

La structure des entiers relatifs

1 Le type

Le type est appelé `int` (comme « integers »).

2 L'ensemble des valeurs

Théoriquement le type `int` correspond à l'ensemble \mathbb{Z} des mathématiques. En fait il est borné en Python, avec borne dépendante de la machine, mais assez grande pour ignorer ce fait ici.

3 Les opérations

Les opérations les plus courantes sont :

| Opération | | | | | | Entrée | Sortie |
|-----------|---|---|----|----|----|----------------------|-------------------|
| + | - | * | // | % | ** | <code>int×int</code> | <code>int</code> |
| == | < | > | <= | >= | != | <code>int×int</code> | <code>bool</code> |

Les opérations `==` `<` `>` `<=` `>=` `!=` sont appelées *les comparaisons* et se retrouveront dans d'autres structures de données également.

4 La sémantique des opérations

Il s'agit respectivement des addition, soustraction, multiplication, division « euclidienne », modulo et puissance, qui sont habituelles sur les entiers relatifs. La division euclidienne fournit toujours un résultat entier (le quotient) et le modulo n'est autre que le reste de cette division.

Noter que la comparaison d'égalité se note `==` et non pas `=` pour ne pas la confondre avec l'affectation de variable en Python. Les comparaisons `<=` `>=` `!=` sont respectivement les versions accessibles au clavier de \leq , \geq et \neq .

5 Exemples

Rien de bien compliqué :

```
>>> type(46)
<type 'int'>
>>> 5 * 7
35
>>> print(5 * 7)
35
>>> 5 // 2
2
>>> 5 % 2
1
>>> 5 ** 3
125
```

et les comparaisons :

```
>>> 5 < 3
```

```
False
>>> 5 == 3 + 2
True
>>> 5 = 3 + 2
File "<stdin>", line 1
SyntaxError: can't assign to literal
>>> 5 <= 5
True
```