}
```

Clic-droit dans le code source, puis, dans le menu contextuel : *Source/Generate Getters and Setters*.
Cocher les différents membres et cliquer sur OK.

# this - getters et setters générés automatiquement

```
class C {
  private int i;
  private String s;
  private Token t;

  public int getI() {
    return i;
  }
  public void setI(int i) {
    this.i = i;
  }
  public String getS() {
    return s;
  }
  public void setS(String s) {
    this.s = s;
  }
  public Token getT() {
    return t;
  }
  public void setT(Token t) {
    this.t = t;
  }
}
```

# this - accès à un membre dont la visibilité est masquée

## Membre/argument

- `defaultColor` : membre de la classe `HTMLToken`
- `dc` : argument de la méthode setDefaultColor

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color dc)
    {
        defaultColor = dc;
    }
}
```

# this - accès à un membre dont la visibilité est masquée

## Membre/argument

- `defaultColor` : membre de la classe `HTMLToken`
- `dc` : argument de la méthode setDefaultColor
- affectation `defaultColor = dc;`
  noms différents, pas de problème : on affecte la valeur de l'argument `dc` au membre `defaultColor`

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color dc)
    {
        defaultColor = dc;
    }
}
```

# this - accès à un membre dont la visibilité est masquée

## Membre/argument

Si on nomme `defaultColor` l'argument de la méthode :

- il masque le membre (de même nom) de la classe

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color defaultColor)
    {
        defaultColor = defaultColor;
    }
}
```

# this - accès à un membre dont la visibilité est masquée

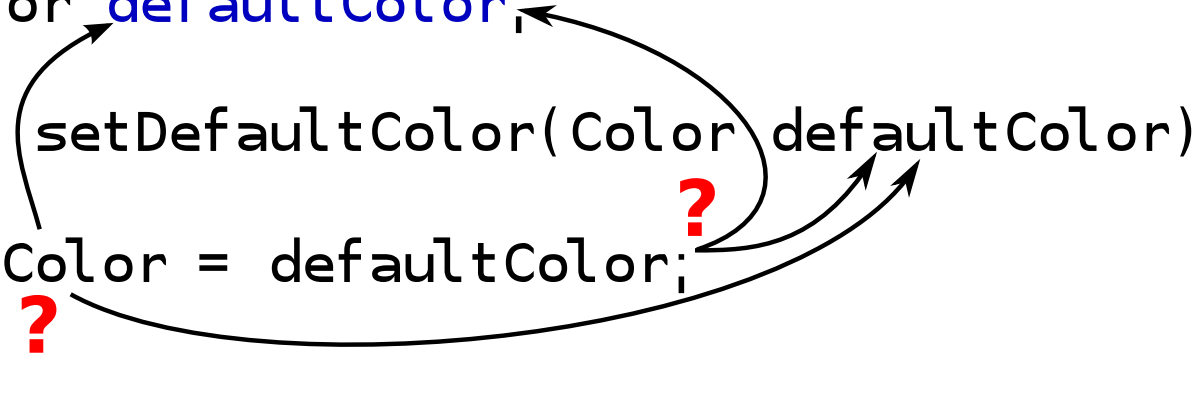
## Membre/argument

Si on nomme `defaultColor` l'argument de la méthode :

- il masque le membre (de même nom) de la classe
- que produit l'instruction d'affectation ?

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color defaultColor)
    {
        defaultColor = defaultColor;
    }
}
```



# this - accès à un membre dont la visibilité est masquée

## Membre/argument

Si on nomme `defaultColor` l'argument de la méthode :

- il masque le membre (de même nom) de la classe
- que produit l'instruction d'affectation ?
- rien : `defaultColor` (des deux côtés du signe =) fait référence à l'argument de la méthode. On affecte à cet argument sa propre valeur.

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color defaultColor)
    {
        defaultColor = defaultColor;
    }
}
```

# this - accès à un membre dont la visibilité est masquée

## Membre/argument

- solution : le mot-clé `this`

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color defaultColor)
    {
        this.defaultColor = defaultColor;
    }
}
```

# this - accès à un membre dont la visibilité est masquée

## Membre/argument

- solution : le mot-clé `this`
- `this.defaultColor` → membre `defaultColor` de la classe
  `defaultColor` → argument de la méthode

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color defaultColor)
    {
        this.defaultColor = defaultColor;
    }
}
```

# this - accès à un membre dont la visibilité est masquée

## Membre/argument

- solution : le mot-clé `this`
- `this.defaultColor` → membre `defaultColor` de la classe
  `defaultColor` → argument de la méthode
- on affecte la valeur de l'argument au membre de la classe

```
public class HTMLToken extends Token
{
    private Color defaultColor;

    public void setDefaultColor(Color defaultColor)
    {
        this.defaultColor = defaultColor;
    }
}
```