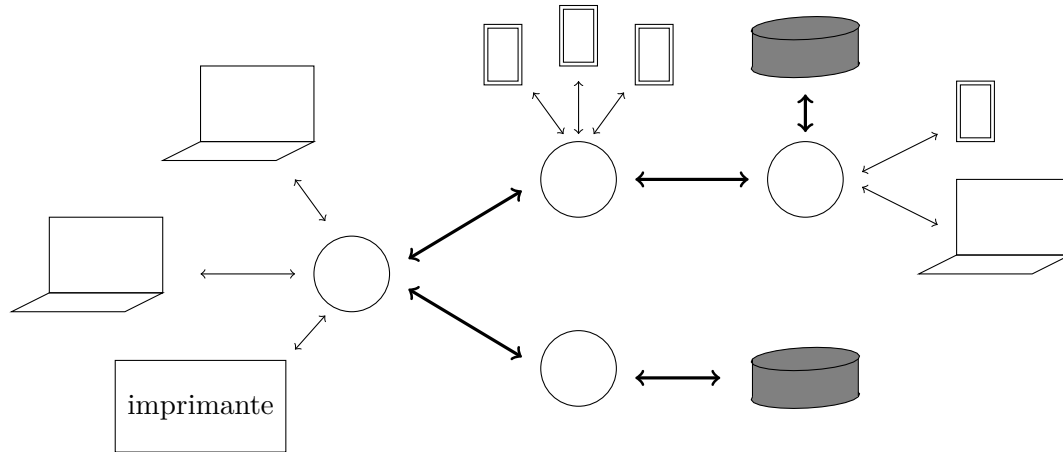


I Initiation au réseau

1 Vue succincte

Un réseau numérique est un ensemble d'ordinateurs ou de matériels numériques connectés entre eux. C'est à dire qu'ils sont capables de communiquer.

expl : ci-dessous est représentée une situation mettant en jeu des ordinateurs, des téléphones, une imprimante, des serveurs et des routeurs. Spécifier chaque acteurs du schéma.



La communication se fait grâce à un langage (*e.g.* TCP/IP (Transmission Control Protocol / Internet Protocol)). Les messages transmis sont codés selon différents niveaux puis envoyés physiquement sur les réseaux (dans les câbles électriques ou sur les ondes wifi) pour ensuite être décodés et être interprétables par l'homme.

Selon le modèle TCP/IP, il existe 4 grands niveaux de codage.

Application (FTP, SMTP, HTML, etc.)	interprète les données
Transport (TCP)	vérifie la fiabilité de la communication
Internet (IP)	effectue le routage et remet en ordre les paquets
Accès réseau (ethernet & lien physique)	découpe et achemine les données

2 Commandes de vérification (linux // windows)

1. on peut vérifier la communication au niveau applicatif à l'aide de la commande `ping`.

```
ping www.duckduckgo.org
```

2. on peut connaître les paramètres réseau de sa propre machine à l'aide de : `ifconfig` // `ipconfig /all`.

3. si `ping` ne fonctionne pas, on peut vérifier la route suivie par les paquets des messages à l'aide de de la commande `tracpath` // `tracert`.

```
tracpath www.duckduckgo.org // tracert www.duckduckgo.org.
```

Exo 1 : dans un terminal // après avoir lancer l'interpréteur de commande (`cmd`),

◇ lancer une requête `ping` et noter l'adresse ip du site choisi ainsi que le temps mis par les paquets,

◇ lancer une requête `ifconfig` // `ipconfig` puis noter l'adresse ip de votre ordinateur et l'adresse MAC de la carte réseau.

3 Coût énergétique

Depuis 2009 (*e.g.* [LeMonde 2009/01/12]), plusieurs études ont essayé de rendre les résultats intelligibles au grand public.

On peut résumer le problème énergétique des nouvelles technologies par la maxime :

Ce n'est pas parce que l'on ne le voit pas que ça n'existe pas.

Comme le schéma le fait penser, chaque communication nécessite un transport de données d'un point à un autre (en moyenne un mail parcourt 15 000 km!) et à chaque étape une action numérique qui requiert de l'énergie pour s'effectuer (pensez en effet que vous devez recharger votre téléphone tous les jours).

On donne les valeurs moyennes équivalent carbone suivantes (ADEME 2017) :

- ◇ envoi d'un SMS coûte 3.10^{-6} g CO₂
- ◇ envoi d'un mail de 1Mo coûte 19g CO₂
- ◇ envoi d'une requête de recherche coûte 7g CO₂

Ainsi que deux comparaisons de coûts énergétiques ([coût énergétique 2017](#))

- ◇ 1 data-center nécessite l'énergie dépensée par 30 000 habitants européens (180 data-centers en France (2017))
- ◇ 600 000 heures de vidéos youtube sont ajoutées et stockées chaque jour (rq : 2,8 Md de vue sous youtube nécessite le fonctionnement d'une centrale nucléaire durant 1 an)
- ◇ lire une page numérique est moins coûteux que lire une page imprimée (2p/feuille en R/V) si on la lit en moins de 3min

Exo 2 : Sachant qu'un internaute français fait en moyenne 135 requêtes/mois ([ConsoGlobe 2017](#)) et qu'il y a près de 50M d'internautes en France.

1. Quel est le coût équivalent CO₂ en France par an ?
2. Il y a 3,3Md de requêtes sous google par jour, quel est le coût CO₂ ?
3. 12 milliards de courriels sont envoyés chaque heure dans le monde (2017), calculer le coût équivalent CO₂ (rq : ces 12Md d'envois nécessitent le fonctionnement de 18 centrales nucléaires pendant cette heure).
4. Sachant 1 tonne équivalent CO₂ équivaut à 1 A/R Paris-New-York ou 14 000 km en ville par une twingo, quelles distances les 3 réponses précédentes représentent-elles ?
5. Évaluer votre propre impact (sms, requêtes, mail, photos (1 photo ≈ 3 mails)). Si les 8 milliards d'abonnements dans le monde ([JournalNet 2017](#)) avaient la même utilisation que vous, quel en serait le coût mondial en CO₂ et nombre d'A/R Paris-NY ?

Vous pouvez compléter ces informations au travers des sites [ademe](#) et [greenit](#).

Exo 3 : proposer des solutions pour diminuer l'impact énergétique dans notre utilisation journalière (autres idées [greenit 2015](#))

- ...
- ...
- ...
- ...
- ...
- ...

II Le PHP : des pages HTML dynamiques

Le PHP (acronyme de Hypertext PreProcessor) est un langage de programmation, créé en 1995 par Rasmus Lerdorf, qui génère des pages HTML à la volée et a acquis ses lettres de noblesses en même temps que l'explosion d'internet vers 1997.

L'idée est de rendre dynamique les pages HTML en les générant *in situ* par un serveur internet selon un code fourni. L'utilisateur qui lit les pages HTML ne lira que le résultat du code PHP mais ne voit jamais le code PHP.

Pour faire fonctionner ce système, il faut utiliser un serveur. Néanmoins, notre propre ordinateur peut imiter le fonctionnement d'un serveur grâce au logiciel EasyPHP. Lancer EasyPHP.

EasyPHP permet de créer un serveur fictif sur notre propre ordinateur et ainsi générer les pages HTML de manière dynamique. Pour se faire, il faut que les fichiers HTML ou PHP soient dans le répertoire `www` (ou `/data/localweb`) d'EasyPHP. Pour visualiser nos fichiers, en bas à droite de l'écran, sélectionner Weblocal.

Afin d'écrire du code PHP, on crée un fichier `.php` dans lequel on écrit une page HTML. Puis il faut alors ouvrir une balise PHP, insérer son code PHP (chaque ligne se termine par `;`) et refermer la balise.

 on ne peut pas valider du code PHP via [<https://html5.validator.nu/>] mais on doit valider le résultat généré par le code PHP.

expl :

```
<!doctype html>
<html lang="fr-FR">
<head>
<meta charset="UTF-8"/>
<title> Ma premiere page dynamique </title>
</head>
<body>
<?php
$n = 3;
$m = 5;
echo "<h1> Ma première page PHP</h1>";
echo "qui calcule la somme de ".$n." et ".$m."<br>";
echo $n+$m;
?>
</body>
</html>
```

Exo 4 : Créer une page .HTML avec ce code et interprétez-le à l'aide d'EasyPHP.
On remarque une grande similitude entre python et PHP, voyons plus en détails.

1 variables et test conditionnel

python

```
i = 4
msg = "i est pair"
if i%2 :
    print(msg)
else :
    print "i est impair"
```

PHP

```
<?php
$i = 4;
$msg = "i est pair";
if ( $i % 2 ){
    echo $msg;
} else {
    echo "i est impair";
}
?>
```

On note que

- ◇ les variables commencent toujours par \$,
- ◇ les conditions sont entre parenthèses (**condition**),
- ◇ les blocs d'instructions ne sont pas nécessairement indentés (il faut néanmoins garder cette habitude) mais sont nécessairement compris entre accolades **{instructions}**.

2 boucles

python

```
i = 0
while i < 4 :
    i = i+1
    print("i vaut "+str(i))
```

```
for i in range(3) :
    print("i vaut "+str(i))
```

PHP

```
<?php
$i = 0;
while ( $i < 4 ){
    $i = $i + 1;
    echo "i vaut $i </br>";
}
?>
```

```
<?php
for ($i = 0; $i < 3; $i++) {
    echo "$i <br>";
}
?>
```

⚠ À noter que sous PHP, les chaînes de caractères entre double guillemets sont interprétées mais pas celles entre simple guillemets. Ainsi avec `$i=3`,

code PHP	"valeur \$i"	'valeur \$i'
résultat	valeur 3	valeur \$i

3 Saisies et traitements

Afin de pouvoir saisir des valeurs et faire des actions en conséquences à l'aide de PHP, il faut associer une page '.php' à la page HTML contenant la saisie des valeurs.

Chaque variable entrée sera alors transmise à la page PHP qui en fera le traitement et créera une page HTML réponse en conséquence.

Il faut donc créer au minimum **2 pages** avec un lien entre elles (*cf. action=...*).

expl : pageSaisies.html

```
<body>
Veillez vous présenter s'il vous plaît.
<form action='pageTraitement.php' method='post'> <!--traité via pageTraitement.php-->
Nom <input type='text' name='nom' /><br>
<input type='radio' name='civilite' value="H"/> M.
<input type='radio' name='civilite' value="F"/> Mme <br>
<input type='submit' value="valider" />
</form>
</body>
```

pageTraitement.php

```
<body>
Bonjour
<?php
$varnom = $_POST['nom'];
$varciv = $_POST['civilite']; // avec des noms de variables explicites
if ( $varciv == "H" ) {
    echo "M. $varnom!";
} else {
    echo "Mme $varnom!";
}
?>
</body>
```

rq : on peut passer les données à l'aide de la méthode GET mais cela laisse visibles les données dans l'URL et ouvre ainsi d'immenses failles de sécurité. On n'utilisera ainsi que la méthode POST.

Exo 5 : créer 2 fichiers avec ces codes dans le répertoire **www** et tester.

Il existe de nombreux autres champs de saisie, en voici quelques uns.

<i>Champ de saisie</i>	<input type="text" name="nom"/>
<i>Bouton de déclenchement</i>	<input type="submit" value="valider"/>
<i>Choix exhaustif</i>	<input type="radio" name="civilite" value="H"/> <input type="radio" name="civilite" value="F"/>
<i>Case à cocher</i>	<input type="checkbox" name="coche"/>
<i>Bouton de remise à zéro</i>	<input type="reset" name="annuler"/>
<i>Variables cachées</i>	<input type="hidden" name="secret" value=21/>
<i>Menu déroulant</i>	<select name="couleur"> <option value="0000ff"> bleu </option> <option value="ff0000"> rouge </option> <option value="00ff00"> vert </option> </select>

Exo 6 : Continuez votre page en ajoutant des éléments dynamiques vus dans ce chapitre.

rq : vous pouvez trouver un tutoriel sur internet, e.g. [<http://www.w3schools.com/php/default.asp>].