

# Aide mémoire NXC

**Attention** : Différence entre les minuscules et les majuscules : `max` et `Max` sont différents

## Commentaire

```
/* commentaire à la C */
// commentaire à la C++
```

## Les données

Type : `bool`, `char`, `short`, `int`, `long`, `float`, `string`...

Déclaration : `Int a; Int a = 0; Int a,b;`

Déclaration de tableaux : `int T[10]; // tableau de 10 cases T[0], T[1], ..., T[9]`

## Les structures de données

```
struct robot
{
  string name;
  int vitesse;
};

task main() {
  robot monRobot;
  monRobot.name = "Astro";
  monRobot.vitesse = 0 ;
  NumOut(10,LCD_LINE3,monRobot.vitesse) ;
}
```

## Les opérateurs

<code>=</code>	Affectation	<code>x = 2</code>
<code>+=, -=</code>	Affectation avec addition / soustraction	<code>x += 2</code>
<code>abs()</code>	Valeur absolue	<code>abs(x)</code>
<code>++, -</code>	Incrémentation / décrémentation	<code>x++ / ++x</code>
<code>*, /, %</code>	Multiplication, division, modulo	<code>x * y</code>
<code>+, -</code>	Addition, soustraction	<code>x + y</code>
<code>&lt;, &gt;, &lt;=, &gt;=</code>	Opérateur relationnel	<code>x &lt; y</code>
<code>==, !=</code>	Test d'égalité / non égalité	<code>x == 1</code>
<code>&amp;&amp;</code>	Et logique	<code>x &amp;&amp; y</code>
<code>  </code>	Non logique	<code>x    y</code>

## Fonctions sur les chaînes de caractères

```
x = StrToNum(str);      // renvoie la valeur numérique de str
str = NumToStr(x);     // renvoie la chaîne de caractère de l'entier x
i = StrLen(str);       // renvoie la longueur de str
str = StrCat(str1,str2); // str est composé de str1 suivi de str2
```

## Les structures algorithmiques

```
if (condition)
{instructions}
```

```
if (condition)
{instructions}
else{instructions}
```

```
while (condition)
{instructions}
```

```
do {instructions}
while (condition)
```

```
for(statement1 ; condition ;
statement2)
{instructions}
```

```
switch(x)
{
case 1:
    // do something when x is 1
    break;
case 2:
case 3:
    // do something else when x is 2 or 3
    break;
default:
    // do this when x is not 1, 2, or 3
    break;
}
```

```
repeat (expression)
{instructions}
```

## Les Fonctions et macros NXC

Déclaration :

```
void MaFonction (int a, int b)
{
    instructions
}
```

```
#define MaMacro {instructions}
```

Utilisation :

```
MaFonction (12, 3) ;
```

```
Ma Macro ( ) ;
```

## Passage de paramètre

Par valeur : int a

Par valeur constante : const int a

Par référence : int & a

Plus de détails sur le site : <http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html>