

1 Typage

Les données utilisées dans un programme peuvent être de différentes natures. Une *chaîne de caractère* permettra l’affichage d’un message à l’écran alors qu’un *entier* sera plutôt utilisé dans un calcul.

2 Typage dynamique

Dans un programme, une variable possède un type déterminé par son contenu. La fonction Python `type()` donne le type d’une variable.

```
1 a = 42
2 type(a)
```

Ce code retourne *int* pour *integer* (*entier*).

De nombreux langages de programmation imposent de définir le *type* de chaque variable. Les raisons sont multiples :

- allouer un espace mémoire adéquat,
- éviter les erreurs de programmation.

Ce n’est pas le cas en Python :

```
1 a = 42
2 type(a)
3 a = "test"
4 type(a)
```

Ce langage de *haut-niveau* veut simplifier la tâche de l’utilisateur en *typant dynamiquement* chaque variable. De prime abord, cela peut paraître appréciable, mais engendre parfois des erreurs de codage dans un programme complexe.

```
1 a = 42
2 b = 5
3 a = "test"
4 a+b
```

3 Un code clair

En Python, il est possible d’indiquer le type d’une variable.

```
1 a: int = 42
```

Cette information n’est qu’indicative. Python ne lèvera pas d’erreur dans le cas ci-après :

```
1 a: str = 42
```

4 Les types de base

- **int** : *integer* ; nombre entier (42)
- **str** : *string* ; chaîne de caractère ("test")
- **bool** : *boolean* ; valeur booléenne (True ou False)
- **float** : *flottant* ; nombre à virgule flottante (4.2)