
SNT - TP4

1 Un nouveau type de variable

Dans le TP2, on a vu deux types de variables:

Affichage Python	type()	Type
<code><class 'str'></code>	<code>'str'</code>	Texte entre guillemets
<code><class 'int'></code>	<code>'int'</code>	Nombre entier

Q1 - Créer une variable, que l'on appellera `x`, en lui affectant la valeur 2.5.

Q2 - En plus, afficher le type de `x` (on utilisera les commandes `print()` et `type()`). Est-ce un type de variable déjà rencontré ?

2 Les flottants

2.1 Flottants vs entiers

On dispose maintenant d'un nouveau type de variable:

Affichage Python	type()	Type
<code><class 'float'></code>	<code>'float'</code>	Nombre à virgule

On appelle ce type de variable le flottant. Il représente tous les nombres à virgule.

Q3 - Créer une variable `x` et lui affecter la valeur 12.0. Quel-est le type de `x` ?

Q4 - Quel sera le type d'une variable ayant pour valeur un nombre à virgule comme 1.0, 2.0, 3.0... ?

△ Attention !!! -

En Python, les variables ayant pour valeur les nombres 0, 1, 2, 3... sont représentées par la classe des entiers, tandis que les variables ayant pour valeur les nombres 0.0, 1.0, 2.0, 3.0... sont représentées par la classe des flottants. Il est important de distinguer ces deux types de variables, car certaines opérations tolèrent un type de variable, mais pas un autre, et ce même si elles sont égales !

Q5 - Créer deux variables x et y, leur affecter la valeur 1 et 1.0. Calculer $z = x + y$. Quel est le type de z ?

En Python, additionner un entier et un flottant ne pose aucun problème. Le nombre entier sera alors considéré comme un flottant avec zéro derrière la virgule. Par exemple, 15 sera considéré comme 15.0.

2.2 Erreurs d'arrondi

Q6 - Créer deux variables a et b, leur affecter respectivement la valeur 0.1 et 0.2. Afficher a+b.

Q7 - A-t-on une différence avec ce qu'il se passe "dans la vraie vie" ? Quelle est la valeur de la différence (approximative) ?

△ Attention !!! -

En Python, les variables de type flottant sont représentées avec un nombre **fini** de 0 et 1 du langage binaire (une version adaptée aux nombres à virgules des cartes de l'activité de début d'année). On ne peut donc pas représenter tous les nombres à virgule que l'on souhaite, d'où les erreurs d'arrondi.

Q8 - Avec Python, peut-on écrire n'importe quel nombre réel, rationnel, décimal ?

2.3 Propagation des erreurs d'arrondi

Q9 - Copier-coller ces lignes de code:

```
x = 10 ** 16
difference = (x + 1.0) - x
print(difference)
```

Quel résultat est affiché ? Que devrait-on avoir "dans la vraie vie" ?

△ Attention !!! -

En Python, plus un nombre flottant est grand, plus le risque d'erreur d'arrondi est important, il faut donc faire attention en manipulant des grands nombres à virgule.

3 Approximation de nombres rationnels

Q10 - Copier-coller ces lignes de code:

```
x = 1 / 3  
print(x)
```

Quel résultat est affiché ? Quel-est le rôle de la commande / ?

Q11 - Faire la même chose avec les nombres $\frac{1}{2}$, $\frac{1}{7}$. Quelle-est la suite de nombres derrière la virgule pour ce dernier nombre ?

Q12 - (**Bonus**) Calculer les nombres rationnels suivants:

$$\frac{22}{7} \quad \frac{303}{106} \quad \frac{355}{113} \quad \frac{103993}{33102}.$$

Quel nombre approche-t-on ?