

Les conditions if / else / else if / switch case en JAVA

Date de dernière mise à jour : 27/06/2007 à 19:36

Source : <http://www.vulgarisation-informatique.com/java-conditions.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Les conditions :

Une condition va vous permettre d'exécuter une portion de code ou non en fonction du résultat de variables booléennes, c'est à dire que vous pourrez dire "si X est faux alors je fais ça, sinon ceci et si aucune des conditions précédentes n'est remplie, je ferais plutôt cela".

Tout ça se code en JAVA avec différentes instructions. Les plus courantes sont les instructions if / else.

Les instructions if / else :

L'instruction if se traduit en français par "si". Elle va vous permettre d'effectuer une action si une condition est vraie ou fausse :

```
if
(
condition)
{
    //code
}
```

Exemple d'instruction if :

```
public class
Test{
    public int
variable =
20;    public
Test(
)
{
    if
(
variable =
=
20)
    {
        System.out.println(
&quot;La variable 'variable' est bien égale à 20&quot;
)
    }
}
}
```

Dans notre exemple, on affichera bien le texte car la variable est bien égale à 20. Notez que l'on utilise l'opérateur == pour les comparaisons de données avec des types primitifs. Le simple signe égal est un signe d'affectation.

L'instruction else se traduit en français par "sinon". Elle va vous permettre d'exécuter une action si la première condition située dans le

"if", n'est pas réalisée. Voici un exemple dans lequel on affichera que la variable n'est pas égale à 20

Exemple d'instruction else :

```
public class
Test{
    public int
variable =
15;    public
Test(
)
    {
        if
(
variable =
=
20)
        {
            System.out.println(
&quot;La variable 'variable' est bien égale à 20&quot;;
)
        }
        else
        {
            System.out.println(
&quot;La variable 'variable' n'est pas égale à 20&quot;;
)
        }
    }
}
```

L'instruction else if :

Voici un exemple d'instruction else if qui sera je pense plus explicite qu'un long discours :

```
public class
Test{
    public int
variable =
15;    public
Test(
)
    {
        if
(
variable =
=
20)
        {
            System.out.println(
&quot;La variable 'variable' est bien égale à 20&quot;;

```

```

)
; }
else
if
(
variable =
=
15)
{
System.out.println(
"La variable 'variable' n'est pas égale à 20 et est égale à 15");
)
; }
else
if
(
variable =
=
10)
{
System.out.println(
"La variable 'variable' n'est pas égale à 20 ni à 15 et est égale à 10");
)
; }
else
{
System.out.println(
"La variable 'variable' n'est ni égale à 20 ni à 15 ni à 10");
)
; }
}
}

```

L'instruction Switch / case :

L'instruction Switch est utile quand vous devez gérer beaucoup de if / else if / else. Elle a une syntaxe plus courte et est plus appropriée pour ce type de cas.

Fonctionnement :

```

switch
(
variable)
{
case
'valeur1':      action1;      break
; case
'valeur2':      action2;      break
; case
'valeur3':'valeur4':  action3;      break
; d

```

```

efault
:      action4;      break
;    }

```

Ici dans notre exemple de fonctionnement de cette instruction, on remarque la ligne **case 'valeur3' : 'valeur4' :** . Cette ligne veut dire que si la variable **variable** vaut soit 'valeur3' soit 'valeur4', alors on exécutera "action4". Vous pouvez mettre autant de valeurs que vous souhaitez. Si aucune valeur ne correspond, la ou les instructions contenues dans le bloc **default** s'exécute(nt).

Voici notre exemple précédent (basé sur des else if) traduit avec l'instruction switch / case :

```

public class
Test{
    public int
variable =
15;    public
Test(
)
{
    switch
(
variable)
    {
        case
20 :      System.out.println(
&quot;La variable 'variable' est bien égale à 20&quot;
)
;      break
;      case
15 :      System.out.println(
&quot;La variable 'variable' n'est pas égale à 20 et est égale à 15&quot;
)
;      break
;      case
10 :      System.out.println(
&quot;La variable 'variable' n'est pas égale à 20 ni à 15 et est égale à 10&quot;
)
;      break
;      default
:      System.out.println(
&quot;La variable 'variable' n'est ni égale à 20 ni à 15 ni à 10&quot;
)
;      break
;    }
}
}

```

La négation :

La négation s'utilise quand on souhaite par exemple dire "si telle variable n'est pas égale à". On peut utiliser l'opérateur != (qui veut dire "différent de".) mais on peut utiliser aussi tout simplement le point d'exclamation, qui veut dire "non". Attention, le point

d'exclamation s'applique à une variable booléenne (true ou false), vous devrez donc mettre parfois des parenthèses. Exemple : **public class**

```
Test{
    public int
variable =
15;    public
Test(
)
{
    if
(
!(
variable =
=
20)
)
{
    System.out.println(
"La variable 'variable' n'est pas égale à 20");
}
}
}
```

Source : <http://www.vulgarisation-informatique.com/java-conditions.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)