

ACT2 Les opérations bit à bit

BILAN COMPETENCES

ANA	Décrire et spécifier les caractéristiques d'un processus, les données d'un problème, ou celles manipulées par un algorithme ou une fonction
COM	Communiquer à l'écrit en utilisant un langage rigoureux et des outils pertinents

Il arrive que l'on veuille connaître ou modifier l'état d'un ou plusieurs bits d'une donnée.

Par exemples :

- Changement d'état d'un bit (activation d'une sortie pour allumer ou éteindre une DEL)
- Tester l'état d'un bit (pour connaître son état : grandeur physique détectée ou non)
- Vérifier si l'ordinateur à joindre est sur le même réseau ou non
- ...

Les opérations bit à bit sont incroyablement simples et sont directement prises en charge par le processeur.

`x << y` # Décalage à gauche

`x >> y` # Décalage à droite

`x & y` # bit à bit ET

`x | y` # bit à bit OU

`~ x` # bitwise non

`x ^ y` # bit à bit OU exclusif

1 Jouer avec les opérations de base

Travail demandé :

- 1) Pour chacun des opérateurs, tester les opérations proposées.
- 2) Expliquer le fonctionnement au niveau des bits de cet opérateur à l'aide de quelques phrases.

- L'opérateur `&` exécutera un AND binaire.

```
>>> 0 & 0
>>> 0 & 1
>>> 1 & 0
>>> 1 & 1
>>> 60 & 30
>>> bin(60 & 28)
```

- Le `|` l'opérateur effectuera un OU binaire.

```
>>> 0 | 0
>>> 0 | 1
>>> 1 | 0
>>> 1 | 1
>>> 60 | 30
>>> bin(60 | 62)
```

- L'opérateur ^ exécutera un XOR binaire.

```
>>> 0 ^ 0
>>> 0 ^ 1
>>> 1 ^ 0
>>> 1 ^ 1
>>> 60 ^ 30
>>> bin(60 ^ 34)
```

- L'opérateur ~ retournera le complément de tous les bits

```
>>> ~ 0
>>> ~ 1
>>> ~ 2
>>> ~ 123
>>> bin(15&(~15))
```

- L'opérateur >> effectuera un "décalage à droite" au niveau du bit.

```
>>> 8 >> 2
>>> bin(8 >> 2)
```

- L'opérateur << effectuera un "décalage vers la gauche" au niveau du bit.

```
>>> 2 << 2
>>> bin(2 << 2)
```

2 Applications concrètes

Pour entrevoir l'intérêt des opérations précédentes, prenons quelques exemples :

Exemple 1 :

Trouver le résultat de la multiplication par 2, de la division par 2.

Exemple 2 :

On voudrait manipuler le bit 4 d'une donnée sur 8 bits connectée à une LED (par ex. les LED de la matrice 5x5 de la carte Micro:Bit).

- 1) Je veux mettre à 0 le bit 4 pour éteindre la LED.
- 2) Je veux mettre à 1 le bit 4 pour allumer la LED.
- 3) Je veux inverser l'état du bit 4 pour éteindre la LED si allumée, l'allumer si elle est éteinte.