

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-05-2-stringcomp-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Dernière vidéo. B minuscule. Chaînes de caractères. B majuscule. Nom de la fonction. Lettre capitale. Deuxième variable. Lettres minuscules. Transparent précédent. Traitement spécifique. Résultat de la concaténation. Chaînes. Chaîne vide. Chaîne de caractère. Fonction.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (en Java).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

String: **comparaisons** (Partie 2)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



```
String s1 = "abc";    // s1 pointe vers le littéral "abc"
String s2 = "abc";    // idem (donc même zone mémoire que s1)
String s3 = s2;       // s3 stocke la même adresse que s2
String s4 = s1 + "";  // s4 contient l'adresse d'une nouvelle chaîne
                     // (construite par concaténation)
System.out.println((s1==s2) && (s2==s3)); // affiche true
System.out.println(s4);                  // affiche abc
System.out.println((s1==s4));            // affiche false
```

Comment faire pour **comparer les contenus référencés** plutôt que les références ?

📌 **Traitement spécifique** aux **String**

Mais en pratique on a plus envie de comparer les chaînes elles-mêmes plutôt que leurs références.

notes

résumé

0m 1s



Comparaison de String

*true si égales
false si différentes*

`chaine1.equals(chaine2)` teste si les chaînes de caractères référencées par `chaine1` et `chaine2` sont constituées des mêmes caractères

```
→ String s1 = "abc";
→ String s2 = "aBc";
→ String s4 = s1 + "";

→ System.out.println(s1.equals(s4)); // true
→ System.out.println(s1.equals(s2)); // false
```



Et pour cela, il y a un traitement spécifique pour comparer les Strings. Pour comparer, effectivement, deux chaînes de caractères, on dispose de la fonction « equals ». Par exemple, pour comparer deux variables « chaîne 1 » et « chaîne 2 », je peux appeler ma fonction « equals » de cette façon-ci. Donc tout d'abord, le nom de la première variable, suivi d'un point, suivi du nom de la fonction « equals », et suivi, en argument à la fonction « equals », de la deuxième variable qu'on veut comparer. Cette fonction « equals » va renvoyer soit « true », si « chaîne 1 » et « chaîne 2 » correspondent à la même chaîne de caractère, et n'ont pas forcément la même référence, et « false », si les deux chaînes sont différentes. Par exemple, j'ai ici, déclaré, une variable « s1 » qui contient une référence vers une chaîne "abc", une variable « s2 » qui contient une référence vers une chaîne "abc", mais avec « b » en lettre capitale, et ici, une variable « s4 », qui est initialisée au résultat de la concaténation de « s1 » et de la chaîne vide. « s1 » correspond à la chaîne "abc". Quand je concatène "abc" avec la chaîne vide, j'obtiens la chaîne, tout simplement, "abc". Et « s4 » va donc contenir une référence vers cette chaîne "abc". On avait vu dans un transparent précédent que les références contenues par « s1 » et « s4 » étaient différentes, donc je ne peux pas comparer « s1 » et « s4 » directement en utilisant le symbole « = ». Par contre, je peux utiliser la fonction « equals ». C'est ce que je fais ici, je vais comparer « s1 » et « s4 » en appelant la fonction « equals »

notes

résumé

0m 13s



Comparaison de String

→ true si égales
→ false si différentes

chaîne1.equals(chaîne2) teste si les chaînes de caractères référencées par chaîne1 et chaîne2 sont constituées des mêmes caractères

```
→ String s1 = "abc";  
→ String s2 = "aBc";  
→ String s4 = s1 + "";  
  
→ System.out.println(s1.equals(s4)); // true  
→ System.out.println(s1.equals(s2)); // false
```



et je peux vérifier que cette fois-ci, j'obtiens bien « true » Parce que les chaînes sont exactement les mêmes, les caractères "abc" sont les mêmes. Si maintenant, je teste si « s1 » est égal à « s2 », et bien il se trouve que la fonction « equals »

notes

résumé

notes

résumé

2m 37s

