

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-01-3-Variables-pt5

Concepts (extraits des sous-titres générés automatiquement) :

Notion de constante. Dernier volet. Valeur d'une variable. Vitesse de la lumière. Nommage des constantes. Convention usuelle de nommage. Long de l'exécution du programme. Valeur de base. Moyen d'une affectation. Déclaration de la variable. Valeur de cette variable. Erreur de compilation. Exemple. Cas. Conventions.



[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/>

Variables

(Partie 5)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



Déclaration de constantes

Il peut arriver que la valeur d'une variable ne doive pas changer après l'initialisation. Dans ce cas, il faut ajouter le mot-clé `final` devant la déclaration:

```
final type identificateur = valeur_initiale;
```

Par exemple:

```
final double VITESSE_DE_LA_LUMIERE = 299792.458;
```

Dans ce cas, on ne peut plus modifier la variable:

```
VITESSE_DE_LA_LUMIERE = 100; // erreur !
```

Dernier volet lié aux fondamentaux sur les variables, la notion de constante. Nous avons vu que l'affectation nous permet de modifier la valeur d'une variable après sa déclaration. Il existe cependant des situations où ceci n'est pas souhaitable. Par exemple, j'ai un programme qui veut travailler avec la vitesse de la lumière. On sait que cette vitesse de la lumière est constante, tout au long de l'exécution du programme, ne va pas changer. Donc on va préciser que cette donnée ne peut pas changer. Dans ce cas là, il faut le faire en précédant la déclaration

notes

résumé

0m 1s



Déclaration de constantes

Il peut arriver que la valeur d'une variable ne doive pas changer après l'initialisation. Dans ce cas, il faut ajouter le mot-clé `final` devant la déclaration:

```
final type identificateur = valeur_initiale;
```

Par exemple:

```
→ final double VITESSE_DE_LA_LUMIERE = 299792.458;
```

Dans ce cas, on ne peut plus modifier la variable:

```
VITESSE_DE_LA_LUMIERE = 100; // erreur !  
Completé
```

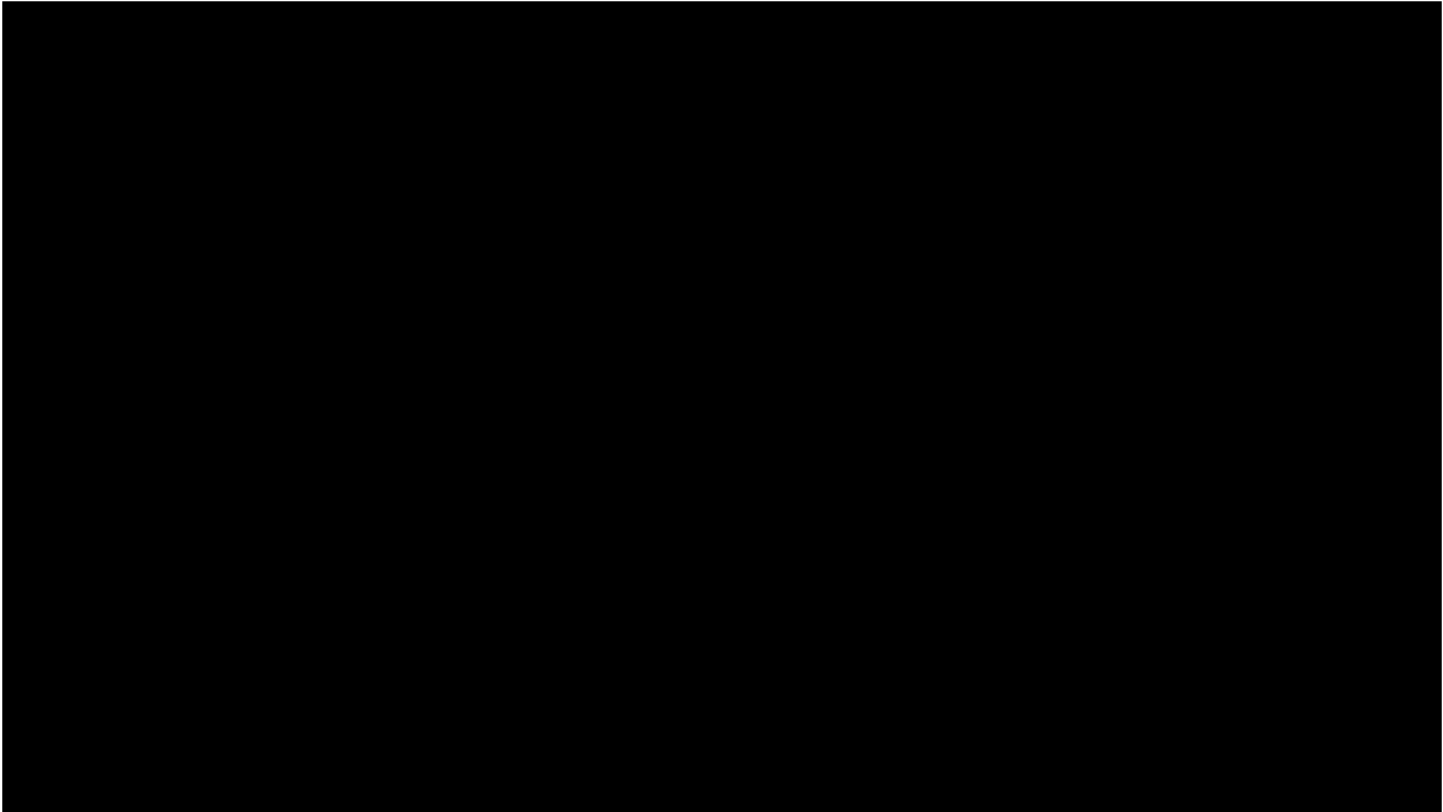
de la variable en question par le mot réservé final, qui veut dire, qu'une fois une valeur de base attribuée, on ne peut plus modifier la valeur de cette variable. Donc ici par exemple, si j'ai déclaré ma constante comme ceci, la constante VITESSE_DE_LA_LUMIERE, si par la suite, j'essaie d'en altérer la valeur au moyen d'une affectation, j'aurai une erreur de compilation.

notes

résumé

0m 37s





Vous noterez au passage qu'il existe des conventions pour le nommage des constantes en java, ici l'identificateur de ma constante `VITESSE_DE_LA_LUMIERE` est constitué de différents mots, tout s'écrit en majuscule et séparé par des soulignés (`_`). Il s'agit là d'une convention usuelle de nommage pour les constantes, à laquelle je vous recommande d'adhérer. à laquelle je vous recommande d'adhérer.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

1m 1s



.....

.....

.....

.....