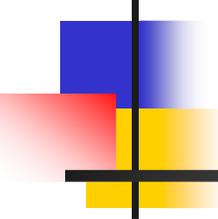
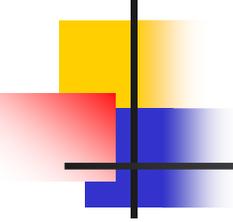


UNIX

- Système d'exploitation
- Concepts élémentaires UNIX
- Commandes Unix



Systeme d'exploitation



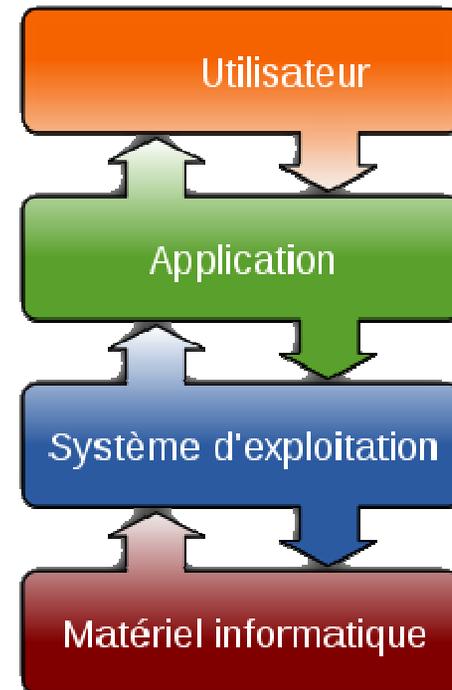
Systeme d'exploitation

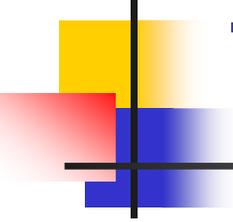
- DOS (1981), Windows (1995)
- UNIX (1969), Linux (1991)
- Systeme1 (1984) ... Systeme7, Mac OS

Systeme d'exploitation

- Ensemble de programmes responsables de la liaison entre

- les applications informatiques de l'utilisateur
- les ressources matérielles d'un ordinateur





Tâches à différents niveaux

Interface utilisateur

- interpréteur de commandes
- interface graphique

Gestion des périphériques

- pilotes

Gestion de la mémoire secondaire

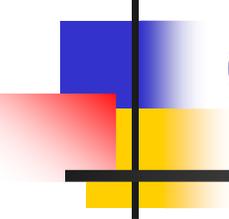
- structuration des données (fichiers)

Gestion de la mémoire principale

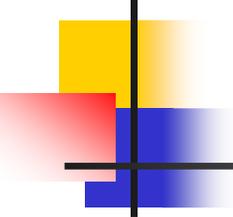
- noyau et l'espace utilisateur

Noyau

- gestion de processus
- allocation de CPU
- gestion des interruptions

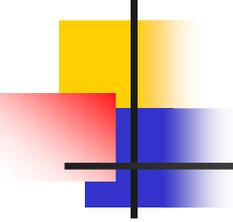


Concepts élémentaires UNIX



UNIX

- 1969, laboratoires Bell
- Quatre concepts élémentaires:
 - Fichiers
 - Droits d'accès
 - Processus
 - Communications inter-processus (IPC)
- Système multi-tâches multi-utilisateurs

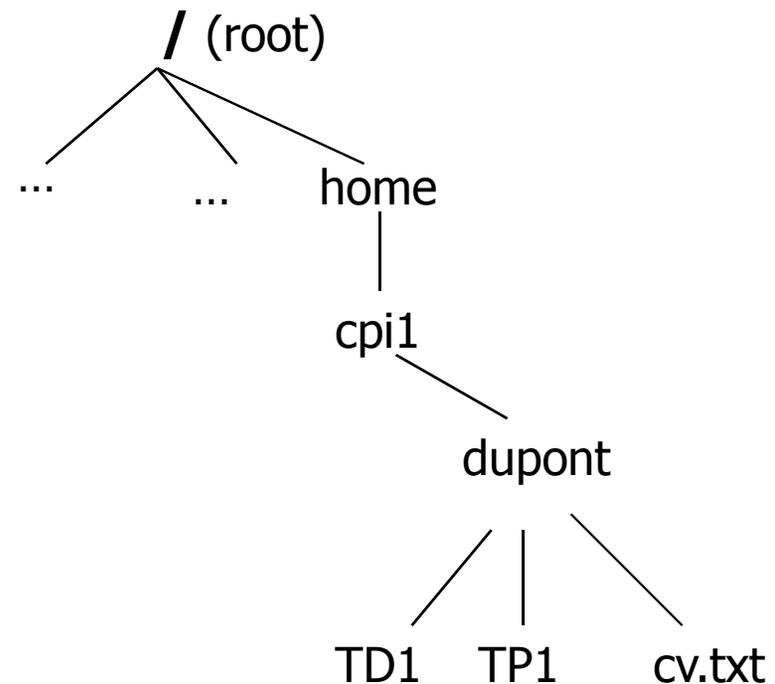


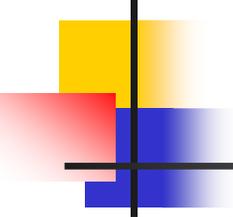
Fichiers

- Unité élémentaire de gestion de ressources
- Référencé dans un système de fichiers par :
 - propriétaire
 - groupe
 - droits d'accès
 - taille
 - date de dernière modification
 - références des blocs de données sur le disque

Arborescence des fichiers

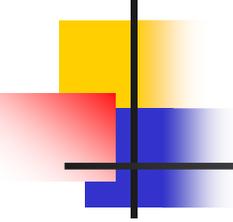
- Dans un répertoire :
 - sous-répertoires
 - fichiers
- Chemin absolu
 - /home/cpi1/dupont
- Chemin relatif
 - /home/cpi1/dupont/TP1
 - cd ../TD1
 - /home/cpi1/dupont/TD1
- Répertoire de connexion
 - /home/cpi1/dupont
 - ~





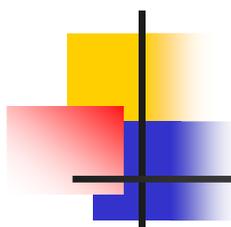
Répertoires importants

- / : root
- /bin : commandes de base pour tous les utilisateurs
- /dev : fichiers pilotant les périphériques
- /etc : fichiers de configuration (fichiers de démarrage)
- /lib : bibliothèques compilées
(nécessaires pour les commandes dans /bin)
- /tmp : fichiers de travail temporaires
- /usr/bin : commandes complémentaires de l'utilisateur
- /usr/include : fichiers d'en-tête des langages
- /home : fichiers des utilisateurs
- /var : contient des données diverses, telles que la boîte mail, des fichiers temporaires et des fichiers journaux



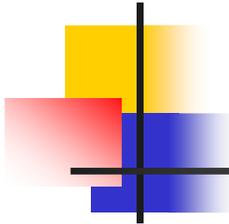
Droits d'accès

- Toute ressource (fichier, répertoire, ...) a
 - un identificateur
 - un propriétaire
 - un ensemble de droits d'accès (en lecture, en écriture, en exécution)
 - les droits du propriétaire
 - les droits du groupe auquel appartient le propriétaire
 - les droits des autres utilisateurs
- Root, administrateur



Droits d'accès : exemple

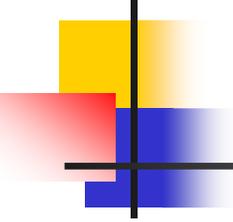
- fichiers : *-rwxr----- nom1*
- répertoires : *drwxr----- nom2*
- droits affichés pour
 - utilisateur : Dupont
 - groupe : CPI1
 - tout le monde



Processus

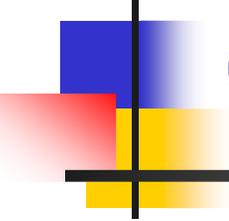
- Unité élémentaire de gestion des traitements
 - un ensemble d'instructions à exécuter
 - un espace mémoire pour les données de travail
 - d'autres ressources, comme des descripteurs de fichiers, des ports réseau, etc.

Threads (« processus légers »): fils d'exécution qui partagent ressources et données

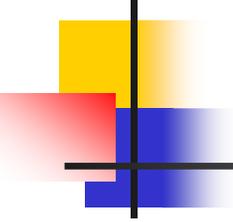


Communication inter-processus

- Etablir une communication entre deux processus via
 - signaux
 - mémoire partagée
 - tube (« pipe »)
 - sockets
- Arbitrer l'utilisation de ressources partagées entre différents processus
 - sémaphores, mutex ...

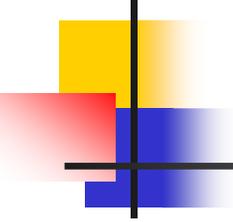


Commandes Unix



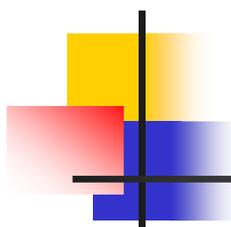
Syntaxe des commandes

- `nom_commande [options] [arguments]`
 - options :
 - `ls -l -a`
 - `ls -la`
 - `ls -l --color` (option à plusieurs caractères)
 - arguments : fichier, expression
 - `grep toto monFichier`
 - `tar -cv -f archive.tar MonRepertoire`



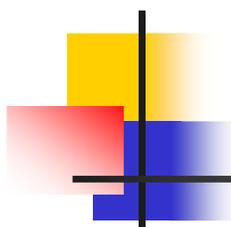
Méta-caractères

- * : suite de caractères
- ? : un seul caractère
- []: un des caractères dans les crochets
 - un ensemble : [hg]
 - un intervalle : [a-k]



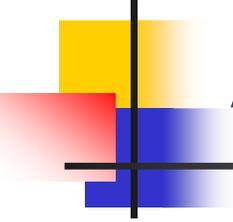
Méta-caractères : exemples

- *ls **
- *ls *.exe*
- *ls ????*
- *ls [ct]**



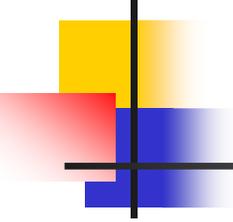
Méta-caractères : exemples

- *ls **
 - afficher le contenu du répertoire courant
- *ls *.exe*
 - afficher tous les fichiers se terminant par .exe
- *ls ????*
 - afficher tous les fichiers dont le nom est composé de 4 caractères exactement
- *ls [ct]**
 - afficher tous les fichiers dont le nom commence par c ou par t



Aide

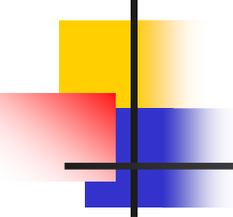
- *man commande*
 - obtenir le manuel d'une commande
- *commande --help*
 - afficher une aide succincte (aide mémoire) et liste les arguments qui peuvent être passés à *commande*



Manipulation des fichiers : chemin

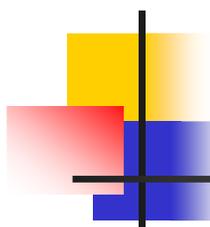
- ***pwd***
 - afficher le répertoire courant

- ***cd chemin***
 - changer le répertoire courant
 - sans argument : retour au répertoire de connexion
 - `cd ~` : retour au répertoire de connexion



Manipulation des fichiers : affichage

- ***ls [-al] chemin1 chemin2 ...***
 - *chemin_i*
 - fichier : afficher description
 - répertoire : afficher contenu
 - options
 - -a : tous les fichiers (aussi les fichiers cachés!)
 - -l : format long (taille, date, droits, etc.)



Manipulation des fichiers : changement de droit

- ***chmod mode fichier***

- changer les droits d'accès au fichier.

- ***chmod a+r fichier***

- tout le monde (« a ») a le droit de lire.

- ***chmod og-w fichier***

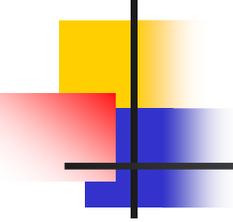
- les autres (« o ») et le groupe (« g ») n'ont pas le droit d'écrire.

- ***chmod u+x fichier***

- l'utilisateur (« u ») a le droit d'exécuter.

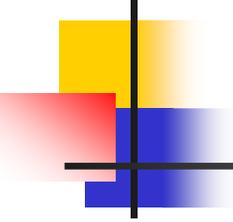
- ***chmod 744 fichier***

- L'utilisateur a tous les droits, le groupe et les autres ne peuvent que consulter le fichier



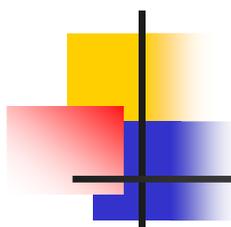
Manipulation des fichiers : copie

- ***cp*** [-i pr] *source dest*
 - *source* : fichier
 - *dest*
 - fichier : copier et renommer
 - répertoire : recopier dans *dest*
 - options
 - -i : confirmation en cas d'écrasement
 - -p : préserve les dates d'accès
 - -r : copie récursive (pour les répertoires imbriqués)



Manipulation des fichiers : création, déplacement, suppression

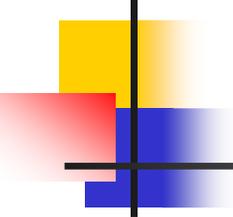
- ***mkdir*** *chemin*
 - créer un répertoire
- ***mv*** *source dest*
 - déplacer/renommer un fichier/répertoire
- ***rm*** *[-ri] nom1 nom2 ..*
 - supprimer des fichiers ou des répertoires : *nom_i*
 - options
 - -r : supprime récursivement un répertoire non vide
 - -i : supprime en demandant une confirmation avant chaque effacement



Manipulation d'archives et compressions

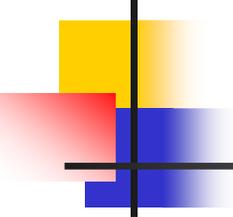
■ ***tar***

- options
 - -c : créer un archive
 - -x : extraire les fichiers
 - -v : obtenir une description du contenu archivé
 - -f : pour spécifier un nom pour l'archive (en paramètre)
 - -z: zipper/dézipper l'archive
- archivage : *tar -cvf archive.tar MonRepertoire*
- restitution : *tar -xvf archive.tar*
- compression + archivage: *tar -czvf archive.tar.gz MonRep*
- décompression + restitution: *tar -xzvf archive.tar.gz*



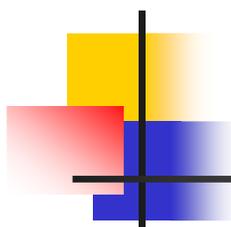
Manipulation de texte

- ***grep*** *[options] motif fichier1 fichier2 ..*
 - chercher les lignes contenant le motif dans les fichiers
 - options
 - -c : seulement le nombre de lignes
 - -n : indique les numéros des lignes trouvés
 - -i : ne distingue pas majuscule et minuscules
- ***sort*** *[-r] fichier*
 - trier les lignes du fichier ou l'entrée standard.
 - option
 - -r : renverser le tri



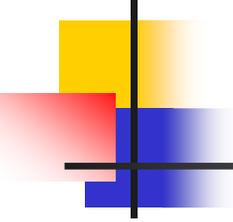
Gestion de processus

- *ps*
 - afficher les processus en cours d'exécution
- *top*
 - afficher les processus les plus actifs en temps réel
- *kill*
 - envoyer un message à un processus donné, généralement pour y mettre fin
 - signal SIGTERM (15) par défaut : arrêter le processus proprement
 - signal SIGKILL (9) : terminer brutalement un processus



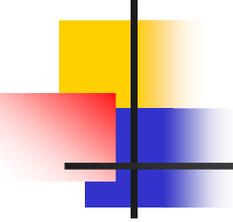
Redirection des entrées/sorties

- Entrée standard : clavier, lecture
- Sortie standard : écran, écriture
- Redirection de l'entrée
 - <
 - exemple
 - *cat* < *fichier*
 - afficher le contenu du **fichier**



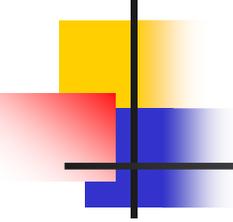
Redirection des entrées/sorties 2

- Redirection de la sortie
 - >
 - exemple
 - *ls > résultats*
 - créer un fichier nommé **résultats** contenant le contenu du répertoire courant
 - >>
 - exemple
 - *ls >> resultats*
 - ajouter à la fin du fichier **résultats** le contenu du répertoire courant



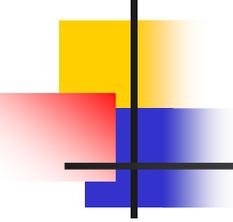
Redirection des entrées/sorties 3

- Redirection entrée sortie
 - exemple
 - *cat < fichier >> resultat*
 - *Effet?*
- Redirection vers des tubes
 - |
 - exemple
 - *ls | sort*
 - afficher le contenu du répertoire courant trié



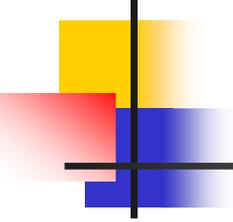
Redirection des entrées/sorties 3

- Redirection entrée sortie
 - exemple
 - *cat < fichier >> resultat*
 - ajouter au fichier **resultat** le contenu du **fichier**
- Redirection vers des tubes
 - |
 - exemple
 - *ls | sort*
 - afficher le contenu du répertoire courant trié



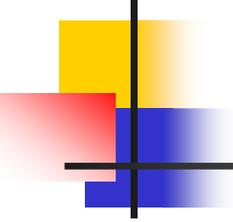
Recherche

- ***find*** [*options*]
 - chercher dans une hiérarchie de répertoires les fichiers vérifiant certaines caractéristiques.
 - exemples
 - ***find / -name charte -print***
 - chercher à partir de la racine tous les fichiers dont le nom est **charte**.
 - ***find . -name *.kwd -print***
 - chercher à partir du répertoire courant tous les fichiers dont l'extension est **.kwd**
- ***which*** *commande*
 - retourner le chemin d'accès complet d'une commande.



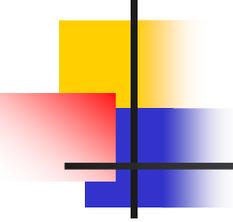
Commandes diverses

- ***who***
 - lister des utilisateurs connectés au système
- ***date***
 - afficher date et heure
- ***file fichier***
 - déterminer le type du fichier



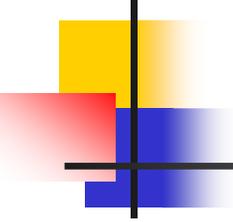
Commandes diverses 2

- ***head*** [-*n*] *fichier*
 - afficher les *n* premiers lignes du fichier
- ***tail*** [+*n*|-*n*] *fichier*
 - +*n* : afficher à partir de la ligne numéro *n*
 - -*n* : afficher le *n* dernières lignes
- ***more*** *fichier*
 - afficher le fichier page par page
- ***wc*** [-*cw*l] *fichier*
 - -*c* : nombre de caractères
 - -*w* : nombre de mots
 - -*l* : nombre de lignes



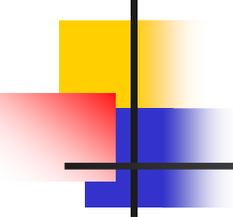
Expressions régulières

- Analyser des chaînes de caractères
- Pattern matching
- Utilisation :
 - Commandes : grep, sed, ...
 - Éditeurs de textes : vi, nedit, ...
 - Langages de programmation : php, perl, ...



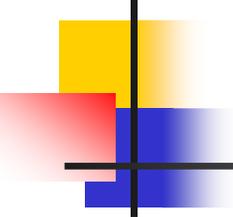
Les symboles ^ et \$

- ^ : début d'un pattern
- \$: fin d'un pattern
- Exemples :
 - ^at
 - cot\$
 - ^mot\$
 - toto



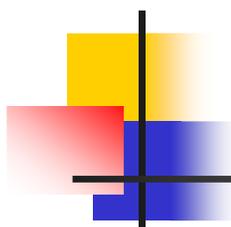
Les symboles $*$, $+$ et $?$

- Nombre de fois qu'un caractère (suite de caractères) puisse apparaître
 - $*$: aucune fois ou plusieurs fois
 - $+$: une fois ou plusieurs fois
 - $?$: aucune fois ou une et une seule fois
- Exemples :
 - ab^*
 - ab^+
 - $ab?$
 - $a?b^+\$$



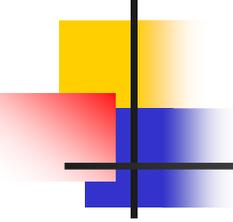
Les symboles *, + et ?

- Nombre de fois qu'un caractère (suite de caractères) puisse apparaître
 - * : aucune fois ou plusieurs fois
 - + : une fois ou plusieurs fois
 - ? : aucune fois ou une et une seule fois
- Exemples :
 - ab^* : chaîne de caractère contenant un a suivi d'un ou d'aucun b ("a", "ab", "abb", ...)
 - ab^+ : chaîne de caractère contenant un a suivi d'au moins un b ("ab", "abb", "abbb", ...)
 - $ab?$: chaîne de caractère contenant un a suivi d'un ou d'aucun b ("a", "ab", mais pas "ab ")
 - $a?b+\$$: chaîne de caractères composée d'aucun ou d'un seul a, suivi d'un ou de plusieurs b, le tout étant situé à la fin de la chaîne



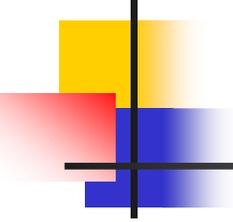
Les accolades $\{ \}$

- Nombre d'occurrences de la chaîne
- Exemples :
 - $ab\{2\}$
 - $ab\{2,\}$
 - $ab\{3,5\}$



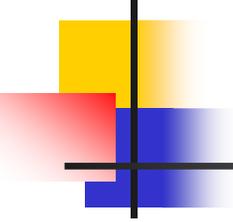
Les accolades $\{ \}$

- Nombre d'occurrences de la chaîne
- Exemples :
 - $ab\{2\}$: chaîne de caractère composée d'un a suivi d'exactly deux b ("abb")
 - $ab\{2,\}$: chaîne de caractère composée d'un a suivi d'au moins deux b ("abb", "abbb",...)
 - $ab\{3,5\}$: chaîne de caractère composée d'un a suivi de trois à cinq b ("abbb", "abbbb" et "abbbbb")



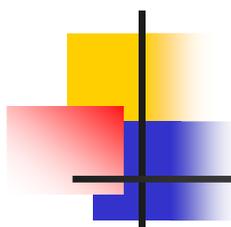
Les parenthèses

- Quantifier une chaîne de caractères
- Exemples :
 - $a(bc)^*$
 - $a(bc)\{1,5\}^*$



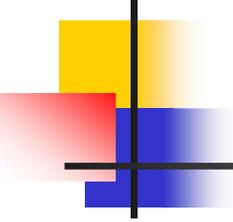
Les parenthèses

- Quantifier une chaîne de caractères
- Exemples :
 - $a(bc)^*$: chaîne de caractères commençant par un a suivi d'aucune ou de plusieurs séquence de caractères "bc"
 - $a(bc)\{1,5\}^*$: chaîne de caractères commençant par un a suivi d'une à cinq fois la séquence de caractères "bc"



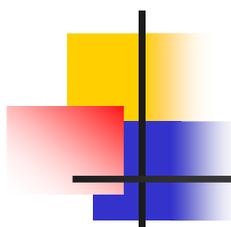
Le symbole |

- Comme opérateur booléen OU
- Exemples :
 - toto|titi
 - (b|cd)ef
 - (a|b)*c



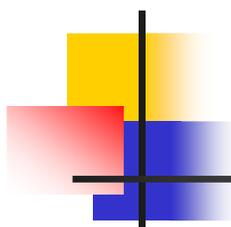
Le symbole |

- Comme opérateur booléen OU
- Exemples :
 - `toto|titi` : chaîne de caractères contenant le mot "toto" ou le mot "titi"
 - `(b|cd)ef` : chaîne de caractères qui contient la séquence de caractères "bef" ou bien la séquence de caractères "cdef"
 - `(a|b)*c` : chaîne de caractères qui contient une alternance de a et de b, se terminant par un c ("bababbbaac", "c", "bc")



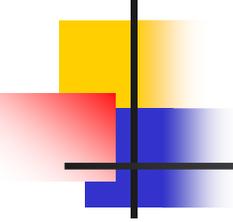
Le symbole .

- N'importe quel caractère unique.
- Exemple :
 - $\wedge.\{3,\}\$$



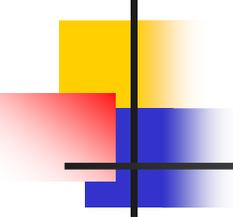
Le symbole .

- N'importe quel caractère unique.
- Exemple :
 - $^{\{3,\}}\$$: chaîne de caractères comportant exactement trois caractères



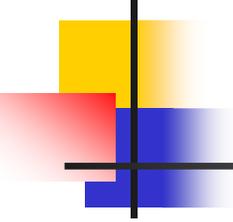
Les crochets []

- Les caractères permis à un endroit précis d'un modèle
- Exemples :
 - [ab]
 - [a-d]
 - $^{\wedge}$ [a-zA-Z]
 - [0-9]%
 - $^{\wedge}$ [a-zA-Z0-9]\$



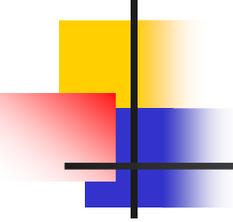
Les crochets []

- Les caractères permis à un endroit précis d'un modèle
- Exemples :
 - [ab] : chaîne de caractères contenant un "a" ou un "b"
 - [a-d] : chaîne de caractères qui contient les lettres minuscules comprises entre le "a" et le "d"
 - ^[a-zA-Z] : chaîne de caractères qui commence par une lettre minuscules ou bien par une lettre majuscule
 - [0-9]% : chaîne de caractères qui contient un pourcentage à un seul chiffre
 - ,[a-zA-Z0-9]\$: chaîne de caractères qui finit par une virgule suivi d'un caractère alphanumérique



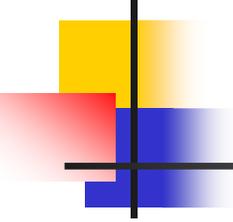
Le ^ dans les crochets

- ^ comme premier symbole dans les crochets: « tout sauf »
 - [^a]
 - %[^a-zA-Z]%



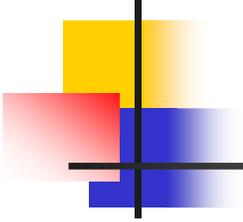
Le ^ dans les crochets

- ^ comme premier symbole dans les crochets: « tout sauf »
 - [^a]: chaîne de caractères qui ne commence pas par « a »
 - %[^a-zA-Z]% : chaîne de caractères avec un caractère qui n'est pas une lettre entre deux signes pourcentage



Notez

- Si vous mettez les expressions régulières entre guillemets, pensez à écrire des *backslash* avant (,), { et }
- Exemple: `grep "^a\(b|c\) \{1,3\}" fichier`



FIN