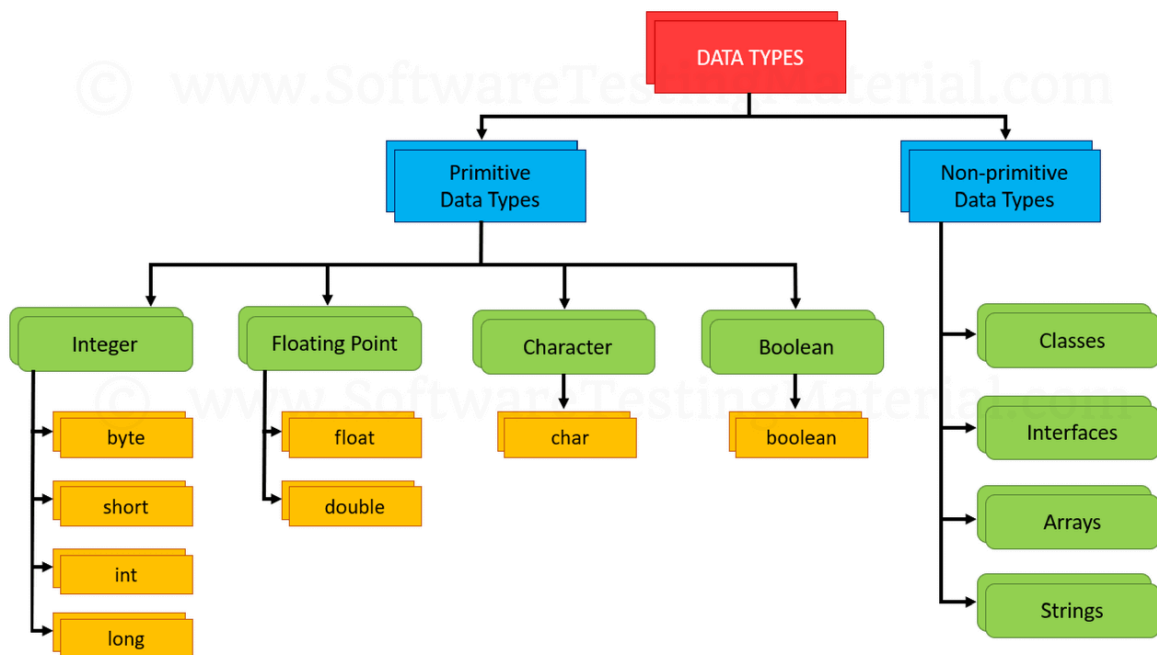


Structure de données

Version : 1.0



Les données

Qu'est ce qu'une donnée ?

Une donnée est une représentation conventionnelle d'une information en vue de son traitement informatique.

Elle définit le **point de départ d'un raisonnement** permettant de déterminer une solution à un problème relatif à cette donnée.

Une donnée peut être une description élémentaire d'une réalité (**une observation**) ou le résultat d'une comparaison de deux événements du même ordre (**une mesure**).

Donnée brute

On parle de **donnée brute** lorsque celle-ci est **dépourvue** de tout **raisonnement**, supposition, constatation ou probabilité.

Si la donnée est considérée comme indiscutable, ou indiscutée par manque de connaissance, elle peut servir de base à une recherche.

Le travail de recherche sur les données brutes permet souvent de leur donner un sens et d'obtenir des informations

Structure de données

Qu'est ce que c'est ?

Une structure de données est une manière d'organiser les données pour les traiter plus facilement.

Types de structures

Il existe différents types de structure de données correspondant à différents types de données ou à différentes contraintes algorithmiques.

Les structures finies

Les variables

Une **variable** est un **indicateur** qui associe un **nom** (l'identifiant) à une **valeur**. Dans la majorité des langages, la variable peut changer de valeur au cours du temps.

En programmation, une variable est un espace de stockage. C'est-à-dire qu'une **valeur** est **adressée à un espace mémoire** par son **identifiant**.

Souvent, une variable possède sept caractéristiques :

- Son nom, c'est l'identifiant sous lequel elle est déclarée.
- Son type, c'est la nature de la valeur de la donnée.
- Sa valeur.
- Son adresse, c'est l'espace mémoire dans lequel elle est stockée.
- Sa portée, c'est la portion de code dans lequel elle est accessible.
- Sa visibilité, c'est une règle qui définit qui peut utiliser la variable.
- Sa durée de vie, c'est le temps d'exécution pendant lequel la variable existe.

Exemple

```
nom = "john";
```

Les constantes

Une **constante** est un **indicateur à valeur fixe**. Cet indicateur possède les mêmes aspects qu'une variable, mais sa valeur ne peut être affectée qu'une seule fois.

Exemple

```
PI = 3.141592653;
```

Les enregistrements

Un enregistrement est une structure de données qui rassemble plusieurs champs contenant des valeurs qui peuvent être de types différents (entrées), comme un extrait d'un tableau à deux dimensions ou d'une base de données.

Les entrées se rapportent généralement à un même objet.

Exemple

```
Le fichier clients d'une entreprise contient plusieurs  
informations sur un client: Nom, Prénom, Numéro client, ...
```

Les structures indexées

Les tableaux

Un tableau représente une séquence finie d'éléments auxquels on peut accéder par référence à une position ou un indice (index).

On parle également de collection de données.

Le temps d'accès à un élément par son index est constant. Cela s'explique par le fait que les éléments d'un tableau sont contigus dans l'espace mémoire. L'accès est immédiat, comme il le serait pour une variable simple.

Exemple

0	"A"
1	"B"
2	"C"

```
tab[0] permet d'accéder au premier élément du tableau. // "A"
```

Les tableaux multidimensionnels

Un tableau multidimensionnel, aussi appelé **matrice**, est un tableau normal (à une dimension) dont les éléments sont eux même des tableaux.

Exemple

0	0	"A"
	1	"B"
	2	"C"
1	0	"D"
	1	"E"
	2	"F"
2	0	"G"
	1	"H"
	2	"I"

L'accès aux éléments se fait au travers de plusieurs index. L'ordre des index est important.

Exemple

```
tab[1][0] // "D"  
tab[0][1] // "B"
```

Les tableaux associatifs

Un tableau associatif, aussi appelé **dictionnaire**, est une tableau associant à un ensemble de clés de type arbitraire, un ensemble de valeurs d'un autre type. Chaque clé est associé à une seule valeur.

Les tableaux associatifs sont souvent utilisés lorsqu'une opération de recherche est fréquente.

Exemple

nom	"DOE"
prenom	"John"
age	26

L'accès aux éléments se fait au travers du nom de la clé.

Exemple

```
tab[ 'prenom' ] // "John"
```

Les vecteurs

Un vecteur est un conteneur d'éléments ordonnés et accessibles par des indices, dont la taille est dynamique, elle est mise à jour automatiquement lors d'ajout ou de suppression d'éléments.

Exemple

```
v = []  
  
v.push( 'A' ) // Empile l'élément A  
v.push( 'B' ) // Empile l'élément B  
v.pop() // Dépile un élément  
v.push( 'C' ) // Empile l'élément C
```

Les structures récursives

Une structure de données récursive est une collection de données pouvant faire référence à des valeurs du même type, cela peut être le cas d'une classe par exemple.

Exemple

Imaginons que nous définissions une classe nommée Employé. Cette Classe, qui est une collection de données, va contenir plusieurs informations tel que le Nom de l'employé, son rôle et la référence à son supérieur.

Le supérieur de notre employé est lui aussi un employé...

```
class Employe {  
    String Nom;  
    String Role;  
    Employe Superieur;  
}
```

On voit que la classe **Employé** contient une variable de type **Employé**.

Types de données

Les types Numérique

Les types de données numériques sont constitués d'entiers, de nombres à virgule flottante et de décimaux dont la précision peut être définie.

Les entiers

Un entier, ou **integer** en anglais, contient un nombre entiers, c'est-à-dire sans décimal, de valeur positive ou négative.

Les décimaux

Un décimal contient un nombre à virgule de valeur positive ou négative dont la précision est définie.

Certains langages, comme SQL, stockent les décimaux de la même façon que les entiers.

Les nombres à virgule flottante

Un nombre à virgule flottante, ou **float** en anglais, contient un nombre à virgule de valeur positive ou négative sans qu'une précision ne soit définie.

Certains langages, comme SQL, stockent les nombres à virgule flottante de la même façon que les chaînes de caractère.

Note

Lorsque l'on code, un décimal ou un nombre à virgule flottante utilisera toujours la notation anglo-saxonne. Il faut utiliser le "point" et non la "virgule".

Exemple

```
4.2
```

Les types chaînes de caractères

Une chaîne de caractères, ou **string** en anglais, est une séquence de caractères, appelé plus communément "texte"

Lorsqu'on code, une chaîne de caractères sera placée entre double guillemets (**double quotes**) ou simple guillemet (**simple quote**). Ainsi, des chiffres placés entre simple ou double quotes seront considérés comme une chaîne de caractère.

Les booléens

Un type booléen, ou **Boolean** en anglais, ne peut contenir que deux valeurs : La valeur Vrai (**true**) ou la valeur Faux (**false**).

Moins courant dans la vie de tous les jours, le Booléen est un type très utilisé dans la programmation informatique.

Pensez à ne pas stocker les boolean entre simple ou double quotes, ils deviendraient alors des chaînes de caractères.

Les types temporel

Dans certains langages, tel que SQL, les types de données temporels sont des types à part entière.

Le stockage des valeurs temporelles doit toujours respecter la notation anglo-saxonne.

Le type Date

Le type de données **Date** représente une date. Il se note avec la syntaxe **YYYY-MM-DD**.

Y pour **Year**, **M** pour **Month** et **D** pour **Day**.

Exemple

La date du 31 décembre 1999 se note

```
"1999-12-31"
```

Le type Time

Le type de données **Time** représente une heure. Il se note avec la syntaxe **HH:ii:ss**.

H pour **Hour**, **i** pour **Minutes** (le **M** étant réservé pour **Month**) et **s** pour **Seconds**.

Exemple

19 heures, 24 minutes et 36 secondes se note

```
"19:24:36"
```

Le type DateTime

Le type de données **DateTime** est une expression regroupant le type de données **Date** et le type de données **Time** séparé par un espace.

Il se note **YYYY-MM-DD HH:ii:ss**.

Exemple

Pour le 31 décembre 1999 à 19 heures 24 minutes et 36 secondes, on note

```
"1999-12-31 19:24:36"
```

Le type Timestamp

Le type de données **Timestamp** est une représentation simplifiée d'une référence au temps.

La manipulation informatique des données liées au temps - **Date** et **Time** - cause divers problèmes tels que la correspondance entre deux dates sur différents fuseaux horaires, l'inégalité du nombre de jours dans les mois ou encore les années bissextiles.

Pour pallier cette difficulté, un système d'horodatage simplifié a été mis en place : Le **Timestamp**.

Timestamp donne le nombre de secondes qui s'écoule depuis le **1er janvier 1970 à 00h00**, date de mise en marche d'UNIX.

Exemple

La date 2022-01-01 00:00:00 correspond au Timestamp

```
1640991600
```

Intérêt du Timestamp

Le protocole d'horodatage, **Timestamp**, offre de nombreux avantages en matière de traitement informatique. Il permet :

- De comparer facilement les dates en faisant une simple opération.
- De contrôler ou de manipuler facilement les langages informatiques et systèmes de gestion des bases de données.
- À tous les utilisateurs d'être sur la même longueur d'onde en matière de date : il est donc universel.

Limite du Timestamp

Le protocole d'horodatage, **Timestamp**, présente néanmoins quelques limites :

- Il est nécessaire de convertir les valeurs **Timestamp** pour les rendre compréhensible par un humain.

- Les dates avant le 1er janvier 1970 à 00h00 donnent un **Timestamp** négatif, ce qui n'est pas idéal pour la manipulation des dates de naissance.
- La limite de comptage des ordinateurs 32 bits est de 2.147.483.647 ce qui correspond à la date du 19 janvier 2038 à 03 heures 14 minutes et 7 secondes. De ce fait, les Timestamp dépassant cette valeur ne pourront pas être déchiffrés.

Les types tableaux et objets

Le type de données tableau , au **Array** en anglais, et objet, ou **Object** en anglais, ont la particularités de pouvoir stocker plusieurs types de données.

Les types NULL

Le type de données **NULL** correspond à l'absence de valeur. En PHP il représente une variable vide.

Un type de données **NULL** ne peut contenir que la valeur **NULL**.

Les types ressources

Une ressource représente une référence vers une ressource externe.

En PHP, par exemple, une ressource peut représenter une connexion vers une base de données.