



Relations d'ordre et logiques

>	>=	<	<=	==	~=	&&	
Plus grand	Plus grand ou égale	Plus petit	Plus petit ou égale	Égale	Différent	Et	Ou

X	~X	Y	X Y	X&&Y	xor(X,Y)
True	False	True	True	True	False
False	True	True	True	False	True
False	True	False	False	False	False

MATLAB et Octave pour débutants

Pour pouvoir contrôler le flux des opérations dans un algorithme, il faut pouvoir vérifier les conditions et agir en conséquence. Pour vérifier si certaines conditions sont remplies ou pas, il existe les mots clés if, else, littéralement, si ou autrement. Si une condition est remplie alors on fait certaines commandes, autrement on en fait d'autres. Les conditions se vérifient à l'aide d'opérations logiques comme plus grand, égal, différent, etc. Il faut aussi connaître l'ordre de priorité entre les opérateurs pour être sûr que le programme s'exécute comme on le souhaite. Les opérations logiques plus grand, égal, et cetera ont comme résultat, vrai ou faux. On peut donc comparer des nombres ou aussi associer des relations, par exemple vérifier A plus grand que B et B plus grand que 0. Les opérations logiques se résument ainsi. Alors majeur, le symbole majeur pour plus grand. Le symbole majeur égal pour plus grand égal plus petit avec le symbole mineur, et cetera. Ensuite on pourra dire, si quelque chose est égal, il faut utiliser le double symbole égal, autrement on a une opération de définition donc par exemple A est égal à 3, on ne va pas rendre si c'est vrai ou faux que A est égal à 3 mais ne va pas simplement dire que A doit être égal à 3.

Notes

Summary



0m 04s

Relations d'ordre et logiques

>	>=	<	<=	==	~=	&&	
Plus grand	Plus grand ou égale	Plus petit	Plus petit ou égale	Égale	Différent	Et	Ou

X	~X	Y	X Y	X&&Y	xor(X,Y)
True	False	True	True	True	False
False	True	True	True	False	True
False	True	False	False	False	False

MATLAB et Octave pour débutants

Donc il faut utiliser le double symbole égal pour vérifier l'égalité entre 2 choses, différence avec le tild, égal, ensuite "et" ou "ou" avec double symbole de en percent (%) ou la barre verticale. Il y a aussi un tableau de vérité qui nous aide à comprendre quand est-ce que les conditions sont remplies ou pas. Par exemple, si X est vrai, alors non X, le tild X ça veut dire, c'est la négation de X, alors c'est faux et naturellement si X est faux, la négation de X est vrai. Ensuite, si on prend 2 variables X et Y, si les 2 sont vraies, alors X ou Y est vrai et X et Y est vrai aussi. Par contre le xor, c'est quelque chose qui est vrai seulement si seulement une des 2 est vraie et donc xor est vrai ou faux, c'est vrai mais dans ce cas, dans la première ligne, xor de vrai virgule vrai, c'est faux. C'est pour ça que la première ligne, quand X et Y sont vrais, xor devient faux et ensuite on peut lire la table de vérité, si X faux et Y est vrai, alors X ou Y c'est vrai et X et Y est faux et le xor comme on voit, est vrai. Si X et Y sont faux, alors X ou Y c'est faux aussi, X et Y c'est faux, et xor de X et Y c'est faux aussi.

Notes

Summary



1m 24s

Priorités

Definition	Ou	Et	Relation	2 points	Addition Substraction	Multiplication Division	Négation	Transposé Puissance	Paranthèse
=		&&	<<= == >>= ~=	:	+ -	* / \	~	' ^	()

MATLAB et Octave pour débutants

La priorité des opérations nous permet de simplifier l'écriture sans toujours mettre des parenthèses. Bien sûr, les parenthèses sont toujours prioritaires. Ensuite, il y a la transposée et la puissance, la négation, la multiplication et division, ainsi de suite jusqu'à la définition. Donc ce tableau nous résume de la droite jusqu'à la gauche les priorités dans les opérations.

Notes

Summary



2m 55s

if – elseif – else

```
>> edit iftest.m
>> iftest(3,4)
a est plus petit que b
>> iftest(4
```

Command Window

```
1 function [] = iftest(a, b)
2 % iftest verifie si a plus petit que b
3
4 if (a < b )
5     disp('a est plus petit que b')
6 end
7
8
9
```

GUI

MATLAB et Octave pour débutants

Voyant comment utiliser le "si" autrement. Alors on va construire une petite fonction simple qu'on va appeler (if test). Alors on a dit qu'on veut définir une fonction, donc c'est dans ce qu'on a fait la semaine passée, il faut dire, commencer avec le mot clé, fonction, dire quelles sont les sorties. Cette fois-ci, je n'ai pas envie d'avoir de sortie donc je dis, il n'y a pas de sortie. Ensemble vide. Et ensuite je dois appeler la fonction comme le fichier, if test, et dire quelles sont les entrées, ici je vais prendre 2 nombres et je vais les comparer. Je peux écrire maintenant le hub de cette fonction, ça on sait déjà ce qu'on va faire, alors, if test vérifie si A est plus petit que B. Première chose à faire, c'est de vérifier avec la commande "if", A plus petit que B. Maintenant je dois faire plusieurs choses, par exemple, je peux juste if écrire, A est plus petit que B. Je dois terminer le bloc de ce "if", ma fonction est faite. Qu'est-ce qu'elle va faire ? Je sauvegarde et j'exécute, if test, alors c'est une fonction donc je dois l'appeler avec des arguments, par exemple, 3 virgule 4 et elle me dit, A est plus petit que B. Alors qu'est-ce qui se passe si je fais 4-3.

Notes

Summary



3m 19s

if – elseif – else

```
>> edit iftest.m
>> iftest(3,4)
a est plus petit que b
>> iftest(4,3)
>> iftest(3,4)
a est plus petit que b
3 < 4
>> |
```

Command Window

GUI

```
1 function [] = iftest(a, b)
2 % iftest verifie si a plus petit que b
3
4 if (a < b) % ici commence le bloc des commandes
5 % a executer si la condition est remplie
6
7     disp('a est plus petit que b')
8     disp([num2str(a) ' < ' num2str(b)])
9
10 end % termine le bloc du if
11
12
13
```

MATLAB et Octave pour débutants

Il n'y a rien qui se passe et effectivement, cette fonction nous dit, il n'y a rien qui doit se faire quand le nombre A est plus grand ou égal à un nombre B. Alors je peux aussi écrire plusieurs choses dans cette partie de code, entre le "if" et le "end", ça s'appelle un bloc. Donc ici, commence le bloc des commandes à exécuter si la condition est remplie. Et "end" termine le bloc du "if". Donc ça veut dire en gros, on peut écrire entre le "if" et le "end" plusieurs commandes qui peuvent être montrées à l'écran. Par exemple, je peux vouloir aussi écrire exactement la condition, A plus petit que B, donc je vais transformer A dans une chaîne de 40 caractères avec la commande num to string (num2str), ensuite je vais mettre le symbole "plus petit" et ensuite aussi transformer le nombre B en chaîne de caractères. Je lie tout ça dans une seule chaîne de caractères, c'est pour ça qu'il y a les parenthèses carrées et je peux sauvegarder et maintenant exécuter ma fonction et 3 plus petit que 4 apparaît.

Notes

Summary



5m 08s

if – elseif – else

```
>> edit iftest.m
>> iftest(3,4)
a est plus petit que b
>> iftest(4,3)
a est plus petit que b
>> iftest(3,4)
3 < 4
>> iftest(3,4)
a est plus petit que b
3 < 4
Les deux nombres sont positifs
>> iftest(4,3)
Les deux nombres sont positifs
>> iftest(-3,-4)
>> iftest(-3,-4)
a est plus grand ou egale a b
>>
```

Command Window

GUI

```
1 function [] = iftest(a, b)
2 % iftest verifie si a plus petit que b
3
4 if ( a < b )
5     disp('a est plus petit que b')
6     disp([num2str(a) ' < ' num2str(b)])
7 else % bloc a executer si a >= b :
8     disp('a est plus grand ou egale a b')
9
10 end
11
```

MATLAB et Octave pour débutants

Alors les espaces à l'intérieur de la parenthèse ne sont pas si importants que ça, on peut aussi rajouter d'autres blocs et d'autres conditions "if", alors maintenant je peux dire, si A est plus grand que 0, et aussi B est plus grand que 0, donc c'est entre les 2, je dois mettre les symboles du "et" donc c'est double ampersand, alors je vais exécuter une autre liste de commande. Ici, je veux simplement dire, les 2 nombres sont positifs. Maintenant, si j'exécute avec 3 et 4, j'ai les 3 phrases qui s'impriment, si je fais avec 4-3, alors les 2 nombre sont positifs, c'est affiché à l'écran. Mais si je prends les nombres moins 3 moins 4, il n'y a rien qui s'affiche. Est-ce que je peux rajouter quelque chose à faire si une des conditions n'est pas remplie ? Alors maintenant, je voudrais imprimer quelque chose à l'écran dans le cas où A n'est pas plus petit que B. Alors ce que je peux faire, c'est de rajouter un bloc à exécuter dans le cas contraire. Donc d'abord if A plus petit que B, alors je fais quelque chose autrement, ça c'est bloc à exécuter. Si A est plus grand ou égal à B. Alors là, on peut dire que A est plus grand ou égal à B. la fonction, elle va afficher quelque chose même dans le cas où A est plus grand que B.

Notes

Summary



6m 57s

if – elseif – else

```
>> edit iftest.m
>> iftest(3,4)
a est plus petit que b
>> iftest(4,3)
a est plus petit que b
>> iftest(3,4)
3 < 4
>> iftest(3,4)
a est plus petit que b
3 < 4
Les deux nombres sont positifs
>> iftest(4,3)
Les deux nombres sont positifs
>> iftest(-3,-4)
>> iftest(-3,-4)
a est plus grand ou egale a b
>> iftest(-3,-3)
a est egale a b
>> iftest(-3,-4)
a est plus grand que b
>>
```

Command Window

GUI

```
1 function [] = iftest(a, b)
2 % iftest verifie si a plus petit que b
3
4 if ( a < b )
5     disp('a est plus petit que b')
6     disp([num2str(a) ' < ' num2str(b)])
7 elseif ( a == b )
8     disp('a est egale a b')
9
10 else % bloc a executer si a >= b :
11     disp('a est plus grand que b')
12
13 end
14
```

MATLAB et Octave pour débutants

On peut aussi rajouter au milieu entre les "if" et le "else" d'autres conditions qui s'exécutent si la première n'est pas remplie. Donc par exemple, j'aimerais dire que si A est égal à B, alors dans ce bloc, on va simplement écrire A est égal à B. Et du coup. Ce qui reste, la condition qui reste si A n'est pas plus petit que B, il n'est pas égal à B, alors c'est juste. A est plus grand que B. Donc je peux exécuter mon test en disant, moins 3 moins 3, A est égal à B, ou encore moins 3 moins 4, A est plus grand que B donc on a la possibilité d'utiliser des conditions avec "if" et cette condition, on peut la compléter avec "else if" ou "else", et chacun de ces blocs est exécuté selon la condition A plus petit que B ou la condition qui est donnée dans le "if".

Notes

Summary



Switch – case – otherwise

```
>> edit switchTest.m
>> |
```

Command Window

```
1 function [grade] = switchTest (points)
2 % switchTest rend la note correspondante au nombre des points
3
4 if (points < 0 ) || ( points > 10)
5     grade = 'NA';
6     return
7
8
9 endfunction
10
```

MATLAB et Octave pour débutants

Pour contrôler ce qui se passe dans un programme, on peut aussi utiliser le mot clé "switch". Alors comment marche ce "switch"? Je fais encore une fonction, cette fois-ci, je l'ai déjà préparée, alors je veux construire une fonction qui s'appelle "switch test", et donner un nombre de points, il rend la note correspondante. D'abord on va contrôler que le nombre de points est positif et plus petit que 10. Donc si le nombre de points est négatif, ou, alors le "ou", on a dit, il faut utiliser cette barre verticale. Pour le faire, moi je dois faire Alt 7. Alors ça dépend de votre clavier. Donc si les points sont négatifs ou les points sont plus que 10, alors en fait, il y a un problème dans l'appel de cette fonction, alors la note, on va dire qu'elle n'est pas définie, NA dit, c'est à dire non acquis. Et puis un n'arrête tout de suite la fonction, ça, on ne l'a pas encore vu. Si on veut arrêter tout de suite l'exécution d'une fonction, on peut utiliser le mot clé Return, ça a du sens seulement depuis qu'on sait faire des conditions, s'il n'y a pas de conditions, une fois qu'on appelle Return, c'est comme dire, voilà, c'est fini. Et maintenant, ici on finit tout de suite si le nombre de points est négatif ou plus grand que 10.

Notes

Summary



11m 23s

Switch – case – otherwise

```
>> edit switchTest.m
>> |
```

Command Window

```
1 function [grade] = switchTest (points)
2 % switchTest rend la note correspondante au nombre des points
3
4 if (points < 0 ) || ( points > 10)
5     grade = 'NA';
6     return
7 end
8
9 switch points
10 case {9,10}
11     grade = '6';
12 case 8
13     grade = '5.5';
14 case 7
15     grade = '5';
16 otherwise
17     grade = '3a'
18
19 endfunction
20
```

MATLAB et Octave pour débutants

Et maintenant, je termine le bloc lié à ce "if". Alors maintenant, on sait que les points, ils sont entre 0 et 10. Maintenant, j'utilise le mot clé "switch". Alors, ce "switch", qu'est-ce qu'il fait? Il regarde si une variable, ici la variable point, elle est dans un certain cas, qu'on appelle "case". Alors le premier cas, c'est par exemple si j'ai 9 ou 10 points. Alors la note va être A, on obtient la meilleure note ou si vous voulez, (en Suisse) on utilise la note 6 pour la meilleure note, ainsi vous décidez ce que vous voulez. Ensuite le cas où on a 8 points. Vous voyez, avant, j'avais 2 cas, 9 et 10, je les ai mis entre accolades, maintenant j'ai un seul cas, alors il me suffit de dire le cas 8, alors dans ce cas, on va obtenir la note 5 virgule 5. Ici, c'est rendu sous forme d'une chaîne de caractères, on pourrait aussi décider que c'est sous forme de noms. Donc à chaque fois, j'ai le droit de faire un petit bloc, exécuter, un bloc ça veut dire, on peut mettre plusieurs commandes à la suite, alors 7, on va dire 25, 25 sur 6, et en fait maintenant, on trouve que nos étudiants, ils doivent toujours avoir au moins 5 de moyenne, sinon, on n'est pas content, alors si dans tout autre cas, on va dire que la note est égale à 3.

Notes

Summary



Switch – case – otherwise

```
>> edit switchTest.m
>> switchTest(9)
ans = 6
>> switchTest(-2)
ans = NA
>> switchTest(7)
ans = 5
>> |
```

Command Window

```
1 function [grade] = switchTest (points)
2 % switchTest rend la note correspondante au nombre des points
3
4 if (points < 0 ) || ( points > 10)
5     grade = 'NA';
6     return
7 end
8
9 switch points
10 case {9,10}
11     grade = '6';
12 case 8
13     grade = '5.5';
14 case 7
15     grade = '5';
16 otherwise
17     grade = '3';
18 end
19
20 end
21
```

GUI

MATLAB et Octave pour débutants

Si on n'a pas fait au moins 7 points, on ne mérite pas la moyenne un peu sévère. Maintenant, j'ai fini tous mes cas donc le "switch" est terminé, je dois fermer le bloc de "switch" avec "end" et puis terminer ma fonction. Maintenant, je vais utiliser ma fonction. Si j'ai fait 9 points, très bien. Si j'ai fait moins de points, ce n'est vraiment pas bon, ce n'est pas acquis du tout. Si j'ai fait 7 points, j'ai la note 5.

Notes

Summary

16m 10s



4.2 Exécution conditionnelle



- Sélection de code à exécuter par des conditions
- If- elseif-else
- switch

MATLAB et Octave pour débutants

Le flux des opérations peut se contrôler avec les mots clés, "if", "else" ou encore "switch" qui est moins sophistiqué et parfois d'utilisation plus claire. Ce sont les premiers éléments de programmation. L'étape suivante est l'utilisation des boucles afin de répéter un bloc de code plusieurs fois.

Notes

Summary



17m 06s