

Introduction au programmation Scilab

Professeur ESSADDOUKI Mostafa (essaddouki@gmail.com)
<http://www.developpement-informatique.com>

10 mai 2018

1 Opérateurs de comparaison.

<code>a&b</code>	logical and
<code>a b</code>	logical or
<code>~a</code>	logical not
<code>a==b</code>	true if the two expressions are equal
<code>a~=b</code> or <code>a<>b</code>	true if the two expressions are different
<code>a<b</code>	true if a is lower than b
<code>a>b</code>	true if a is greater than b
<code>a<=b</code>	true if a is lower or equal to b
<code>a>=b</code>	true if a is greater or equal to b

2 Structure conditionnelle IF

L'instruction if permet d'exécuter une instruction si une condition est satisfaite. If utilise une variable booléenne pour effectuer son choix : si le booléen est vrai, alors l'instruction est exécutée. Une condition est fermée lorsque le mot clé end est rencontré.

```
1 if ( %t ) then
2     disp("Hello_!")
3 end
```

Si la condition n'est pas satisfaite, l'instruction else permet d'exécuter une instruction alternative, comme dans le script suivant.

```
1 if ( %f ) then
2     disp("Hello_!")
3 else
4     disp("Goodbye_!")
5 end
```

Lorsque nous devons combiner plusieurs conditions, l'instruction elseif est utile.

```
1 i=2
2 if ( i == 1 ) then
3     disp("Hello_!")
4 elseif ( i == 2 ) then
5     disp("Goodbye_!")
6 elseif ( i == 3 ) then
7     disp("Tchao_!")
8 else
9     disp("Au_Revoir_!")
10 end
```

2.1 Exemple 1

Ecrire un programme qui permet d'afficher le message « non admis » si la note de l'étudiant est inférieure à 10 et admis dans le cas où la note est supérieure à 10

```

1 note=input("saisir_une_note")
2 if (note<10) then
3     disp("non_admis")
4 else
5     disp("admis")
6 end

```

2.2 Exemple 2

Ecrire un programme qui permet de tester le signe d'un nombre ;

```

1 nb=input("saisir_un_nombre")
2 if (nb<0) then
3     disp("nombre_est_negatif")
4 elseif (nb>0) then
5     disp("nombre_est_positif")
6 else
7     disp("nombre_est_nul")
8 end

```

2.3 Exemple 3

Écrire un programme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- « Poussin » de 6 à 7 ans
- « Pupille » de 8 à 9 ans
- « Minime » de 10 à 11 ans
- « Cadet » après 12 ans

```

1 age=input("saisir_l'age_de_")
2 if(age>=6 & age <=7) then
3     disp("Poussin")
4 elseif (age>=8 & age <=9) then
5     disp("Pupille")
6 elseif (age>=10 & age <=11) then
7     disp("Minime")
8 elseif (age>=12) then
9     disp("Cadet")
10 else
11     disp("age_n'appartient_a_aucune_categorie")
12 end

```

```

1 age=input("saisir_l'age_de_")
2 if(age<6) then
3     disp("age_n'appartient_a_aucune_categorie")
4 elseif (age <=7) then
5     disp("Poussin")
6 elseif (age <=9) then
7     disp("Pupille")
8 elseif (age<=11) then
9     disp("minime")
10 else
11     disp("cadet")

```

```
12 end
```

2.4 Exemple 4

Ecrivez un programme qui lira au clavier l'heure et les minutes, et il affichera l'heure qu'il sera une minute plus tard. Par exemple, si l'utilisateur tape 21 puis 32, l'algorithme doit répondre : "Dans une minute, il sera 21 heure(s) 33".

NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.

```
1 h=input("saisir_l'heure")
2 m=input("saisir_la_minute")
3 if(h==23) then
4     if(h==59) then
5         h=0
6         m=0
7     else
8         m = m+1
9     end
10 else
11     if (m==59) then
12         h=h+1
13         m= 0
14     else
15         m=m+1
16     end
17 end
18 disp("Dans_une_minute,_il_sera",h,'_heure(s)',m)
```

3 SELECT

L'instruction select permet de combiner plusieurs branches de manière claire et simple.

En fonction de la valeur d'une variable, cela permet d'effectuer l'instruction correspondant au mot clé case. Il peut y avoir autant de branches que nécessaire.

```
1 i=2
2 select i
3 case 1
4     disp("One")
5 case 2
6     disp("Two")
7 case 3
8     disp("Three")
9 else
10     disp("Other")
11 end
```

3.1 Exemple

Écrire un programme qui à partir d'un nombre compris entre 1 et 7 affiche le jour correspondant ?

```
1 i=input("saisir_un_nombre")
2 select i
3   case 1
4     disp("Lundi")
5   case 2
6     disp("Mardi")
7   case 3
8     disp("Mercredi")
9   case 3
10    disp("Jeudi")
11   case 3
12    disp("Vendredi")
13   case 3
14    disp("Samedi")
15   case 3
16    disp("Dimanche")
17 else
18   disp("numero_invalid")
19 end
```