

TYPAGE DES VARIABLES

Ce qui suit complète [le livret de cours1 Python page 3 B1](#).

Dans un langage de programmation, le type d'une variable est la catégorie à laquelle cette variable appartient. Ce type peut être d'une catégorie prédéfinie : entier, réel, booléen, caractère. La plupart des langages de programmation propose ces types basiques.

Les langages proposent aussi bien souvent des types plus complexes : listes, tableaux etc.

I. PROBLÉMATIQUE DU TYPAGE.

- Comment une variable reçoit-elle son type initial à l'intérieur d'un programme ?
- Comment lors de l'exécution du programme est vérifiée la concordance entre le type d'une variable et le type de la valeur affectée à cette variable ?
- Peut-on effectuer des opérations avec des valeurs de types différents (ex : $N = 1 + '2'$) ? Ce qui suppose des conversions implicites de type.

Ces 3 questions posent la problématique du typage.

II. ATTRIBUTION INITIALE DES TYPES AUX VARIABLES A L'INTERIEUR DU PROGRAMME : TYPAGE EXPLICITE VS TYPAGE IMPLICITE.

➤ Lorsque l'attribution du type à une variable est écrite clairement dans une ligne de code par une déclaration de type, on parle de typage explicite. On pourrait dire aussi typage déclaratif ou typage impératif mais ces termes ne sont pas employés.

Lorsque l'attribution du type à une variable est sous-entendue, sans qu'il y ait donc de déclaration de type, on parle de typage implicite.

➤ Exemple : On veut déclarer une variable N de type entier et lui assigner la valeur 3.

- En langage C, cela s'écrit : `int N = 3`

Le type de la variable N en tant qu'entier est déclaré clairement par le programmeur à l'aide du mot clé int. Le typage est donc ici explicite. Puis il y a affectation de l'entier 3 à cette variable N.

- En Python, cela s'écrit simplement : `N = 3`

Le type de N n'est pas déclaré explicitement mais comme 3 est un entier et qu'il est affecté à N, le compilateur ou l'interpréteur va déduire de ce contexte le type de la variable N. Ce mécanisme déductif s'appelle l'inférence de type. Cette inférence de type a induit ici un typage implicite.

➤ Certains langages comme le langage C peuvent faire à la fois du typage explicite et du typage implicite dans un même programme.

III. VERIFICATION DU TYPE DES VARIABLES : TYPAGE STATIQUE VS TYPAGE DYNAMIQUE.

- Lorsque la vérification du type des variables se fait à la compilation, on parle de typage statique.
Exemples : Langages C, C++, Java sont à typage statique.
- Lorsque la vérification des types se fait à l'exécution (« à la volée »), on parle de typage dynamique.
Exemples : Python, Javascript sont à typage dynamique.
- En général, le typage statique induit le typage explicite à l'intérieur du code . Et le typage dynamique induit en général le typage implicite par inférence de type à l'intérieur du code.

IV. CONVERSIONS IMPLICITES DE TYPE : TYPAGE FORT VS TYPAGE FAIBLE.

Bien qu'il n'existe pas de définition exacte du typage fort ou faible, on peut dire :

- Typage fort (Python, C++, Java etc.) : pas de conversions implicites de type.
Ainsi donc, l'exemple $N = 1 + '2'$ renvoie une type error. Un changement de type doit être écrit explicitement dans le code.
- Typage faible (Php, Javascript, Perl etc.) : conversions implicites de type autorisées !
Ainsi donc, l'exemple $N = 1 + '2'$ renvoie quelque chose et non une type error.
- Suivant les langages, le typage sera plus ou moins fort ou faible :
Certains langages autorisent $N = 1 + '2'$ mais pas $N = 1 + 'a'$.
D'autres n'autoriseront aucun des deux ou autoriseront les deux (Javascript typage très faible).

V. CONCLUSION.

Le typage est l'une des caractéristiques essentielles (avec le paradigme-style de programmation) qui permet de classer les langages de programmation.

Voici un tableau avec quelques exemples de langages différenciés suivant le typage :

Typage	Fort	Faible
Statique	Ada, C++, Java, Haskell, etc.	Langage C
Dynamique	Python, Ruby etc.	Javascript qui autorise "2" + 4 → "24" Php

Ainsi Python fait partie des langages à typage et