DOSSIER PROGRAMMATION

Table des matières

[I. Langage de Programmation 4](#_Toc32919608)

[A. Language C 4](#_Toc32919609)

[1. Introduction 4](#_Toc32919610)

[2. Les variables 4](#_Toc32919611)

[3. Condition IF et case 4](#_Toc32919612)

[4. Boucle While, until 4](#_Toc32919613)

[5. Boucle for 4](#_Toc32919614)

[6. Les fonctions 4](#_Toc32919615)

[B. Language Java 4](#_Toc32919616)

[1. Introduction 5](#_Toc32919617)

[2. Les variables 5](#_Toc32919618)

[3. Condition IF et case 5](#_Toc32919619)

[4. Boucle While, until 5](#_Toc32919620)

[5. Boucle for 5](#_Toc32919621)

[6. Les fonctions 5](#_Toc32919622)

[C. Language Python 5](#_Toc32919623)

[1. Introduction 5](#_Toc32919624)

[2. Les variables 5](#_Toc32919625)

[3. Condition IF et case 5](#_Toc32919626)

[4. Boucle While, until 5](#_Toc32919627)

[5. Boucle for 5](#_Toc32919628)

[6. Les fonctions 5](#_Toc32919629)

[D. Language C # 5](#_Toc32919630)

[1. Introduction 6](#_Toc32919631)

[2. Les variables 6](#_Toc32919632)

[3. Condition IF et case 6](#_Toc32919633)

[4. Boucle While, until 6](#_Toc32919634)

[5. Boucle for 6](#_Toc32919635)

[6. Les fonctions 6](#_Toc32919636)

[II. Système d’exploitation 6](#_Toc32919637)

[A. Linux 6](#_Toc32919638)

[1. Architecture 6](#_Toc32919639)

[2. La console 7](#_Toc32919640)

[3. Connexion via SSH 23](#_Toc32919641)

[4. compiler un programme depuis les sources 36](#_Toc32919642)

[5. Les scripts bach 37](#_Toc32919643)

[III. HTML & CSS 51](#_Toc32919644)

[A. Description 51](#_Toc32919645)

[B. Les balises et attribut 52](#_Toc32919646)

[1. Les balises blocs 52](#_Toc32919647)

[2. Les balises lignes 53](#_Toc32919648)

[3. Les balises orphelines 53](#_Toc32919649)

[4. Les balises tableaux 53](#_Toc32919650)

[5. Les balises formulaires 54](#_Toc32919651)

[6. Les attributs 56](#_Toc32919652)

[7. Le CSS 59](#_Toc32919653)

[C. Les média Query 62](#_Toc32919654)

[1. Explication 62](#_Toc32919655)

[2. Les règles 62](#_Toc32919656)

[D. Mémento 64](#_Toc32919657)

[1. Les balises. 64](#_Toc32919658)

[*2.* *Les propriétés CSS* 67](#_Toc32919659)

# Langage de Programmation

## Language C

### Introduction

### Les variables

### Condition IF et case

### Boucle While, until

### Boucle for

### Les fonctions

## Language Java

### Introduction

### Les variables

### Condition IF et case

### Boucle While, until

### Boucle for

### Les fonctions

## Language Python

### Introduction

### Les variables

### Condition IF et case

### Boucle While, until

### Boucle for

### Les fonctions

## Language C #

### Introduction

### Les variables

### Condition IF et case

### Boucle While, until

### Boucle for

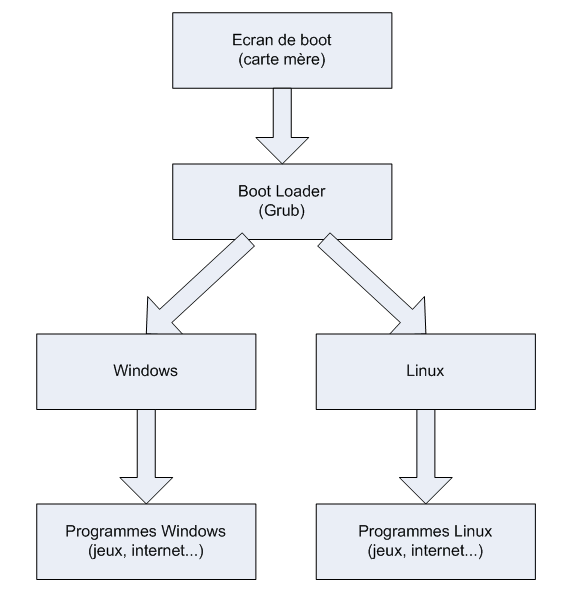
### Les fonctions

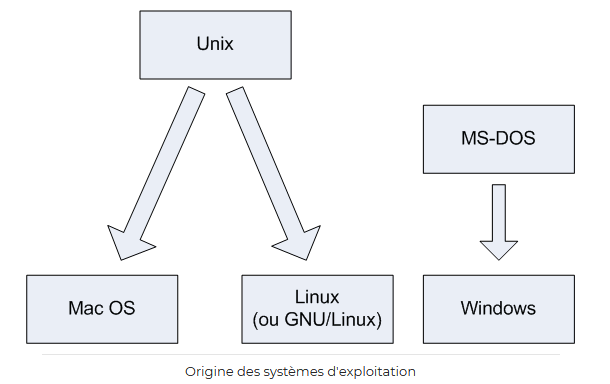
# Système d’exploitation

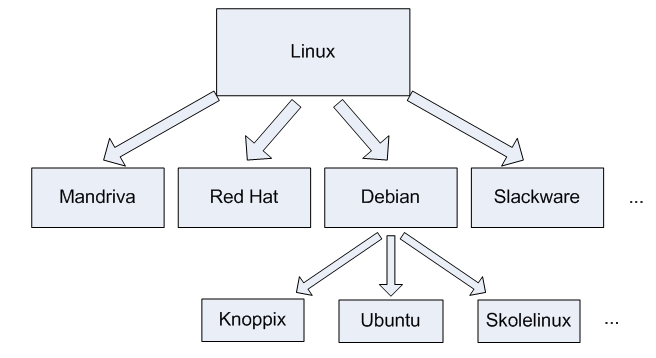
## Linux

### Architecture

Richard Stallman créa le projet GNU







### La console

Pour ouvrir la console en mode graphique CTRL + T, ou ATRL +ALT +F1 a F7. Toute commande est décris dans le manuel en tapant dans la console : man + commande

***$Date***

**Description :** Donne la date

**Paramètre :**

***$ls***

**Description :** liste les fichiers et dossier

**Paramètre :**

1. **–a (--all) :** affiche même les fichiers cachés.
2. **–F :** indique le type d’élément
3. **–l :** liste détaillé.
4. **–h :** Affiche la taille en Ko, Mo, Go.
5. **–t :** trie par date de dernière modification.

***$pwd***

**Description :** Affiche le dossier actuel

**$which**

**Description :** Affiche l’emplacement d’une commande

**$cd**

**Description :** change de dossier.

**$du**

**Description :** taille occupée par les dossiers

**Paramètre :**

1. **–h :** Affiche la taille en Ko, Mo, Go.
2. **–a :** Affiche la taille des dossiers et des fichiers.
3. **–s :** Affiche juste la taille total.

**$cat**

**Description :** affiche tout le fichier

**$less**

**Description :** affiche le fichier page par page

**$head**

**Description :** affiche le début d’un fichier

**Paramètre :**

1. **–n + nombre:** affiche les N premières lignes

**$tail**

**Description :** affiche le début d’un fichier

**Paramètre :**

1. **–n + nombre:** affiche les N dernières lignes
2. **–f :** Permet de suivre un fichier en cours d’écriture.

**$touch**

**Description :** crée un fichier.

**$mkdir**

**Description :** crée un dossier

**Paramètre :**

1. **–p :** Crée tous les dossiers intermédiaires ex : mkdir -p animaux/vertebres/chat

**$cp**

**Description :** copie un fichier le 1er est le fichier copié le deuxième le fichier crée

**Paramètre :**

1. **–R :** Copie les dossiers ainsi que les sous-dossiers.

**$mv**

**Description :** déplace un fichier le 1er est le fichier à déplacer le deuxième le chemin

**$rm**

**Description :** supprime des fichiers et dossiers

**Paramètre :**

1. **–i :** Demande confirmation
2. **–f :** Force la suppression quoi qu’il arrive
3. **–v :** décris ce qu’il se fait.
4. **–r :** Supprime un dossier et tout son contenu

**$ln**

**Description :** crée un lien entre deux fichiers (par défaut ; physiques)

**Paramètre :**

1. **–s :** Crée un lien symbolique

**$sudo**

**Description :** Exécute une commande en root.

**Paramètre :**

1. **su:** Devenir root

**$adduser**

**Description :** ajouter un utilisateur

**$passwd**

**Description :** changer le mot de passe

**$deluser**

**Description :** supprimer un compte

**$addgroup**

**Description :** Ajoute un groupe.

**$** **usermod**

**Description :** modifie un utilisateur.

**Paramètre :**

1. **–l :** renomme l’utilisateur
2. **–g :** change de groupe

**$delgroup**

**Description :** supprime un groupe

**$chown**

**Description :** Gère les propriétaires d’un fichier

**Paramètre :**

1. **propriétaire : groupe :** Change le propriétaire d’un fichier ainsi que son groupe propriétaire.
2. **- R :** Affecte les sous dossiers

**$chgrp**

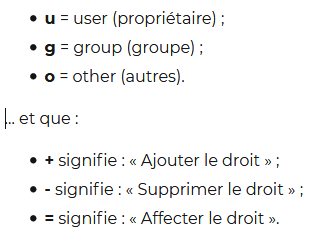
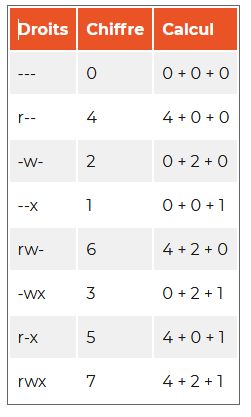
**Description :** Change le groupe propriétaire d’un fichier

**$chmod**

**Description :** modifie les droits d’accès.

**Paramètre :**

1. **- R :** Affecte les sous dossiers



**$apt-get + update**

**Description :** mettre à jour le cache des paquets

**$apt-get + search**

**Description :** rechercher un paquet

**$apt-get + install**

**Description :** télécharge et installe un paquet

**$apt-get + upgrade**

**Description :** Va mettre à jour l’intégralité des paquets en allant chercher sur internet , utiliser **$apt-get + update** avant afin de mettre à jour le cache.

**$man + commande**

**Description :** utilise le manuel afin d’avoir une description détaillé de la commande.

Résumé de la syntaxe du SYNOPSIS

Voici un petit résumé de la syntaxe du SYNOPSIS pour vous souvenir de la façon dont chaque élément doit être interprété :

* gras : tapez le mot exactement comme indiqué ;
* souligne : remplacez le mot souligné par la valeur qui convient dans votre cas ;
* [-hvc] : toutes les options -h, -v et -c sont facultatives ;
* a|b : vous pouvez écrire l'option « a » OU « b », mais pas les deux à la fois ;
* option… : les points de suspension indiquent que l'option peut être répétée autant de fois que vous voulez.

**$apropos + mots clé**

**Description :** cherche une commande en fonction d’un mot clés.

**$commande -h**

**Description :** utilise l’aide décrivant la commande.

**$whatis + commande**

**Description :** une petite description de la commande

**$locate + fichier**

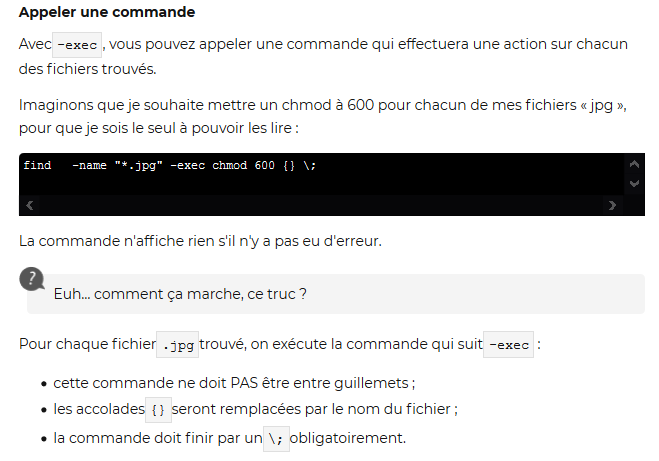
**Description :** va rechercher un fichier dans une base de donnée. Son défaut est si la base de donnée n’est pas à jour elle ne pourra pas trouver le fichier, cas lorsque vous venez de créer un fichier. ( utiliser la commande **$sudo updatedb** )

**$find**

**Description :** cherche un fichier dans le dossier et sous dossier par défaut, un répertoire peut être décris et exécute ce qu’on lui demande.

**Paramètre :**

1. **-name :** cherche un fichier via son nom « nom ».
2. **-size (-)(+) :** cherche un fichier via sa taille k=kilos, M=Méga et G = Giga
3. **-atime :** cherche un fichier depuis la date de dernière modification commence par 0 pour un jour.
4. **-type (d)(f) :** cherche un fichier via son type d = directories, et f= files
5. **-printf  :** voir manuel, peut afficher les résultats en fonction de ce que l’on veut.
6. **-delete :** supprime les fichiers trouvés
7. **-exec :** voir ci-dessous.

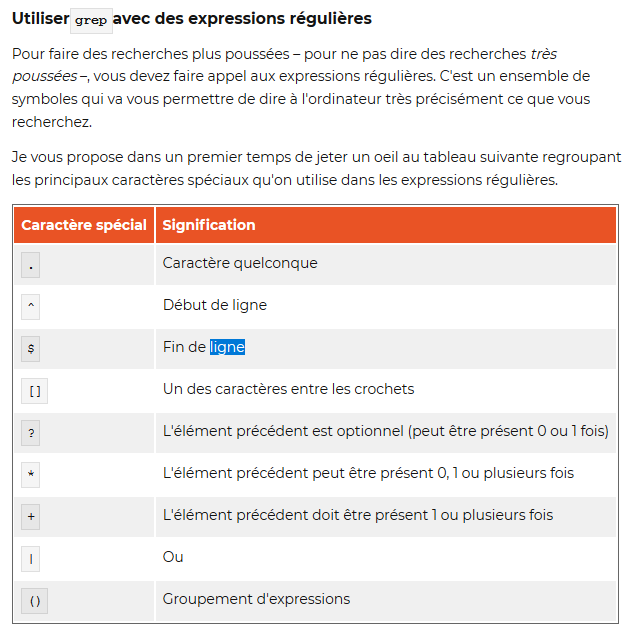


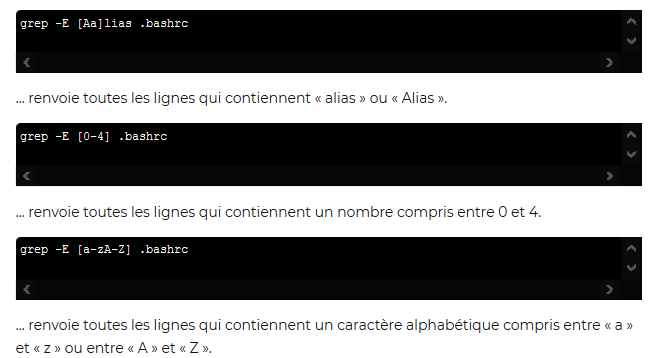
**$grep+filtre+fichier**

**Description :** filtre les données d’un fichier utilisé des guillemets lorsque le filtre est une suite de plusieurs mots.

**Paramètre :**

1. **-i :** ne tiens pas compte de la CASSE
2. **-n :** connaître le numéro des lignes.
3. **-v :** inverse la recherche ignore un mot.
4. **-r :** recherche dans tout les fichiers et sous dossiers, **$grep+filtre+dossier/**
5. **-E :** Utilise des expressions régulière voir ci-dessous.





**$** **sort + fichier**

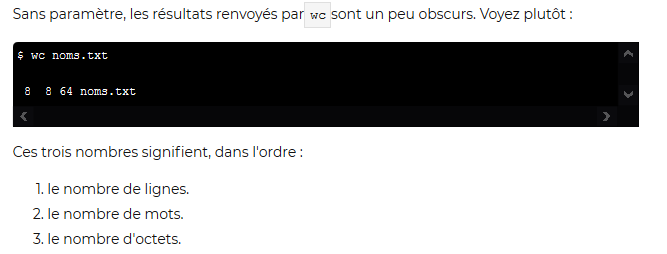
**Description :** trie des lignes

**Paramètre :**

1. **–o :** écrit le résultat dans un fichier **$sort + fichiertrié +fichierenregistré**
2. **–r :** trie en ordre inverse
3. **–R :** trie de manière aléatoire
4. **–n :** trie des nombres

**$** **wc + fichier**

**Description :** compte le nombre de ligne.



**Paramètre :**

1. **–l :** compte le nombre de ligne
2. **–w :** compte le nombre de mots
3. **–c :** compte le nombre d’octets
4. **–m :** compte le nombre de caractère

**$** **uniq + fichier**

**Description :** supprime les doublons ( peut être enregistré dans un fichier **$uniq fichier1 fichier2** )

**Paramètre :**

1. **–c :** compte le nombre d’occurrences
2. **–d :** affiche uniquement les lignes présentes en double

**$** **cut + fichier**

**Description :** coupe une partie du fichier



**Paramètre :**

1. **–d :** indique le délimiteur
2. **–f :** indique le numéro du ou des champs à couper



**$** (chevron) Commande + **>** fichier

**Description :** redirige le résultat dans un fichier, sera créer si il n’existe pas, ou supprime les données si il y en avait déjà.



**$** (double chevron) Commande + >**>** fichier

**Description :** redirige le résultat à la fin d’un fichier, sera créer si il n’existe pas.



**$** (2 chevron) Commande + >>fichier **2>** fichier

**Description :** redirige les erreurs dans un fichier. (2>> peut être utiliser pour ajouter à la fin d’un fichier )



**$**Commande + >>fichier **2>&1**

**Description :** redirige les erreurs le même fichier standard.



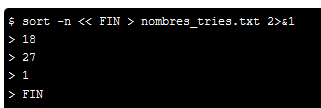
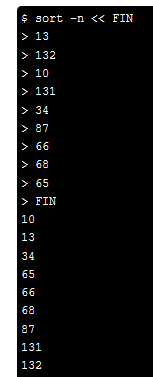
**$**commande **<** fichier

**Description :** lit depuis un fichier



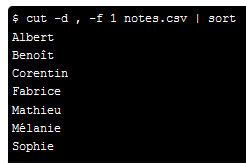
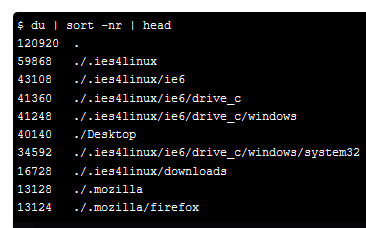
**$**commande << Mots

**Description :** lit depuis le clavier jusqu’à ce que le mots soit écrit.



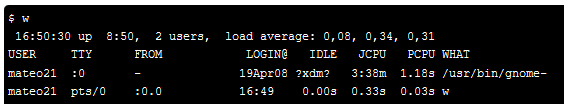
**$**commande **|** commande **|**commande ect..

**Description :** enchaine une suite de commande en renvoyant le résultat à la prochaine commande.



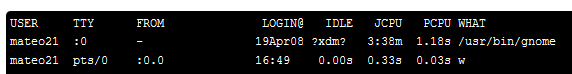
**$**w

**Description :** décris ce que font les utilisateurs connectés.



**$**who

**Description :** décris qui est connecté



**$**ps

**Description :** liste les processus de l’ordinateur en statique.

**Paramètre :**

1. **–ef :** liste tout les processus
2. **–ejH :** Affiche les processus en arbre.
3. **–u :** liste les processus lancé par l’utilisateur.

**$**ps

**Description :** liste les processus de l’ordinateur en dynamique.

**$**kill + PID

**Description :** tue un processus via son PID.

**Paramètre :**

1. **–9 :** force l’arrêt.



**$**killall + nom

**Description :** tue un processus via son nom

**$**&

**Description :** lance un processus en arrière-plan ( petit inconvénient si vous quitter le terminal le processus serra arrêté)



**$**nohup

**Description :** lance un processus en arrière-plan et détaché de la console



**$**CTRL + Z

**Description :** met en pause l’exécution d’un programme

**$**bg

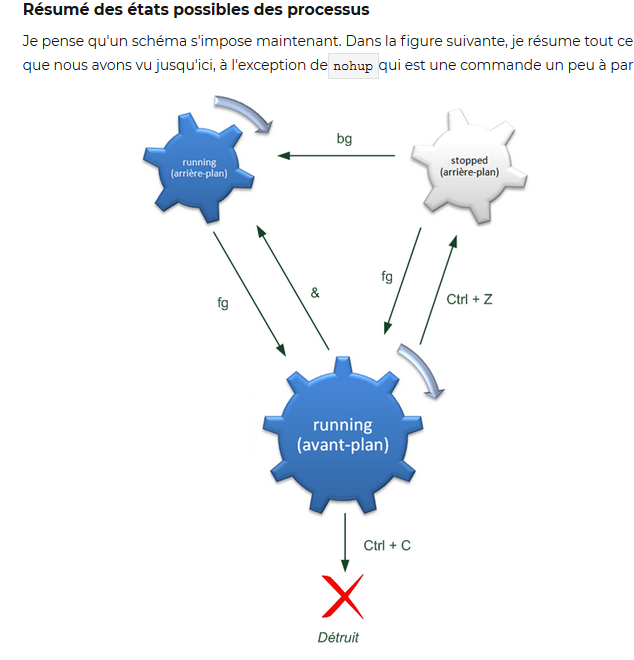
**Description :** passe le processus en arrière plan.

**$**jobs

**Description :** permet de connaître les processus qui tourne en arrière plan.

**$**fg

**Description :** reprend un processus au premier plan.

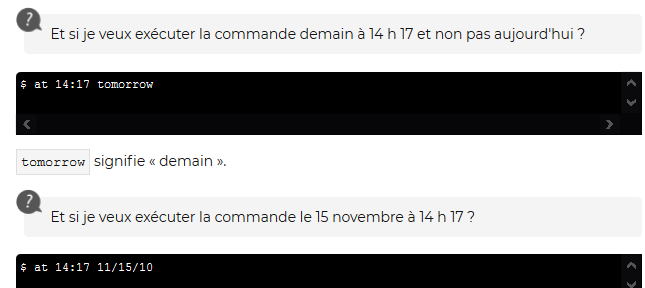


**$**screen

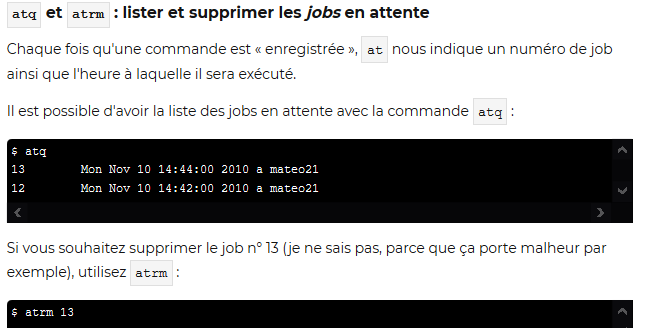
**Description :** permet d’avoir plusieurs console dans une seule console.

**$**at

**Description :** execute une commande plus tard mais une seul fois.







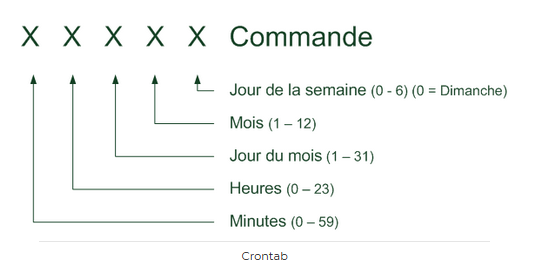


**$**crontab

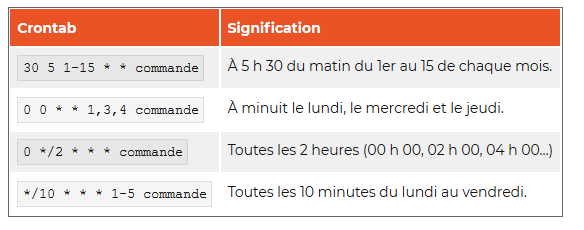
**Description :** execute des commande de manière répétitive.

**Paramètre :**

1. **–l :** Affiche la crontab actuel
2. **–e :** modifie la crontab
3. **–r :** suprimme la crontab







**$**tar

**Description :** assemble des fichier dans une archive.

**Paramètre :**

1. **–c :** crée une affiche tar



1. **–r :** ajoute un fichier dans l’archive
2. **-x :** extrait les fichiers de l’archive
3. **–v:** affiche les détails des opérations.
4. **–f:** assemble l’archive dans un fichier
5. **–tf:** affiche le contenu sans extraire
6. **–zcvf:** archive et compresse en gzip



1. **–jcvf:** archive et compresse en bzip2
2. **–zxvf:** archive et décompresse en gzip
3. **–jxvf:** archive et décompresse en bzip2



**$**gzip + fichier.tar

**Description :** compresse un fichier



**$**gunzip + fichier.tar.gz

**Description :** décompresse un fichier



**$**bzip2 + fichier.tar

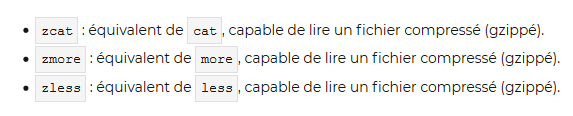
**Description :** compresse un fichier



**$**bunzip2 + fichier.tar.gz

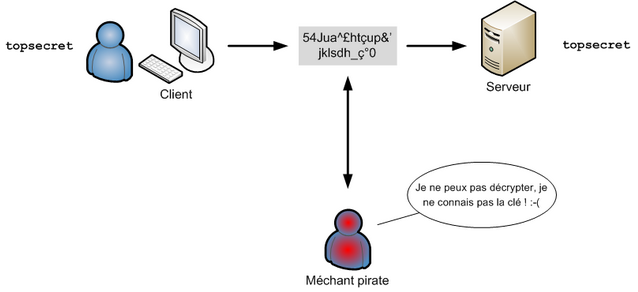
**Description :** décompresse un fichier

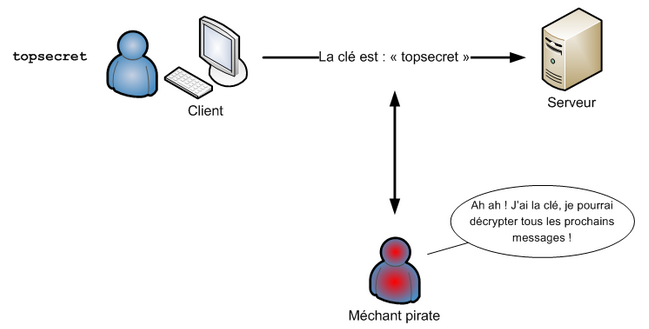




### Connexion via SSH

Le chiffrement symétrique





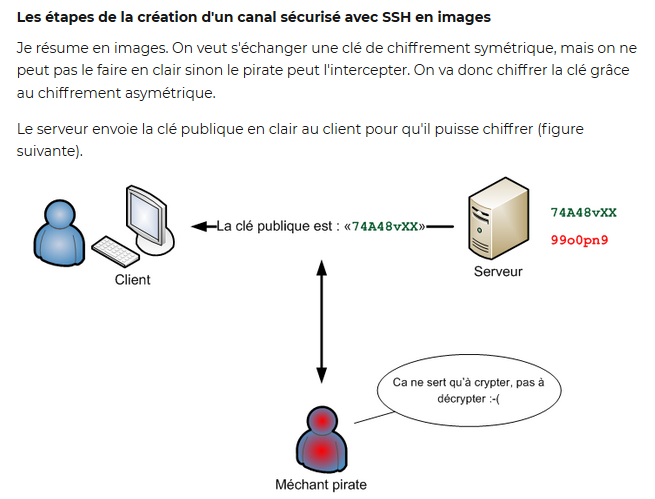
Le chiffrement symétrique est donc puissant, mais il a un gros défaut : il faut communiquer « discrètement » la clé de chiffrement… mais c'est impossible : il faut bien envoyer la clé en clair au début !

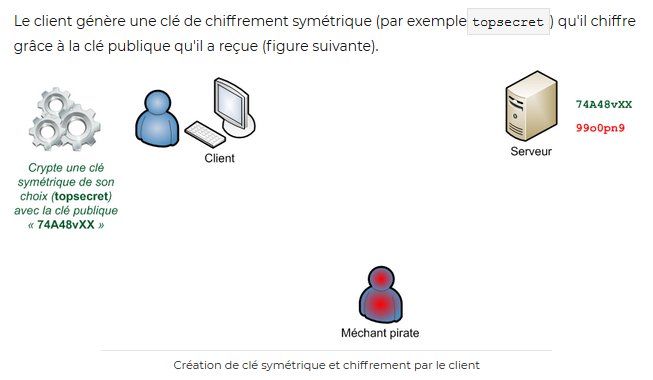
Le chiffrement asymétrique

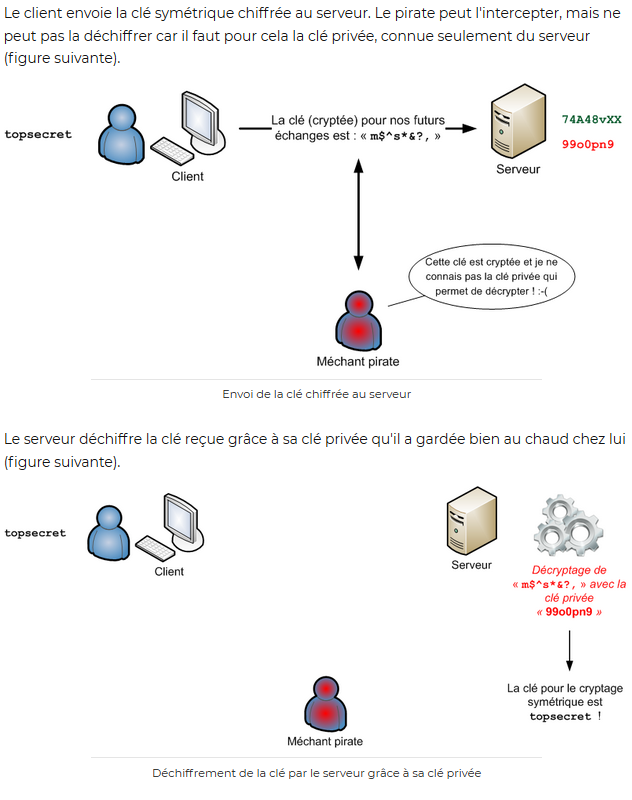
Le chiffrement symétrique utilise une seule clé pour chiffrer et déchiffrer.  
Le chiffrement asymétrique, lui, utilise une clé pour chiffrer, et une autre pour déchiffrer.

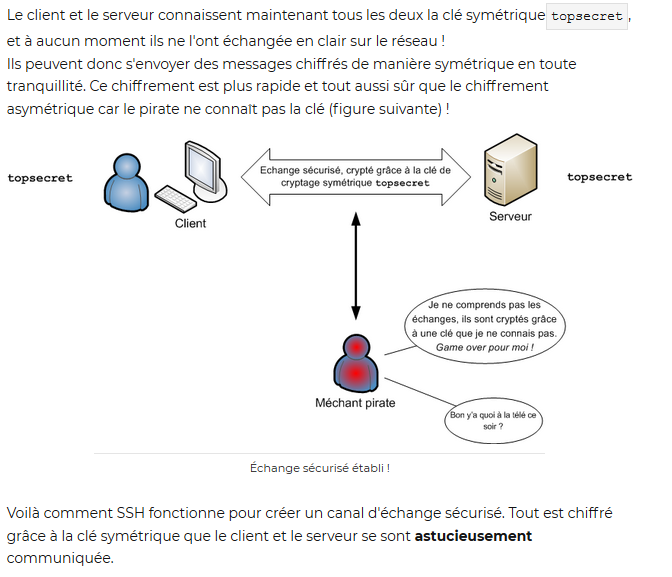
Il y a donc deux clés :

* une clé dite « **publique** » qui sert à **chiffrer**
* une clé dite « **privée** » qui sert à **déchiffrer**.

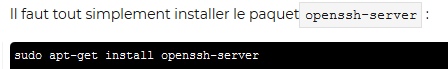


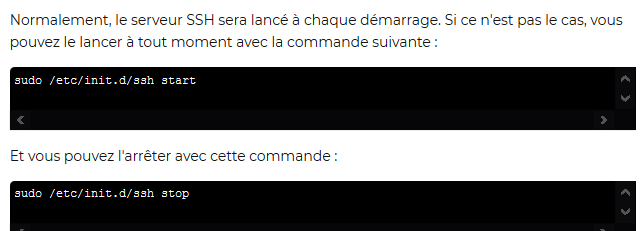






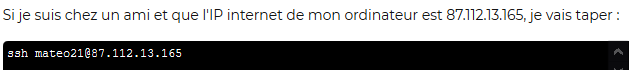
Transformer sa machine en serveur





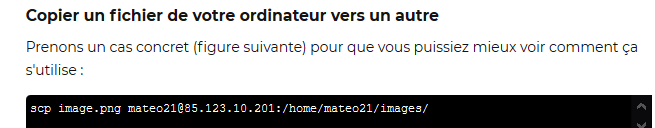
Se connecter via une machine linux

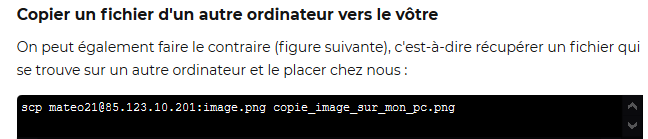


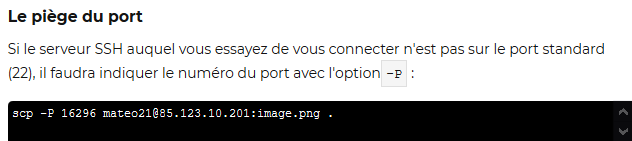


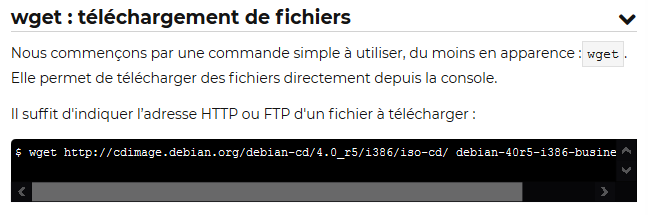


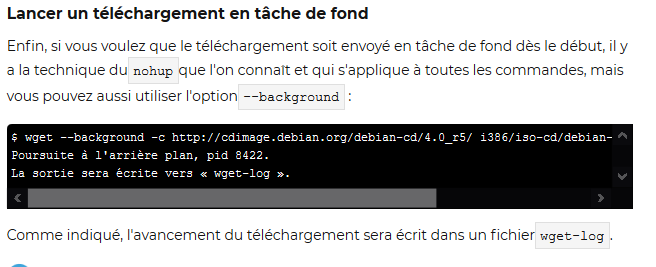


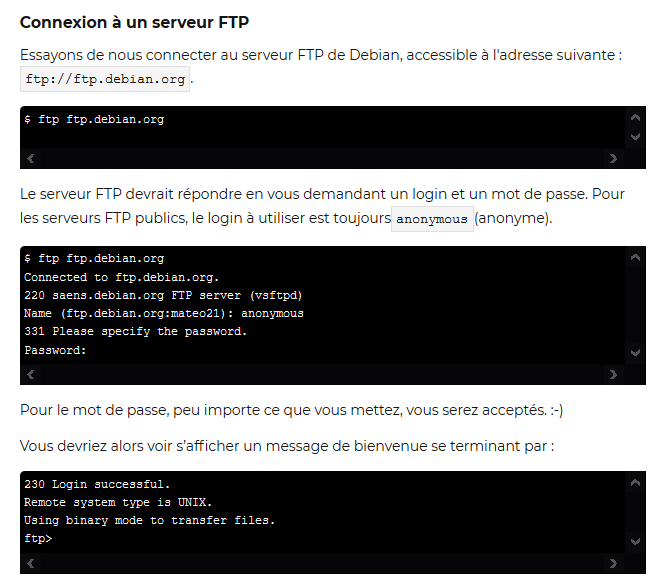


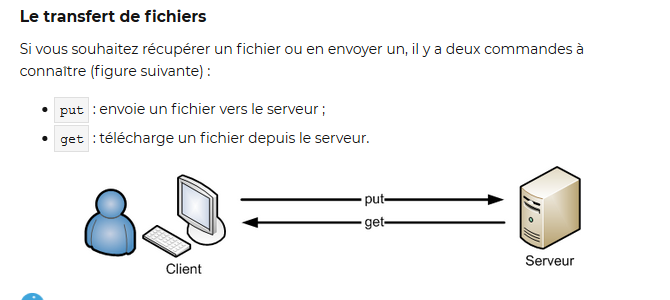


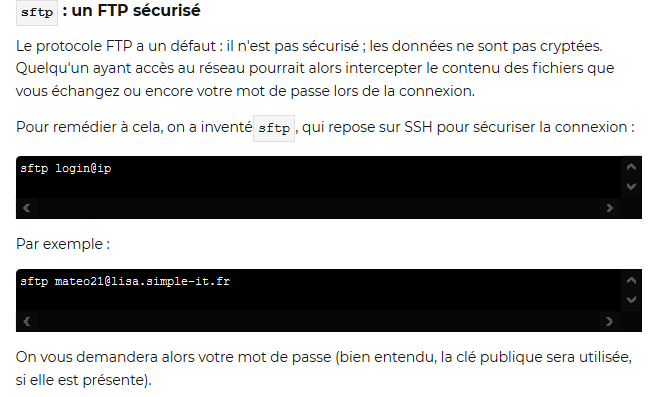








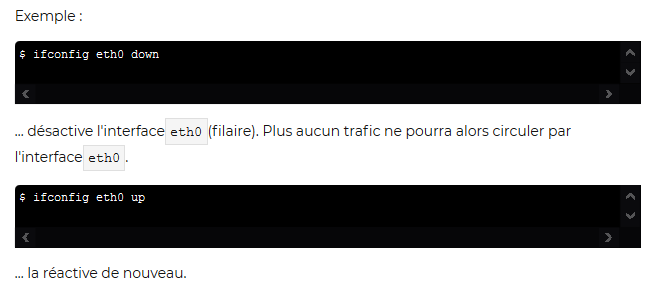




**$ rsync -arv --delete --backup –backup dir=/home/mateo21/fichiers\_supprimes Images/ mateo21@IP\_du\_serveur:mes\_backups/**

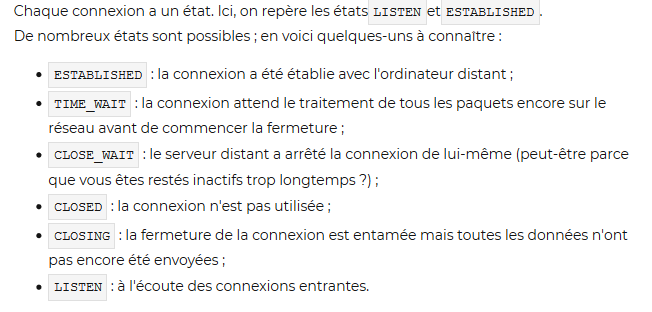






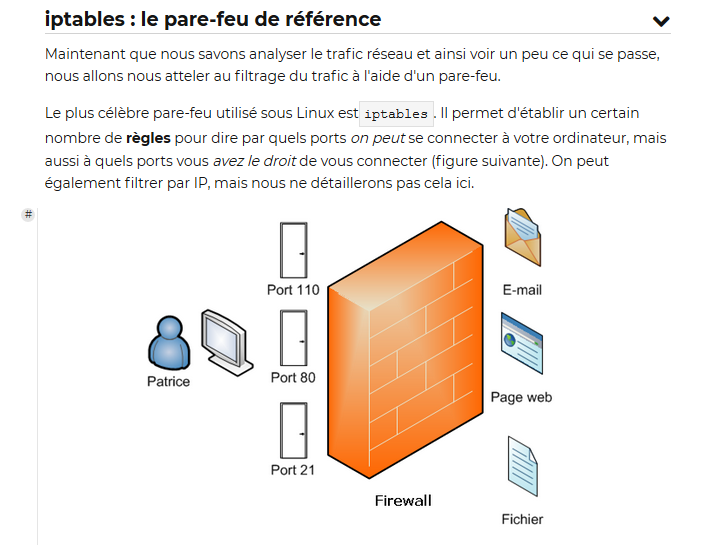




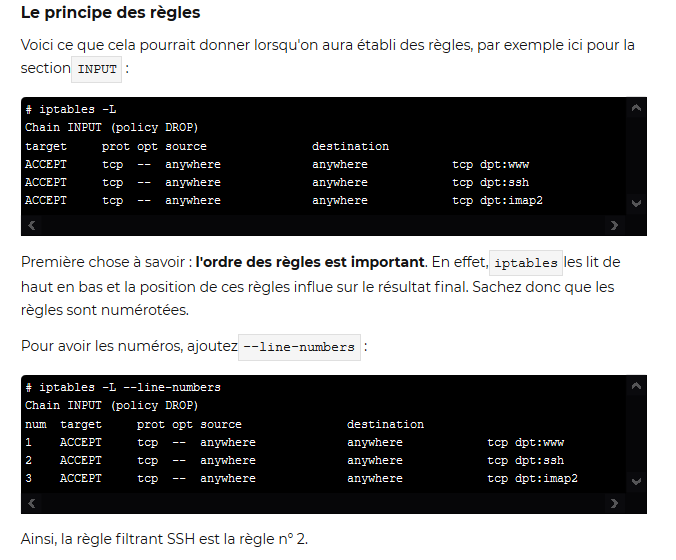


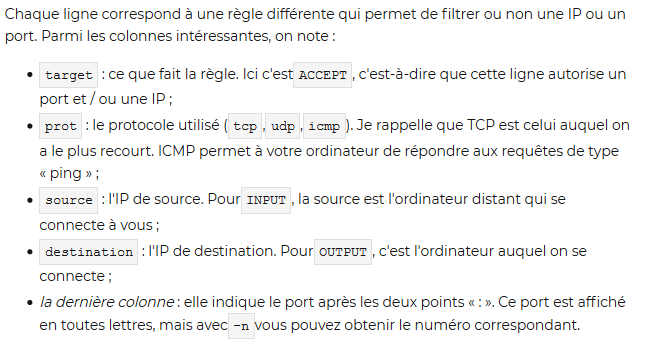


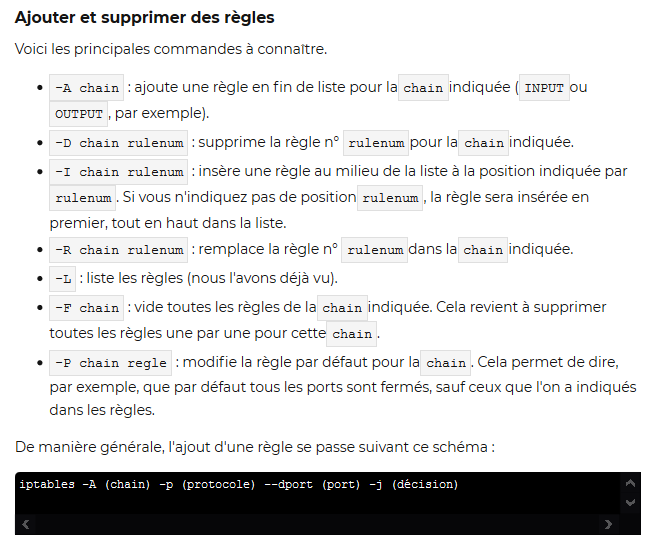






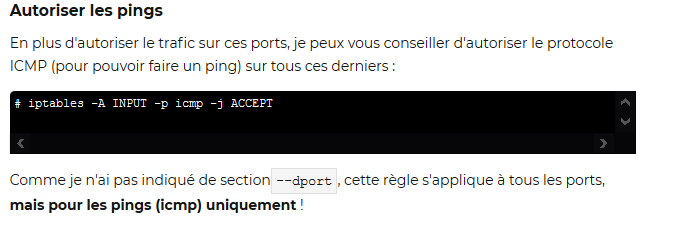
\*

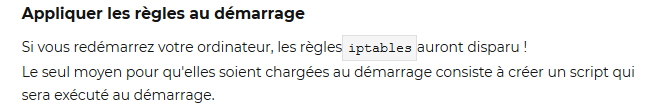


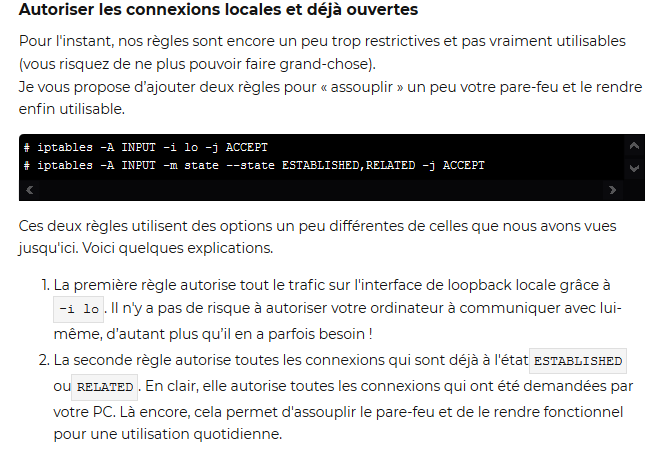


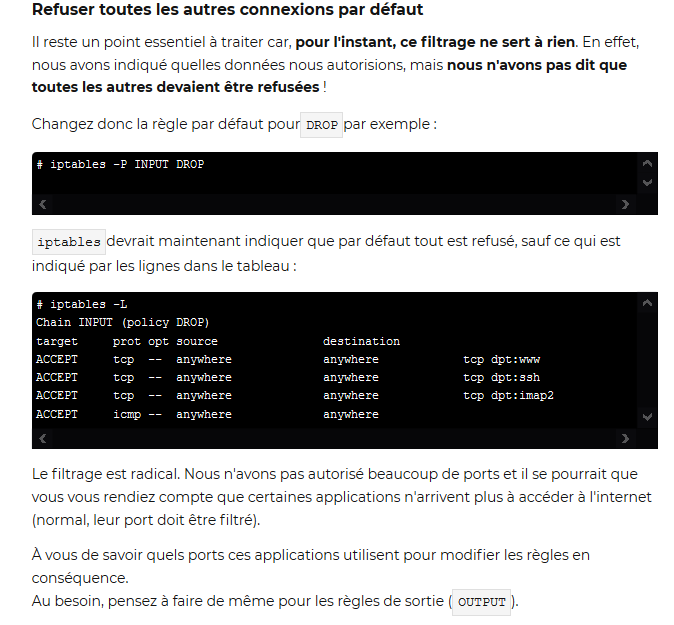










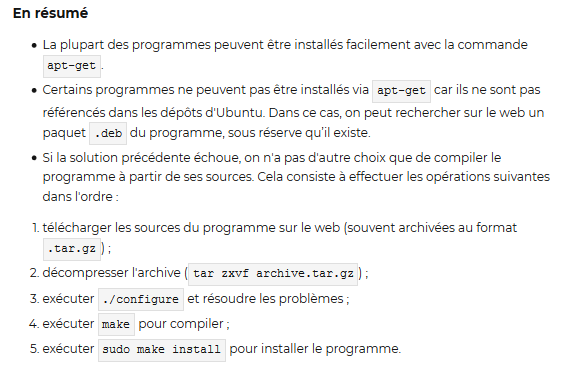


### compiler un programme depuis les sources



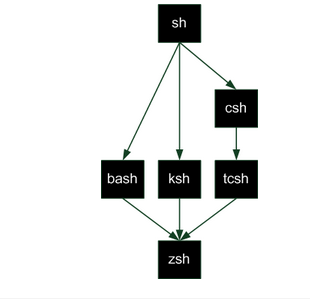


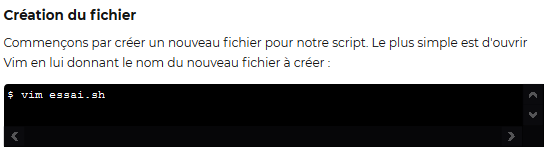




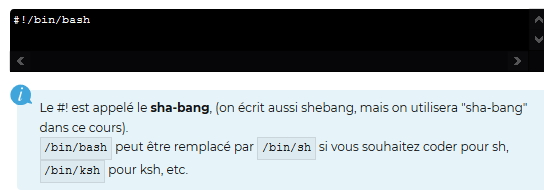
### Les scripts bach

#### Introduction

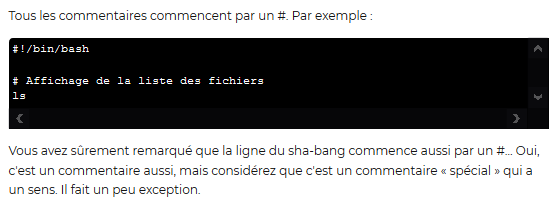






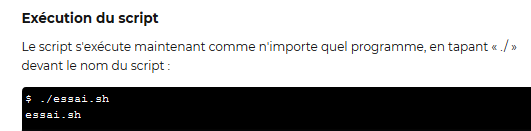


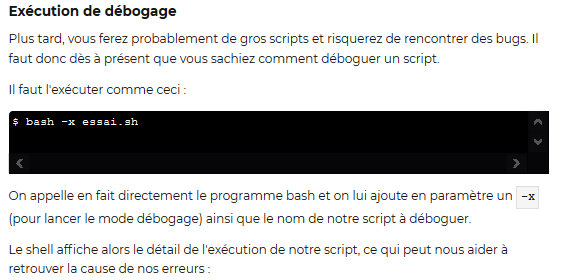


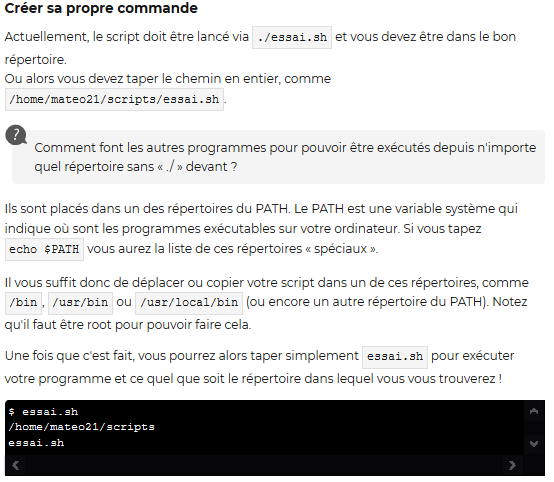






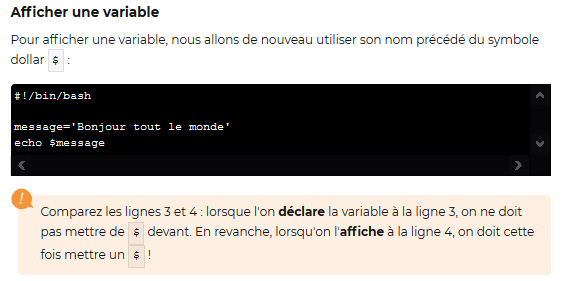


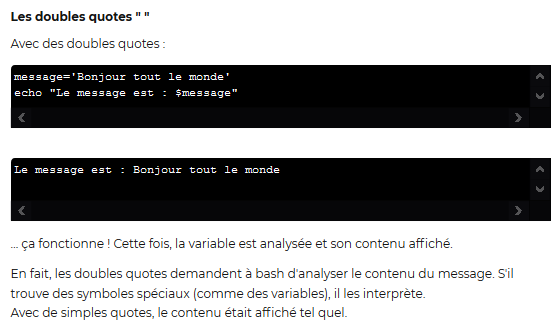


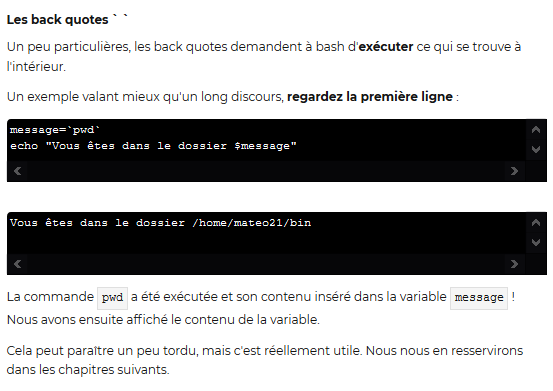


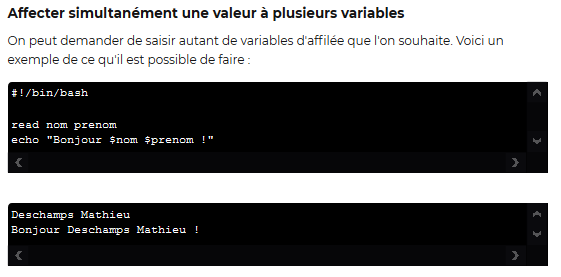
#### Les variables

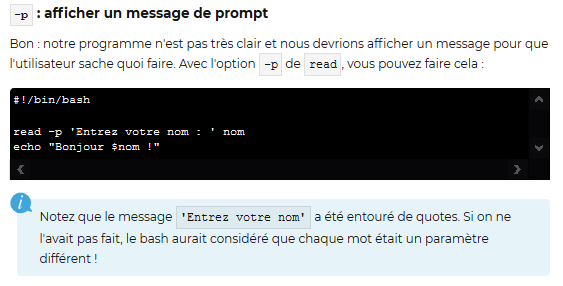


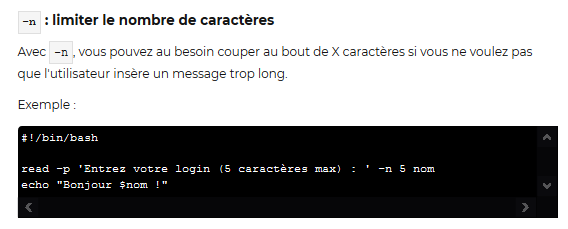






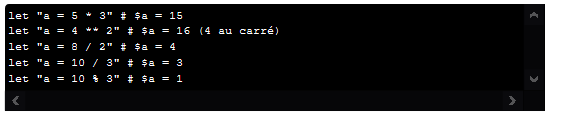




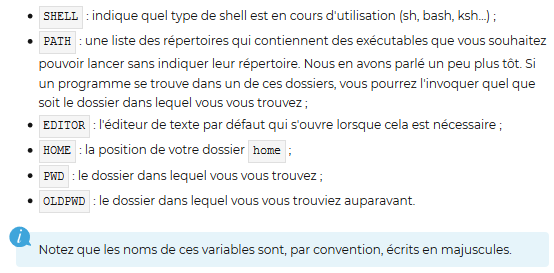


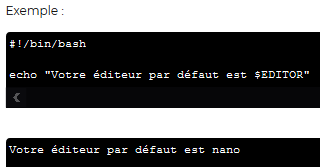


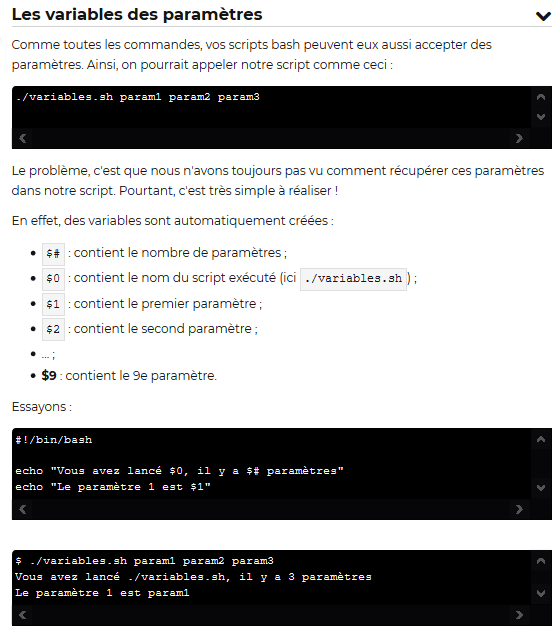


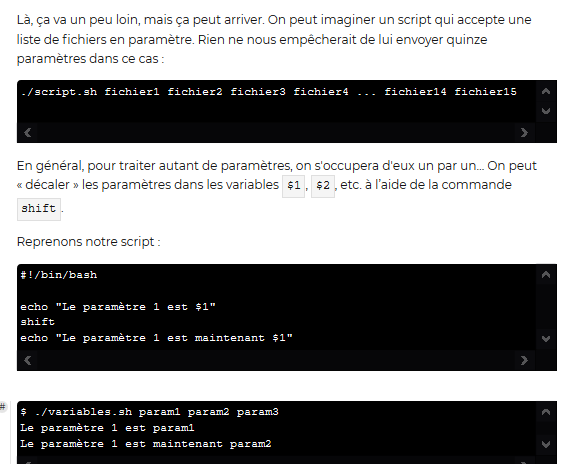


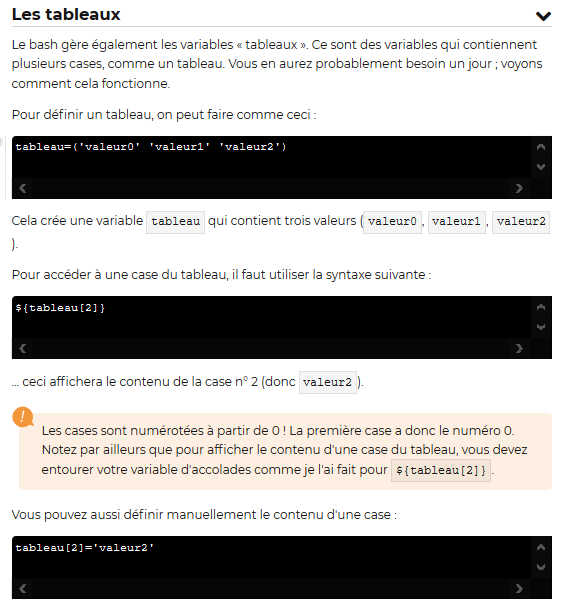


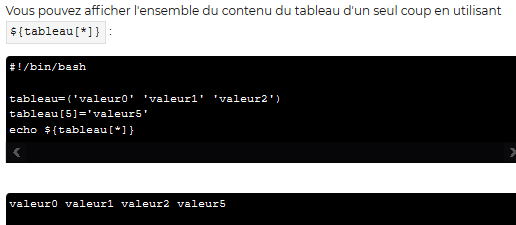




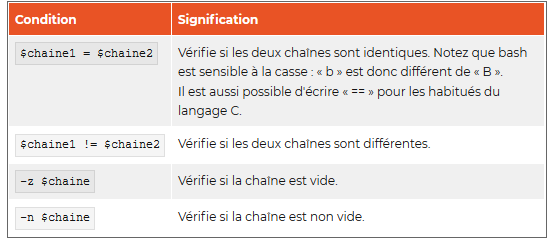


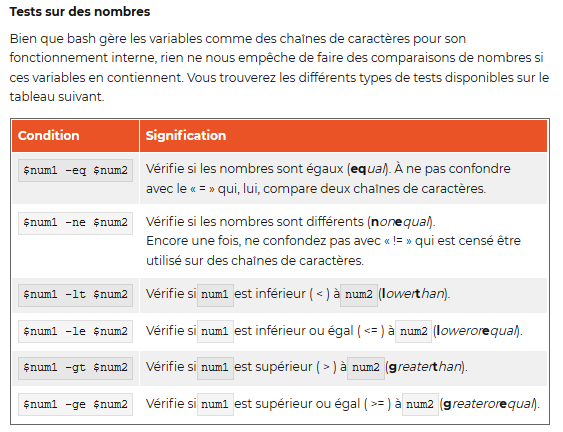




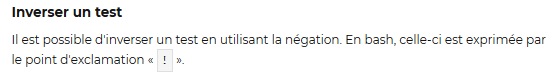
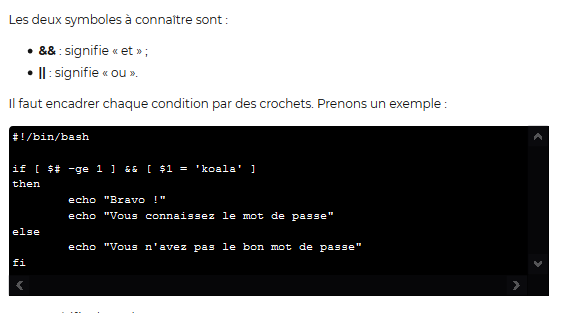


#### Comparaison

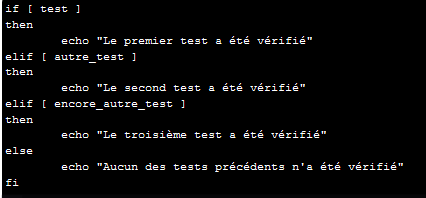


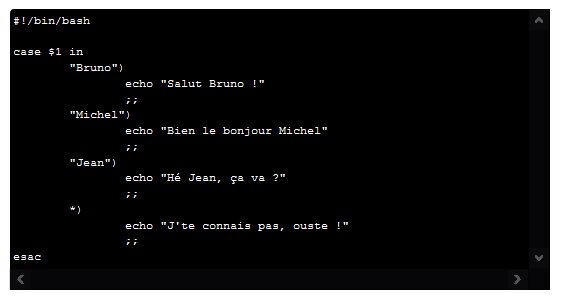


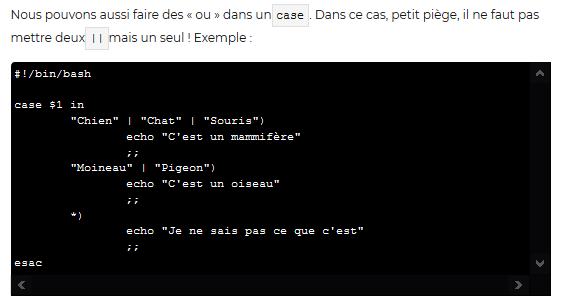




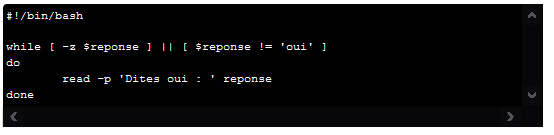
#### Condition IF et case

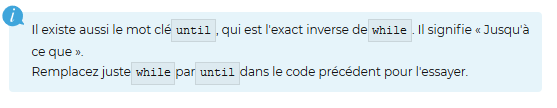






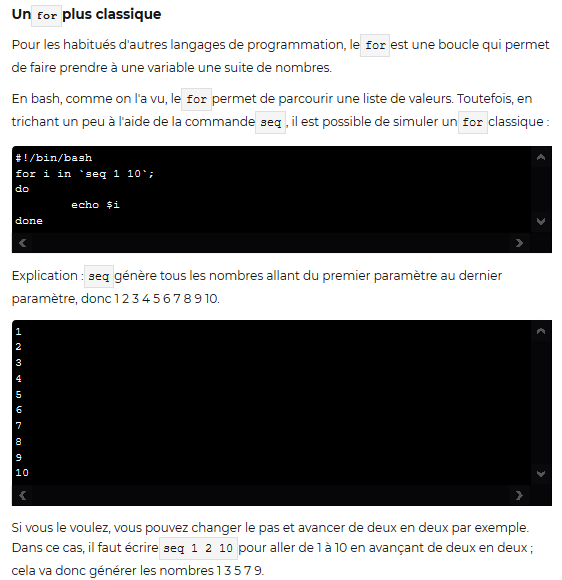
#### Boucle While, until



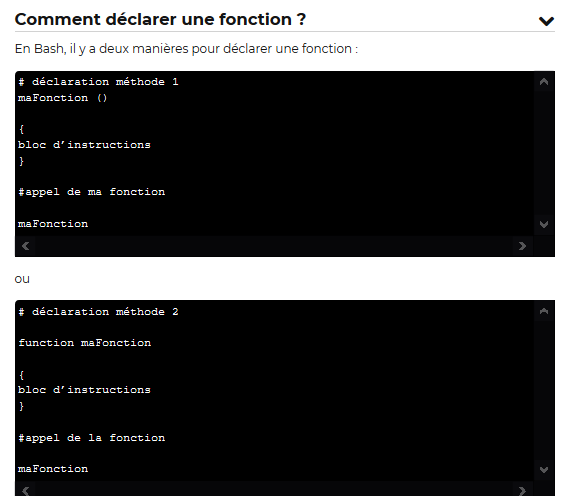


#### Boucle for





#### Les fonctions



# HTML & CSS

## Description

Le Web a été inventé par Tim Berners-Lee au début des années 1990.

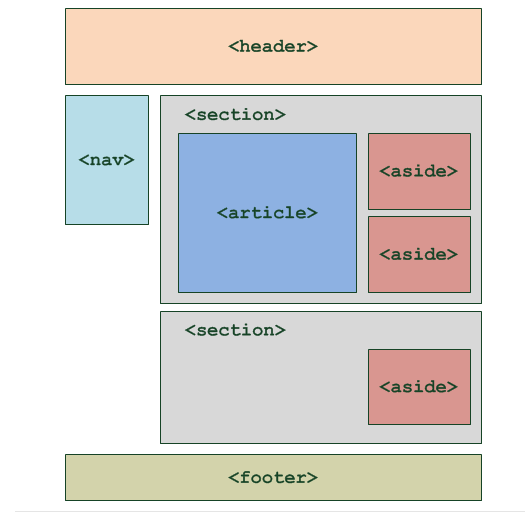
Pour créer des sites web, on utilise deux langages informatiques :

* HTML : permet d'écrire et organiser le contenu de la page (paragraphes, titres…) ;
* CSS : permet de mettre en forme la page (couleur, taille…).

Il y a eu plusieurs versions des langages HTML et CSS. Les dernières versions sont HTML5 et CSS3.

Pour écrire du code HTML ou CSS il faut un éditeur de texte comme Sublime Text.

Structure d’une page HTML



## Les balises et attribut

### Les balises blocs

< !DOCTYPE html> : Permet d’utiliser HTML 5 .

<**html**></**html**>  : Englobe tout le contenu de la page

<**head**></**head**> : cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !

<**header**></**header**> : La plupart des sites web possèdent en général un en-tête, appelé header en anglais. On y trouve le plus souvent un logo, une bannière, le slogan de votre site…

<**body**></**body**> : c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

<**footer**></**footer**> : À l'inverse de l'en-tête, le pied de page se trouve en général tout en bas du document. On y trouve des informations comme des liens de contact, le nom de l'auteur, les mentions légales, etc.

<**nav**></**nav**> : La balise<nav>doit regrouper tous les principaux liens de navigation du site. Vous y placerez par exemple le menu principal de votre site. Généralement, le menu est réalisé sous forme de liste à puces à l'intérieur de la balise<nav>

<**section**></**section**> : La balise<section>sert à regrouper des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page.

<**aside**></**aside**> : La balise<aside>est conçue pour contenir des informations complémentaires au document que l'on visualise. Ces informations sont généralement placées sur le côté (bien que ce ne soit pas une obligation).

<**article**></**article**> : La balise<article>sert à englober une portion généralement autonome de la page. C'est une partie de la page qui pourrait ainsi être reprise sur un autre site. C'est le cas par exemple des actualités (articles de journaux ou de blogs).

<**p**></**p**> : incère un paragraphe.

<**h1**>< /**h1**> : Titre. Situé dans le body. Peut aller de h1 à h6.du plus important au moins important

<**ul**></**ul**> : crée une liste non ordonnée.

<**ol**></**ol**> : crée une liste ordonnée.

<**figure**></**figure**> : crée une figure.

<**figcaption**></ **figcaption** > : crée une légende dans une figure.

<**style**></ **style** > : permet d’insérer du CSS.

<**div**></ **div** > : permet d’englober certaine chose.

<**audio**></**audio**> : incère un fichier audio (du texte peut être inséré afin d’afficher un message si le format n’est pas supporté)

<**vidéo**></**vidéo**> : incère un fichier vidéo (du texte peut être inséré afin d’afficher un message si le format n’est pas supporté)

<**source**> : sert à proposer différents format audio ou vidéo.

### Les balises lignes

< !-- Ceci est un commentaire -->

<**titre**>< /**titres**> : Titre de la page. Situé dans le head.

<**em**>< /**em**> : met un peu en valeur du texte.

<**strong**>< /**srong**> : met beaucoup en valeur du texte.

<**mark**>< /**mark**> : fait ressortir du texte.

<**li**></**li**> : incère une puce pour les listes.

<**a**></**a**> : créer un lien vers une autre page html ou un id.





<**span**></**span**> : permet d’englober du texte.

### Les balises orphelines

<**img** /> : incère une image.

<**meta** /> : balise d’encodage

<**link**/> : incère un fichier CSS.



<**br** /> : permet d’aller à la ligne

### Les balises tableaux

<**table**>< /**table**> : indique le début et la fin d’un tableaux.

<**caption**>< /**caption**> : indique le titre d’un tableau.

<**thead**>< /**thead**> : indique l’en-tête d’un tableau.

<**tbody**>< /**tbody**> : indique le corps d’un tableau.

<**tfoot**>< /**tfoot**> : indique le bas d’un tableau.

<**th**>< /**th**> : indique les cellules d’en-tête d’un tableau.

<**tr**>< /**tr**> : indique le début et la fin d’une ligne d’un tableau.

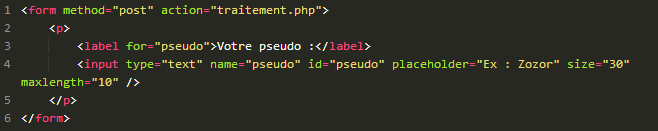
<**td**>< /**td**> : indique le début et la fin du contenu d’une cellule

### Les balises formulaires

<**form**>< /**form**> : indique le début et la fin d’un formulaire (doit mettre obligatoirement une balise block pour y faire figurer du texte)

<**label**>< /**label**> : indique la description d’avant saisie d’un formulaire.

<**input**>< /**input**> : indique le début de saisie de texte d’un formulaire.

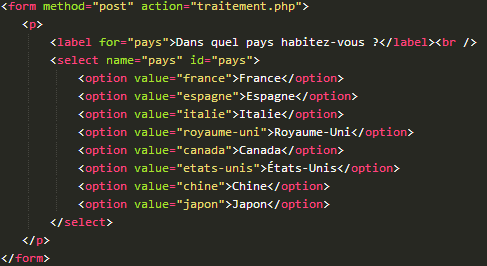




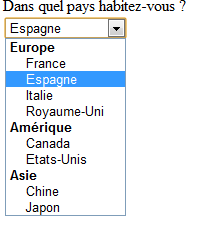
<**textarea**>< / **textarea** > : crée une zone de texte multiligne .

<**select**>< /**select**> : indique le début et la fin d’une liste déroulante.

<**option**>< /**option**> : permet d’insérer un choix dans une liste déroulante.

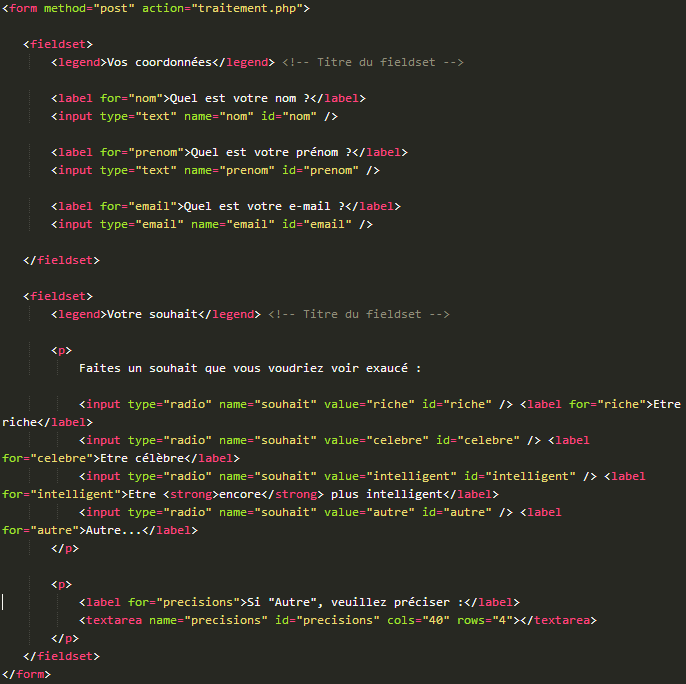


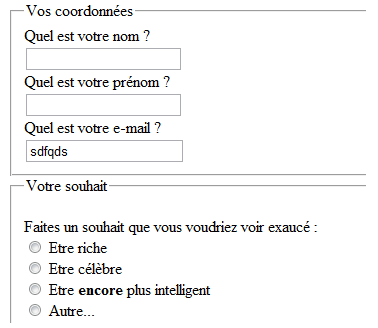
<**optgroup**>< / **optgroup** > : permet de séparer les options en fonction d’un groupe précis.



<**fieldset**>< / **fieldset** > : permet de regrouper les champs de saisie par thème .

<**legend**>< / **legend** > : titre du regroupement après une balise fieldset.





### Les attributs

Href= « » : lien vers ou ça retourne.

* **mailto : email** : ouvre la boite mail par défaut pour envoyer un mail.

title= « » : affiche une bulle au survol via la souris.

target= « » :

* **\_blank** : force l’ouverture d’un nouvel onglet.

class= « » : donne un nom de catégorie à une balise.

id= « » : donne un nom unique à une balise.

src= « » : pour l’insertion d’image, d’un fichier audio ou d’une vidéo.

alt= « » : pour nommer une image.

style= « » : insère du CSS.

colspan= « » : fusionne deux cellule de tableaux horizontalement.

rowpan= « » : fusionne deux cellule de tableaux verticalement.

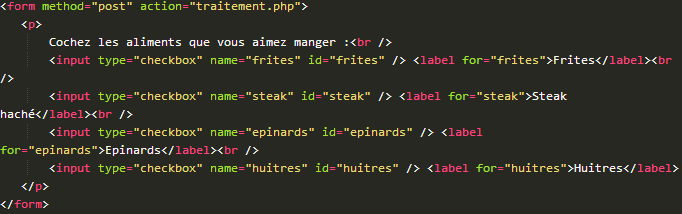
method= « » : indique par quel moyen les données vont être envoyé dans un formulaire.

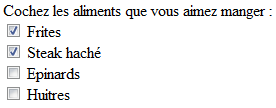
* **get** : donné envoyé dans l’adresse http limité à 255 caractères
* **post** : méthode la plus adapté permettant d’envoyer un grand nombre d’information.

action= « » : indique l’adresse de la page php qui va traiter les informations dans un formulaire.

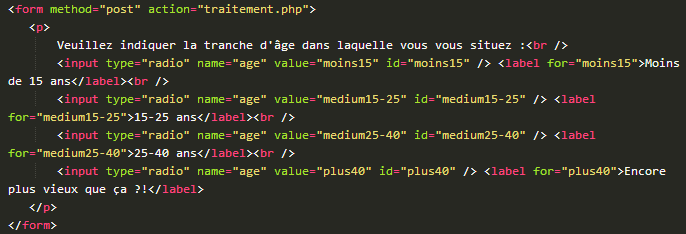
type= « » : indique le type de texte dans une balise input d’un formulaire.

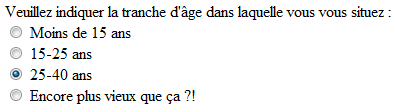
* **text** : permet la saisie de texte court.
* **Password**: permet la saisie de password.
* **Email :** permet la saisie d’email.
* **url:** permet la saisie d’url.
* **tel :** permet la saisie de numéro de tléphone.
* **number :** permet la saisie de nombre.
* **range :** permet la saisie d’un nombre via un curseur (s’utilise avec les attribut min,max,step)
* **color :** permet la saisie d’une couleur.
* **date :** permet la saisie d’une date.
* **search :** crée un champ de recherche.
* **checkbox :** crée une saisie de case à cocher.





* **radio :** crée une saisie unique d’une case à cocher parmi plusieurs propositions. ( l’attribut name doit être le même)





* **submit :** sert à l’envoie d’un formulaire.
* **reset :** remet à zéro le formulaire.
* **image :** sert à l’envoie d’un formulaire sous forme d’image.
* **button :** utiliser en Javasript pour exécuter une action sur la page.

checked : s’utilise dans les balise input d’un formulaire pour cocher par défaut une valeur.

name= « » : indique le nom du formulaire.

For = « » : s’utilise dans les balise label d’un formulaire pour associer à un attribut id d’une balise input .

size = « » : s’utilise dans les balise input d’un formulaire pour agrandir le champ de saisie.

maxlenght = « » : s’utilise dans les balise input d’un formulaire pour limiter le nombre de caractères.

value = « » : s’utilise dans les balise input d’un formulaire pour pré-remplir une valeur par défaut dans le champ de saisie.

placeholder = « » : s’utilise dans les balise input d’un formulaire pour donner une indication sur le champ de saisie.

row = « » : s’utilise dans les balise textarea d’un formulaire pour modifier le nombre de lignes qui peut être affichés

cols = « » : s’utilise dans les balise textarea d’un formulaire pour modifier le nombre de colonnes qui peut être affichés.

Value = « » : s’utilise dans les balise option et input d’un formulaire pour attribuer une valeur.

selected : s’utilise dans les balise option d’un formulaire pour attribuer une valeur par défaut.

label : s’utilise dans les balise optgroup d’un formulaire pour nommer les groupes d’une liste déroulante.

autofocus = « » : s’utilise dans les balise input d’un formulaire pour que le curseur au chargement de la page soit directement dans le champ de saisie.

required = « » : s’utilise dans les balise input d’un formulaire pour que le champs de saisie soit obligatoirement remplie avant envoie.

controls : s’utilise dans les balise audio et vidéo pour ajouter les boutons de lectures et de pause.

loop : s’utilise dans les balise audio et vidéo pour que la musique se joue en boucle.

autoplay : s’utilise dans les balise audio et vidéo pour que la musique se joue automatiquement au chargement de la page.

Preload = « » : s’utilise dans les balise audio et vidéo pour indiquer si la musique se précharge ou non dès le chargement de la page.

* **Auto  :** le navigateur décide.
* **metadata  :** charge uniquement les métadonnées.
* **none  :** ne pré télécharge pas.

poster : affiche une image de fond de vidéo avant que la vidéo commence.

### Le CSS

La mise en forme se fait :

* **Balise {}** : pour ajouter du css dans une seul balise.
* **Balise1, Balise2 {}** : pour ajouter du css dans une ou plusieurs balise séparé par une virgule.
* **.class {}** : pour ajouter du css pour tout une class.
* **.class :nth-child(numéro ) {}** : pour ajouter du css pour la N ième balise de la classe.
* **#id {}** : pour ajouter du css pour un id.
* **\* {}** : sélectionne toute les balises.
* **Balise1 Balise2 {}** : sélectionne toute les balise2 se trouvant dans balise1.
* **Balise1 + Balise2 {}** : sélectionne la première balise2 se trouvant dans balise1.
* **Balise1[attribut]**: sélectionne toute les balise1 contenant l’attribut : attribut.
* **Balise1[attribut= « Valeurs »]**: sélectionne toute les balise1 contenant l’attribut : attribut et la valeur exacte : Valeurs.
* **Balise1[attribut\*= « Valeurs »]**: sélectionne toute les balise1 contenant l’attribut : attribut et le mot : Valeurs.
* **Balise1 : hover :** au survol.
* **Balise1 : active :** lors du clic ou de la sélection.
* **Balise1 : focus :** lors du focus.
* **Balise1 : visited :** lorsque le lien a été visité.

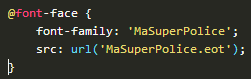
Dans le fichier CSS

/\* commentaire \*/

Color : indique une couleur ; ( en hexadécimale, rgb(255,255,255),rgba(255,255,255,1) )

Font-size : indique la taille du texte ; ( px, em, ex, small, medium ect )

Font-family : indique la police ; ( police1,police2, ect )

 pour télécharger sa police.

Font-style : normal, italic ;

Font-weight : bold ;

text-decoration : habille le texte ; ( blink, underline, line-through, overline )

text-align : positionne le texte ; ( left, center, right, justify )

float : permet de faire flotté une image dans du texte ; ( left, right)

clear : permet de stopper le float ; ( left, right, both )

**background :** permet d’habiller l’arrière-plan.

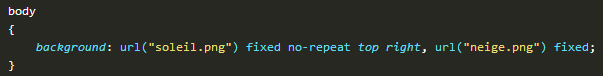
Background-color : donne une couleur à l’arrière-plan ; ( red, blue, ect )

Background-image : insère une image en arrière-plan ; ( url(« image ») )

Background-attachment : définie le style d’arrière-plan ; ( fixed, scroll )

Background-repeat : définie le style d’arrière-plan ; ( no-repeat ,repeat-x ,repeat-y, repeat )

Background-position : définie la position d’arrière-plan ; ( px px, top bottom left center right ect)



opacity : définie la transparence ; ( entre 0 et 1 )

**border :** permet de mettre des bordures. ( border-top,border-bottom, border-left,border-right)

border-color : donne une couleur à la bordure ; ( red, blue, ect )

border-style : insère un style de bordure; ( none, solid,dotted,dashed,double,groove,ridge,inset,outset )

Background-width : définie la largeur de la bordure ; ( px )



Border-radius : définie une bordure arrondie ; ( GaucheHpx DroiteHpx GaucheBpx DroiteBpx, px/px )

Box-shadow : définie une ombre ; ( DecalageHpx DecalageVpx Adoucissementpx couleur inset )

text-shadow : définie une ombre ; ( DecalageHpx DecalageVpx Adoucissementpx couleur inset )

width : définie la largeur d’un bloc ; (px,%)

min-width : définie la largeur minimal d’un bloc ; (px,%)

max-width : définie la largeur maximal d’un bloc ; (px,%)

height : définie la hauteur d’un bloc ; (px,%)

min-height : définie la hauteur minimal d’un bloc ; (px,%)

max- height : définie la hauteur maximal d’un bloc ; (px,%)

max- height : définie la hauteur maximal d’un bloc ; (px,%)

**padding :** définie la marge intérieur d’un bloc  ( padding-top, padding -bottom, padding -left, padding -right)

padding : définie la marge intérieur d’un bloc ; (px)

**margin :** définie la marge extérieur d’un bloc  ( margin-top,margin-bottom, margin-left,margin-right)

margin : définie la marge extérieur d’un bloc ; (px, auto)

overflow : permet de définir le dépassement du texte dans un bloc ; (visible,hidden,scroll,auto)

word-wrap: permet de forcer la sécure des très longs mots ; (break-word)

dispplay : définie le type de balise ;( flex,block,inline-block,none ect )

flex-directtion : définie l’agencements des blocs ; ( row, column,row-reverse,column-reverse )

flex-wrap : définie le retour à la ligne ; (wrap, nowrap , wrap-reverse )

justify-content : définie l’alignement de l’axe principal horizontalement par défaut ; (flex-start, flex-end,center,space-betxeen,space-around )

align-item : définie l’alignement de l’axe secondaire vertical par défaut ; (flex-start, flex-end,center,strech,baseline )

align-self : définie l’alignement d’un seul éléments sur l’axe secondaire vertical par défaut ;(flex-start, flex-end,center,strech,baseline )

align-self : définie l’alignement d’un seul éléments sur l’axe secondaire vertical par défaut ;(flex-start, flex-end,center,strech,baseline )

align-content : définie l’alignement de plusieurs ligne ; (flex-start, flex-end,center,space-betxeen,space-around,stretch )

order : Change l’ordre des éléments ( chiffre )

flex : fait grossir l’élément ; ( chiffre )

vertical-align : modifie l’alignement vertical des éléments ; ( baseline,top,middle,bottom,px,% )

position : modifie le type de positionnement ; ( absolute, fixed, relative)

left : modifie le positionnement à gauche; ( px )

right : modifie le positionnement à droite; ( px )

top : modifie le positionnement en haut; ( px )

bottom : modifie le positionnement en bas; ( px )

z-index : indique quel élément doit apparaître au-dessus; ( chiffre )

border-collapse : sert à coller les bordure d’un tableau entre elles; ( collapse,separate )

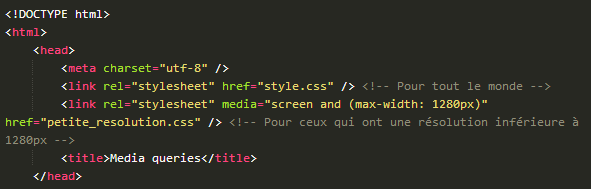
Vérifier la validité de son code HTML : [http://validator.w3.org](https://validator.w3.org).

## Les média Query

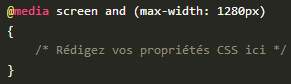
### Explication

Pour charger du CSS différent en fonction de la taille de l’écran afin d’adapter votre site sur toute les plateformes, vous pouvez :

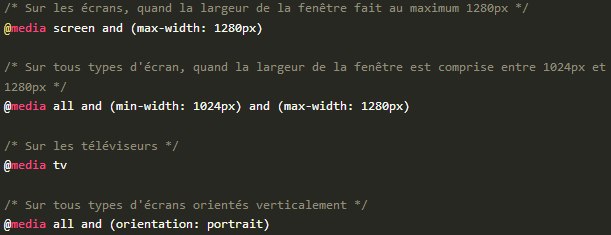
* Soit ajouter une balise link dans le code html qui renvoie à un autre fichier en fonction de la taille de l’écran.

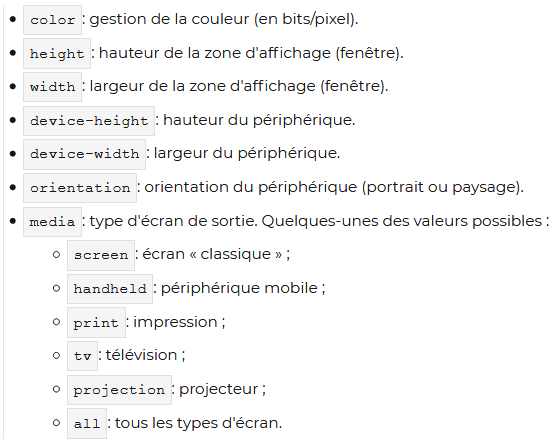


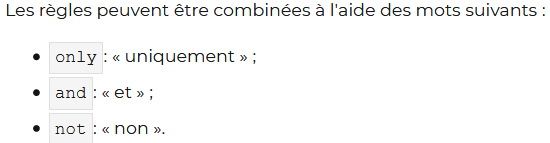
* Soit ajouter le code CSS directement dans votre fichier CSS



### Les règles







## Mémento

### Les balises.

Balises de premier niveau

| **Balise** | **Description** |
| --- | --- |
| <html> | Balise principale |
| <head> | En-tête de la page |
| <body> | Corps de la page |

#### Balises d'en-tête

| **Balise** | **Description** |
| --- | --- |
| <link /> | Liaison avec une feuille de style |
| <meta /> | Métadonnées de la page web (charset, mots-clés, etc.) |
| <script> | Code JavaScript |
| <style> | Code CSS |
| <title> | Titre de la page |

#### Balises de structuration du texte

| **Balise** | **Description** |
| --- | --- |
| <abbr> | Abréviation |
| <blockquote> | Citation (longue) |
| <cite> | Citation du titre d'une œuvre ou d'un évènement |
| <q> | Citation (courte) |
| <sup> | Exposant |
| <sub> | Indice |
| <strong> | Mise en valeur forte |
| <em> | Mise en valeur normale |
| <mark> | Mise en valeur visuelle |
| <h1> | Titre de niveau 1 |
| <h2> | Titre de niveau 2 |
| <h3> | Titre de niveau 3 |
| <h4> | Titre de niveau 4 |
| <h5> | Titre de niveau 5 |
| <h6> | Titre de niveau 6 |
| <img /> | Image |
| <figure> | Figure (image, code, etc.) |
| <figcaption> | Description de la figure |
| <audio> | Son |
| <video> | Vidéo |
| <source> | Format source pour les balises<audio>et<video> |
| <a> | Lien hypertexte |
| <br /> | Retour à la ligne |
| <p> | Paragraphe |
| <hr /> | Ligne de séparation horizontale |
| <address> | Adresse de contact |
| <del> | Texte supprimé |
| <ins> | Texte inséré |
| <dfn> | Définition |
| <kbd> | Saisie clavier |
| <pre> | Affichage formaté (pour les codes sources) |
| <progress> | Barre de progression |
| <time> | Date ou heure |

#### Balises de listes

| **Balise** | **Description** |
| --- | --- |
| <ul> | Liste à puces, non numérotée |
| <ol> | Liste numérotée |
| <li> | Élément de la liste à puces |
| <dl> | Liste de définitions |
| <dt> | Terme à définir |
| <dd> | Définition du terme |

#### Balises de tableau

| **Balise** | **Description** |
| --- | --- |
| <table> | Tableau |
| <caption> | Titre du tableau |
| <tr> | Ligne de tableau |
| <th> | Cellule d'en-tête |
| <td> | Cellule |
| <thead> | Section de l'en-tête du tableau |
| <tbody> | Section du corps du tableau |
| <tfoot> | Section du pied du tableau |

#### Balises de formulaire

| **Balise** | **Description** |
| --- | --- |
| <form> | Formulaire |
| <fieldset> | Groupe de champs |
| <legend> | Titre d'un groupe de champs |
| <label> | Libellé d'un champ |
| <input /> | Champ de formulaire (texte, mot de passe, case à cocher, bouton, etc.) |
| <textarea> | Zone de saisie multiligne |
| <select> | Liste déroulante |
| <option> | Élément d'une liste déroulante |
| <optgroup> | Groupe d'éléments d'une liste déroulante |

#### Balises sectionnantes

| **Balise** | **Description** |
| --- | --- |
| <header> | En-tête |
| <nav> | Liens principaux de navigation |
| <footer> | Pied de page |
| <section> | Section de page |
| <article> | Article (contenu autonome) |
| <aside> | Informations complémentaires |

#### Balises génériques

| **Balise** | **Description** |
| --- | --- |
| <span> | Balise générique de type inline |
| <div> | Balise générique de type block |

### Les propriétés CSS

#### Propriétés de mise en forme du texte

| **Propriété** | **Valeurs (exemples)** | **Description** |
| --- | --- | --- |
| font-family | *police1, police2, police3*, serif, sans-serif, monospace | Nom de police |
| @font-face | *Nom et source de la police* | Police personnalisée |
| font-size | 1.3em, 16px, 120%... | Taille du texte |
| font-weight | bold, normal | Gras |
| font-style | italic, oblique, normal | Italique |
| text-decoration | underline, overline, line-through, blink, none | Soulignement, ligne au-dessus, barré ou clignotant |
| font-variant | small-caps, normal | Petites capitales |
| text-transform | capitalize, lowercase, uppercase | Capitales |
| font | - | Super propriété de police. Combine : font-weight, font-style, font-size, font-variant, font-family. |
| text-align | left, center, right, justify | Alignement horizontal |
| vertical-align | baseline, middle, sub, super, top, bottom | Alignement vertical (cellules de tableau ou éléments inline-block uniquement) |
| line-height | 18px, 120%, normal... | Hauteur de ligne |
| text-indent | 25px | Alinéa |
| white-space | pre, nowrap, normal | Césure |
| word-wrap | break-word, normal | Césure forcée |
| text-shadow | 5px 5px 2px blue *(horizontale, verticale, fondu, couleur)* | Ombre de texte |

#### Propriétés de couleur et de fond

| **Propriété** | **Valeurs (exemples)** | **Description** |
| --- | --- | --- |
| color | *nom*, rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20... | Couleur du texte |
| background-color | *Identique à color* | Couleur de fond |
| background-image | url('image.png') | Image de fond |
| background-attachment | fixed, scroll | Fond fixe |
| background-repeat | repeat-x, repeat-y, no-repeat, repeat | Répétition du fond |
| background-position | *(x y)*, top, center, bottom, left, right | Position du fond |
| background | - | Super propriété du fond. Combine : background-image, background-repeat, background-attachment, background-position |
| opacity | 0.5 | Transparence |

#### Propriétés des boîtes

| **Propriété** | **Valeurs (exemples)** | **Description** |
| --- | --- | --- |
| width | 150px, 80%... | Largeur |
| height | 150px, 80%... | Hauteur |
| min-width | 150px, 80%... | Largeur minimale |
| max-width | 150px, 80%... | Largeur maximale |
| min-height | 150px, 80%... | Hauteur minimale |
| max-height | 150px, 80%... | Hauteur maximale |
| margin-top | 23px | Marge en haut |
| margin-left | 23px | Marge à gauche |
| margin-right | 23px | Marge à droite |
| margin-bottom | 23px | Marge en bas |
| margin | 23px 5px 23px 5px *(haut, droite, bas, gauche)* | Super-propriété de marge. Combine : margin-top, margin-right, margin-bottom, margin-left. |
| padding-left | 23px | Marge intérieure à gauche |
| padding-right | 23px | Marge intérieure à droite |
| padding-bottom | 23px | Marge intérieure en bas |
| padding-top | 23px | Marge intérieure en haut |
| padding | 23px 5px 23px 5px *(haut, droite, bas, gauche)* | Super-propriété de marge intérieure. Combine : padding-top, padding-right, padding-bottom, padding-left. |
| border-width | 3px | Épaisseur de bordure |
| border-color | *nom*, rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20... | Couleur de bordure |
| border-style | solid, dotted, dashed, double, groove, ridge, inset, outset | Type de bordure |
| border | 3px solid black | Super-propriété de bordure. Combine border-width, border-color, border-style. Existe aussi en version border-top, border-right, border-bottom, border-left. |
| border-radius | 5px | Bordure arrondie |
| box-shadow | 6px 6px 0px black *(horizontale, verticale, fondu, couleur)* | Ombre de boîte |

#### Propriétés de positionnement et d'affichage

| **Propriété** | **Valeurs (exemples)** | **Description** |
| --- | --- | --- |
| display | block, inline, inline-block, table, table-cell, none... | Type d'élément (block, inline, inline-block, none…) |
| visibility | visible, hidden | Visibilité |
| clip | rect (0px, 60px, 30px, 0px) *rect (haut, droite, bas, gauche)* | Affichage d'une partie de l'élément |
| overflow | auto, scroll, visible, hidden | Comportement en cas de dépassement |
| float | left, right, none | Flottant |
| clear | left, right, both, none | Arrêt d'un flottant |
| position | relative, absolute, static | Positionnement |
| top | 20px | Position par rapport au haut |
| bottom | 20px | Position par rapport au bas |
| left | 20px | Position par rapport à la gauche |
| right | 20px | Position par rapport à la droite |
| z-index | 10 | Ordre d'affichage en cas de superposition. La plus grande valeur est affichée par-dessus les autres. |

#### Propriétés des listes

| **Propriété** | **Valeurs (exemples)** | **Description** |
| --- | --- | --- |
| list-style-type | disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none | Type de liste |
| list-style-position | inside, outside | Position en retrait |
| list-style-image | url('puce.png') | Puce personnalisée |
| list-style | - | Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image. |

#### Propriétés des tableaux

| **Propriété** | **Valeurs (exemples)** | **Description** |
| --- | --- | --- |
| border-collapse | collapse, separate | Fusion des bordures |
| empty-cells | hide, show | Affichage des cellules vides |
| caption-side | bottom, top | Position du titre du tableau |

#### Autres propriétés

| **Propriété** | **Valeurs (exemple)** | **Description** |
| --- | --- | --- |
| cursor | crosshair, default, help, move, pointer, progress, text, wait, e-resize, ne-resize, auto... | Curseur de souris |

# Langage PHP

## Introduction

**PHP** : c'est un langage que seuls les serveurs comprennent et qui permet de rendre votre site dynamique. C'est PHP qui « génère » la page web comme on l'a vu sur un des schémas précédents.

Parmi les concurrents de PHP, on peut citer les suivants :

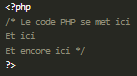
* ASP .NET : conçu par Microsoft, il exploite le framework (c'est-à-dire un ensemble de bibliothèques qui fournissent des services pour les développeurs) .NET bien connu des développeurs C#. Ce langage peut être intéressant si vous avez l'habitude de développer en C# .NET et que vous ne voulez pas être dépaysés.
* Ruby on Rails : très actif, ce framework s'utilise avec le langage Ruby et permet de réaliser des sites dynamiques rapidement en suivant certaines conventions.
* Django : il est similaire à Ruby on Rails, mais il s'utilise en langage Python.Java et les JSP (Java Server Pages) : plus couramment appelé « JEE » ou « Java EE », il est particulièrement utilisé dans le monde professionnel. Il demande une certaine rigueur. La mise en place d'un projet JEE est traditionnellement un peu plus longue et plus lourde mais le système est apprécié des professionnels et des institutions. C'est ce qui est utilisé sur le site des impôts français, par exemple

**MySQL** : c'est ce qu'on appelle un SGBD (Système de Gestion de Base de Données). Pour faire simple, son rôle est d'enregistrer des données de manière organisée afin de vous aider à les retrouver facilement plus tard. C'est grâce à MySQL que vous pourrez enregistrer la liste des membres de votre site, les messages postés sur le forum, etc. Le langage qui permet de communiquer avec la base de données s'appelle le SQL. Voici un code en langage SQL.

Parmi les concurrents de MySQL, je vous conseille de connaître (au moins de nom) les suivants :

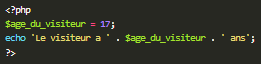
* Oracle : c'est le SGBD le plus célèbre, le plus complet et le plus puissant. Il est malheureusement payant (et cher), ce qui le réserve plutôt aux entreprises qui l'utilisent déjà massivement. Il existe cependant des versions gratuites d'Oracle, notamment pour ceux qui veulent apprendre à s'en servir.
* MariaDB : variante libre de MySQL, qui a été créée depuis que ce dernier a été racheté par... Oracle. Oui, je sais, on peut penser que ce monde est fou ! MySQL appartient à Oracle, mais reste bien une base de données à part. MariaDB est une copie (fork) de MySQL qui a la volonté de rester libre et indépendante.
* Microsoft SQL Server : édité par Microsoft, on l'utilise souvent en combinaison avec ASP .NET, bien qu'on puisse l'utiliser avec n'importe quel autre langage. Il est payant, mais il existe des versions gratuites limitées.
* PostgreSQL : il s'agit d'un SGBD libre et gratuit comme MySQL, qui propose des fonctionnalités plus avancées. Parfois comparé à Oracle, il lui reste cependant du chemin à parcourir. Il dispose d'une communauté un peu moins importante que MySQL et Oracle. OpenClassrooms utilise PostgreSQL.

Voici la forme d’une balise PHP :



## Les variables

Pour déclarer une variable il suffit de mettre le symbole « $ » suivis du nom de la variable, exemple :

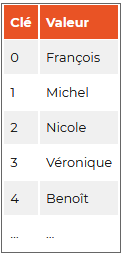


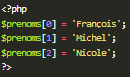
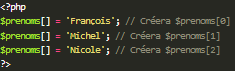
Il n y a pas besoin de préciser le type ( Int, float,string ect )

Les tableaux

Déclarer un tableaux :

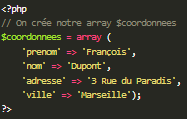
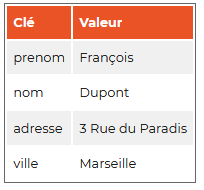
Pour créer un tableau numéroté en PHP, on utilise généralement la fonction array

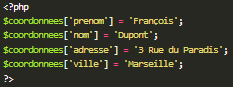


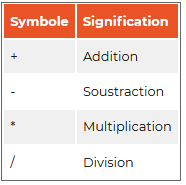


Déclarer un tableau associatif :

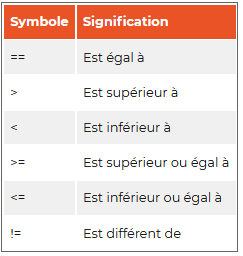
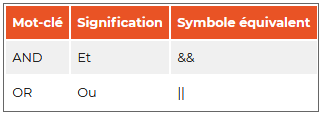




Les calculs

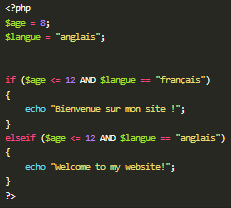
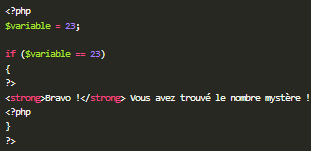
 

Les symboles de comparaison et conditions multiples

## Condition IF, case et ternaires

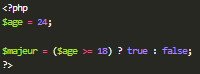
### IF,Elseif , Else

### SWITCH/CASE

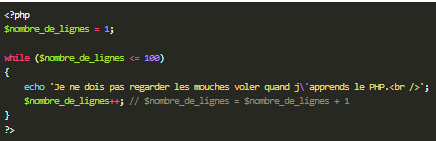


### Conditions Ternaires

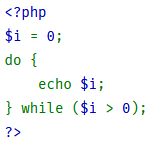
La condition testée est $age >= 18. Si c'est vrai, alors la valeur indiquée après le point d'interrogation (ici true) sera affectée à la variable $majeur. Sinon, c'est la valeur qui suit le symbole « deux-points » qui sera affectée à $majeur.

## Boucle While, until, for

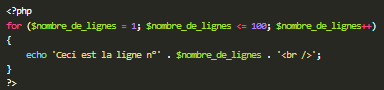
### Boucle While

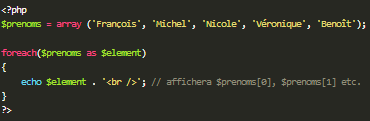


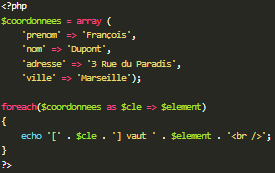
### Boucle do-While

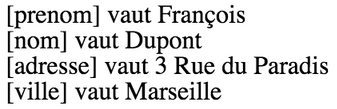
la différence avec une boucle While est que les instructions dans la boucle seront exécuter au moins une fois.

### Boucle For



Permet de parcourir un tableaux la valeur elements prend la valeur de la variable 0 puis 1 puis 2 Ect à chaque tour de boucle





## Les tableaux

## Les fonctions PHP

Phpinfo() ; Pour connaitre l'ensemble des informations relatives au PHP utilisé par le serveur web.

Echo « Texte » ; permet d’afficher du texte.

Array\_key\_exists(‘cle’,tableaux) : pour vérifier si une clé existe dans l'array ;

in\_array(‘valeur’,tableaux) : pour vérifier si une valeur existe dans l'array ;

array\_search(‘valeur’,tableaux) : pour récupérer la clé d'une valeur dans l'array ;