QU'EST-CE QUE LE PHP ?

APPLICATION À LA BIOINFORMATIQUE

Mostafa KRIAT

HTML, présenté dans le numéro précédent de L'OPÉRON, permet la création de pages lisibles sur internet. Des actions limitées sont possibles comme l'ouverture de nouvelles pages, l'envoi de courriels... Mais cette interactivité est limitée : une action dépendant de paramètres fournis par l'utilisateur n'est pas possible car il faut, derrière la page, un moteur, un langage de programmation permettant d'exécuter des actions, comme le calcul du prix d'une commande sur le site upbm... Un des langages possibles est PHP dont l'utilisation, avec des étudiants, est tout à fait envisageable dans le cadre de la bioinformatique en BTS/DUT.
Il nous est apparu intéressant de présenter des outils de création de pages internet qui peuvent être d'intérêt pour les étudiants, soit parce qu'ils peuvent les utiliser, soit parce qu'ils en tirent des connaissances sur la façon dont l'outil peut fonctionner, ce qui leur permet une intervention subtile. Un usage pédagogique de ces outils est possible, et nous essayerons de le montrer.

L'enseignement de la bioinformatique offre une occasion intéressante pour s'initier à la programmation. Les outils de base d'analyse des séquences sont essentiellement basés sur le traitement des chaînes de caractères représentées par les séquences de nucléotides et d'acides aminés.

1. LE CHOIX DU LANGAGE DE PROGRAMMATION

L'analyse des séquences à l'aide de l'outil informatique nécessite le choix d'un langage de programmation. Il existe deux grandes familles de langage de programmation en informatique :
- les langages compilés,
- les langages interprétés ou langages de scripts.

La différence majeure entre ces deux types se situe principalement au niveau de leur exécution.
Les programmes écrits en langage compilé (exemples : C, C++, Pascal) nécessitent une compilation du code qui consiste en une traduction du code en langage binaire et donc proche du langage machine. Cette étape, réalisée une seule fois par le compilateur, permet une exécution rapide du programme.
Dans le cas des programmes écrits en langage interprété (exemples : PHP, Perl, Javascript), le code ou script est lu au fur et à mesure par un interpréteur qui le transforme en instructions compréhensibles par le processeur. La lenteur de l'exécution des programmes écrits avec ce type de langage est largement compensée par la puissance des machines actuelles et la simplicité de l'environnement du développement. En effet un simple éditeur de texte tel que le bloc notes suffit pour écrire un programme.
Le langage PHP "*Hypertext PreProcessor*" est un langage de programmation interprété qui a été développé spécialement pour le web. C'est un projet "*Open Source*" ouvert, gratuit et qui bénéficie de l'apport d'une communauté de développeurs très active dans le monde.

Cas de Java (différent de Javascript) : c'est un langage très utilisé actuellement car il est indépendant de la machine. En effet, le compilateur crée un code dépendant du microprocesseur et du système d'exploitation : le code applicable sous Windows ne fonctionnera pas sous Unix ou MacOs. Un programme écrit en Java est compilé mais le code obtenu est interprété par une machine java virtuelle installée sur l'ordinateur, qui fabrique un code utilisable et donc spécifique du système d'exploitation. Un programme comme OpenOffice utilise Java.

Mostafa KRIAT
Lycée Polyvalent Astier
Quartier Roqua
07200 AUBENAS
Courriel : mkriat@yahoo.com

2. L'ENVIRONNEMENT DE TRAVAIL AVEC PHP

Le langage PHP est intimement lié au développement du Web. L'écriture et la mise au point de scripts PHP nécessitent :

- un **éditeur de texte** pour écrire les scripts (bloc notes). Il ne faut pas confondre éditeur de texte et logiciel de traitement de texte. Un éditeur permet de manipuler du texte au format brut (caractères alphanumériques) sans formatage.
- un **interpréteur PHP** qui vérifie la syntaxe du code et le traduit en langage machine compréhensible pour le processeur. On parle également de moteur PHP.
- un **serveur de pages web** qui renvoie les pages interprétées sous forme de pages HTML.
- un **navigateur** qui interprète le code HTML renvoyé par le serveur.

Pour comprendre le fonctionnement de l'environnement de travail, il est important de rappeler le fonctionnement du web dynamique basé sur la notion de client-serveur (voir figure).

Le poste client envoie une requête pour un fichier exemple bonjour.php vers le serveur Web. Celui-ci transfère le fichier demandé au moteur PHP qui vérifie la syntaxe et décode les lignes PHP. L'interpréteur transfère le fichier décodé le plus souvent au format HTML vers le serveur, qui le transfère à son tour au navigateur du poste client. Contrairement au javascript, autre langage interprété mais qui s'exécute au niveau du poste client, on note que le **langage PHP s'exécute du côté serveur**.

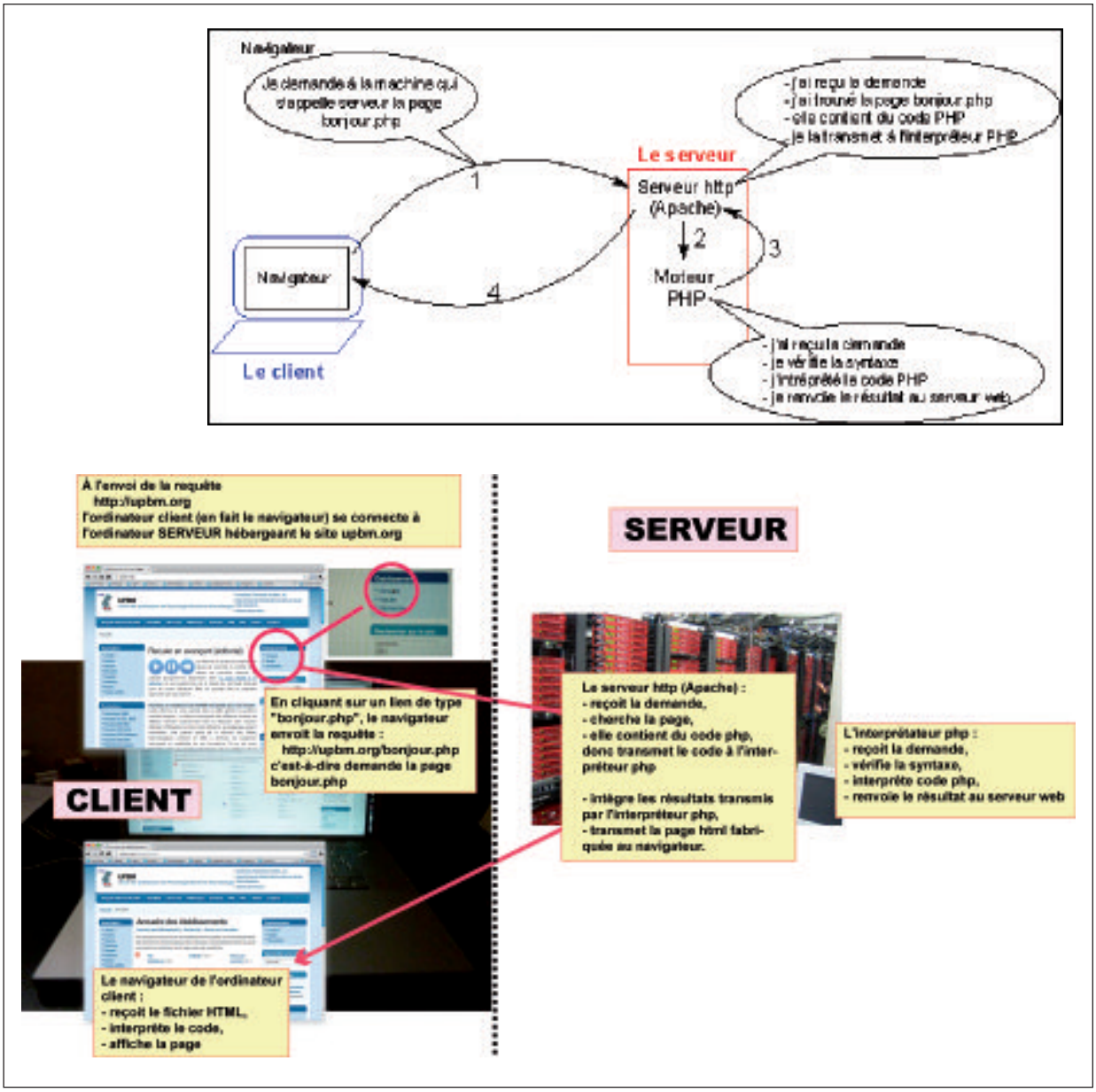


FIGURE - FONCTIONNEMENT DE PHP

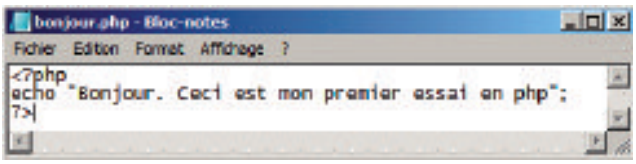
Tester ses productions PHP peut se faire en utilisant un serveur distant sur lequel sont déposés les fichiers utiles, mais il est très pratique de transformer son propre ordinateur en client et serveur à la fois. Sous Windows, il existe des solutions tout en un qui regroupent : **un serveur web (Apache), le moteur PHP et un gestionnaire de base de données MySQL**. C'est notamment le cas avec les outils tels que : EasyPHP, Xampp, Wampp… Les mêmes types d'outils existent sous Linux ou MacOs.

L'installation d'EasyPHP ne pose aucune difficulté ; pour interpréter les scripts réalisés, il suffit des les déposer dans un sous-dossier (exemple : mes_scripts) du dossier www crée par EasyPHP et les appeler à partir de son navigateur en local grâce à l'adresse du serveur local : http://localhost/mes_scripts/.
Il est à noter que la plupart des hébergeurs offrent un service complet pour interpréter les pages en PHP, il suffit simplement de déposer les scripts réalisées sur son espace et les appeler à travers un navigateur.

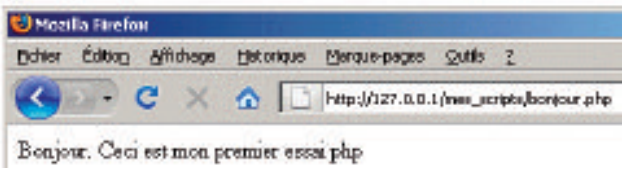
3. EXERCICES

■ EXERCICE 1 : INSTALLATION ET TEST

Une fois EasyPHP installé et lancé, nous allons créer un dossier spécialement dédié aux scripts réalisés : mes_scripts.
Pour tester le bon fonctionnement de l'installation, nous allons réaliser un premier script avec la commande php echo qui affiche une chaîne de caractères.



Script saisi dans le bloc-notes et enregistré dans le fichier ***bonjour.php***
À noter que la chaîne de caractères est encadrée par des guillemets (anglais) et la commande echo se termine par un point-virgule.



Résultat de l'exécution du script ***bonjour.php*** par le moteur php et affiché par le navigateur. Notez l'adresse du serveur local (dans la barre d'adresse avec http://**127.0.0.1**/….

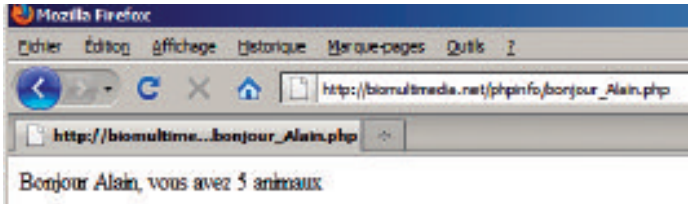
■ EXERCICE 2 : AFFICHER BONJOUR ALAIN ET FAIRE LA SOMME DES SES CHATS ET DE SES CHIENS

Dans tout langage de programmation les **variables** constituent les éléments principaux d'un programme. Une variable est une zone mémoire qui va contenir des valeurs qui peuvent changer au cours de l'exécution du programme. Il existe deux types de données contenues dans une variable php : les **nombres** et les **chaînes de caractères**. Le langage PHP n'est pas un langage typé, c'est à dire qu'il n'impose pas de d'associer un type de donnée à une variable.
Pour distinguer les variables, php leur donne un nom commençant par $ comme **$nom**.

Script saisi dans le bloc-notes et enregistré dans le fichier ***bonjour_Alain.php***

```
<?php
 $nom="Alain";
 $chats=2;
 $chiens=3;
 $total=$chats+$chiens;
 echo "Bonjour $nom, vous avez $total animaux";
?>
```

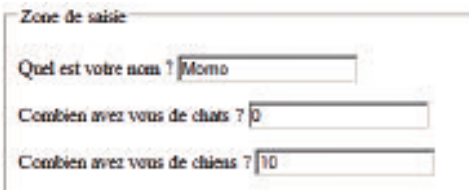
Notez que les variables contenant des nombres ne comportent pas de guillemets. Mettre les guillemets les transforment en chaîne de caractères…
La variable **$total** résulte d'une somme obtenue grâce à l'opérateur **+**.



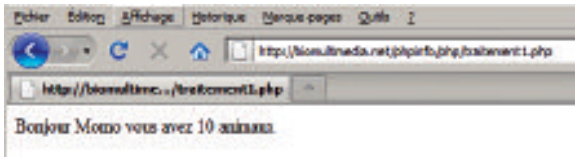
Résultat de l'exécution du script ***bonjour_Alain.php*** par le moteur php et affiché par le navigateur. Notez l'adresse du serveur en ligne **(http://biomultimedia.net/….**

■ EXERCICE 3 : RÉALISER UN AFFICHAGE DYNAMIQUE QUI TIENT COMPTE DES DONNÉES SAISIES PAR LE VISITEUR

Dans le cas précédent les valeurs affectées aux trois variables sont fixes. Si on souhaite laisser la possibilité à l'utilisateur de saisir les valeurs de son choix via le navigateur, on doit utiliser des **formulaires**. Grâce aux formulaires on peut envoyer des informations qui seront traitées par les scripts php.

Formulaire html de saisie ***bonjour_dynamique.html***	Commentaires
<form action="traitement1.php" method="post"> <fieldset> <legend>Zone de saisie</legend> Quel est votre nom ? <input name="nom" type="text" /> Combien avez vous de chats ? <input name="chats" type="text" /> Combien avez vous de chiens ? <input name="chiens" type="text" /> <input type="submit" name="Envoyer" /> </fieldset> </form>	Notez qu'une fois le bouton Envoyer sera cliqué c'est le script **bonjour2.php** qui sera exécuté sur le serveur
	Aspect de la page du formulaire dans le navigateur

Afin que le script puisse récupérer les valeurs des variables saisies, il est nécessaire de le préciser dans le script ***traitement1.php*** à l'aide de la variable d'environnement POST.

Script : ***traitement1.php***	Résultat du traitement des données saisies
<?php $nom=$_POST ['nom']; $chats=$_POST ['chats']; $chiens=$_POST ['chiens']; $total=$chats+$chiens; echo "Bonjour $nom vous avez $total animaux"; ?>	

■ EXERCICE 4 : ADAPTER LA RÉPONSE EN FONCTION DU NOMBRE TOTAL D'ANIMAUX

Dans le cas précédent la réponse peut être améliorée dans les cas où l'utilisateur a un seul animal ou n'a pas d'animaux. En effet si le total des animaux est égal à 0 ou à 1 la réponse ne sera pas judicieuse. On souhaite l'adapter et répondre :
- vous avez 1 animal si le total est égal à 1,
- vous n'avez pas d'animaux si le total est égal à 0.

Pour distinguer ces différentes réponses nous avons besoin de la structure conditionnelle if elseif.

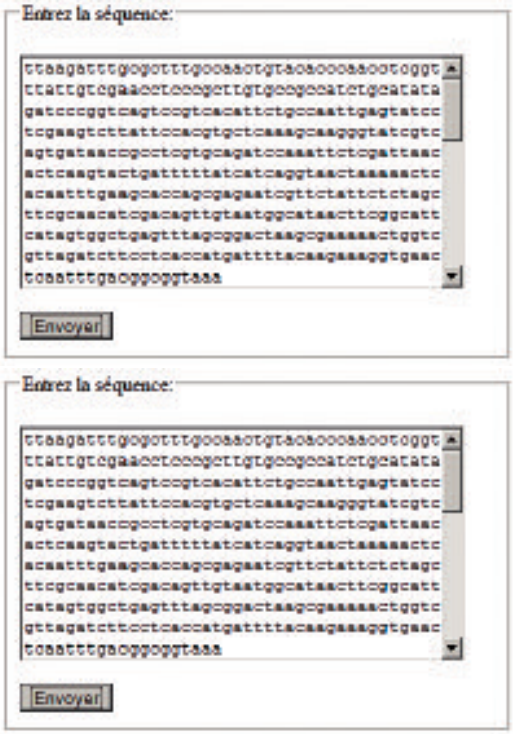
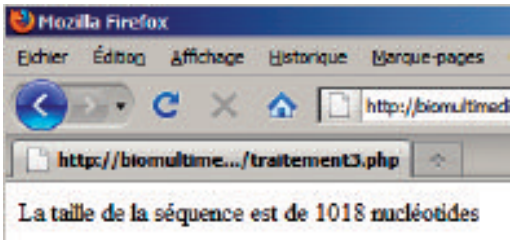
Script : ***traitement2.php***

```
<?php
        Dans un premier temps on définit les variables.
$nom=$_POST ['nom'];
$chats=$_POST ['chats'];
$chiens=$_POST ['chiens'];
$total=$chats+$chiens;
        On utilise ensuite la structure conditionnelle if elseif :
            Si le total est égal à 0
            alors Afficher le message 1
            sinon
                    si le total est égal à 1
                    alors Afficher le message 2
                    sinon afficher le message 3
if ($total = = 0)
        {
                echo "Bonjour $nom vous n'avez d'animaux";
        }
        elseif ($total = = 1)
                {
                echo "Bonjour $nom vous avez $total animal";
                }
                else
                        {
                        echo "Bonjour $nom vous avez $total animaux";
                        }
?>
```

Remarque : le test d'égalité utilise « = = » et non pas « = ». Un signe égal isolé permet, dans $nom="Mostafa"; l'affectation de la valeur de droite Mostafa dans le conteneur (variable) de gauche $nom.

EXERCICE 5 : DÉTERMINATION DE LA TAILLE D'UNE SÉQUENCE D'ADN

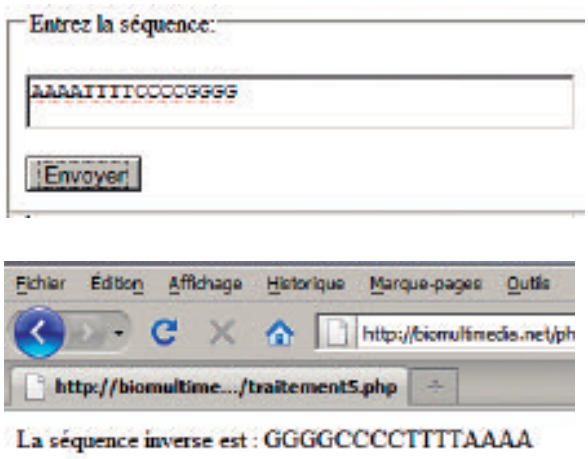
La détermination de la taille d'une chaîne de caractères est une opération de base en bioinformatique. La fonction php ***strlen*** permet de déterminer le nombre de caractères d'une chaîne.

Script : ***traitement3.php***	Commentaires et aperçu
Fichier formulaire de saisie html <form action="traitement3.php" method="post"> <fieldset> <legend>Entrez la séquence: </legend> <textarea name="sequence" cols="40" rows="10"></textarea> <input name="Envoyer" type="submit" value="Envoyer" /> </fieldset> </form> Fichier de traitement php <?php $sequence=$_POST ['sequence']; $taille =strlen($sequence); strlen est une fonction intégrée dans php qui calcule la taille d'une chaîne de caractères echo "La taille de la séquence est de $taille nucléotides"; ?>	Aperçu formulaire de saisie  Résultat de l'analyse 

EXERCICE 6 : DÉTERMINATION DE LA SÉQUENCE INVERSE

L'objectif est de changer l'orientation 5'->3' ou inversement. La fonction **strrev** permet de réaliser l'opération en une seule étape.

<?php
$sequence=$_POST ['sequence'];
$reverse =strrev($sequence);
echo "La séquence inverse est : $reverse";
?>



■ EXERCICE 7 : DÉTERMINATION DE LA SÉQUENCE COMPLÉMENTAIRE

L'objectif est de déterminer la séquence complémentaire d'une séquence d'ADN en utilisant les règles de complémentarité. Deux solutions peuvent être proposées.
Formulaire inchangé à l'exception du nom du fichier de traitement en php (traitement4.php)

```
<?php
$sequence=$_POST ['sequence'];                    récupération de la séquence dans la variable $sequence
$sequence_Maj= strtoupper($sequence);             transformation de toutes les lettres en majuscules.
$A_complement =strtr($sequence, A, t);            transformation de tous les A en t dans une nouvelle variable
$T_complement =strtr($A_complement, T, a);        transformation de tous les T en a de la nouvelle variable précédente
$C_complement =strtr($T_complement, C, g);        transformation de tous les C en g de la nouvelle variable précédente
$G_complement =strtr($C_complement, G, c);        transformation de tous les G en c de la nouvelle variable précédente

$complement =strtoupper($G_complement);           rétablissement, dans la variable $complement de la chaîne en majuscules.
echo "La séquence complémentaire est : $complement";
?>
```

Astuce :
Le passage des lettres par les minuscules est indispensable pour éviter que les A transformés en T soient, à l'étape suivante, T transformés en A, modifiés de nouveau !
Exemple de fonctionnement :

```
AATCGTTAG
ttTCGTTtG
ttaCGaatG
ttagGaatG
ttagcaatc
TTAGCAATC
```

Fichier traitement5.php

Une deuxième méthode plus élégante permet de transformer en une seule étape toutes le lettres en utilisant la notion de tableau (array) associatif de paires qui seront remplacées. Les lettres ne sont traitées qu'une fois.

```
<?php
$sequence=$_POST ['sequence'];                    récupération de la séquence dans la variable $sequence
$regles_complementarite = array(A=>T,T=>A,C=>G,G=>C);   tableau indiquant les règles
$complement =strtr($sequence, $regles_complementarite);   application de la transformation
echo "La séquence complémentaire est : $complement";
?>
```

■ EXERCICE 8 : DÉTERMINATION DU POURCENTAGE DE G,C

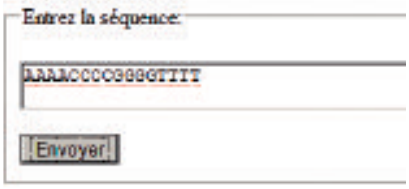
On va utiliser une autre fonction de traitement des chaînes de caractères dans php.
La fonction substr_count permet de déterminer le nombre d'un caractère dans une chaîne.
Un rapport affichera ensuite, avec la taille de la séquence, la fréquence des nucléotides et le pourcentage GC.

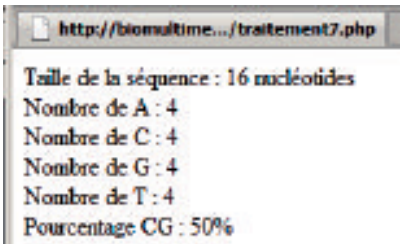
```
function GC_content($seq) {
        $number_of_G=substr_count($seq,"G");
        $number_of_C=substr_count($seq,"C");

$gc_porcentaje=round(100*($number_of_G+$number_of_C)/strlen($seq),2);
        return "G+C %: $gc_porcentaje\n\n";
}
```

```
<?php
$sequence=$_POST ['sequence'];
$taille_sequence=strlen($sequence);
$nombre_A=substr_count($sequence,"A");
$nombre_C=substr_count($sequence,"C");
$nombre_G=substr_count($sequence,"G");
$nombre_T=substr_count($sequence,"T");
$pourcentage_CG=round(100*($nombre_C+$nombre_G)/$taille_sequence,2);

echo "Taille de la séquence : $taille_sequence nucléotides<br />";
echo "Nombre de A : $nombre_A <br />";
echo "Nombre de C : $nombre_C <br />";
echo "Nombre de G : $nombre_G <br />";
echo "Nombre de T : $nombre_T <br />";
echo "Pourcentage CG : $pourcentage_CG%";
?>
```





FORMULAIRES

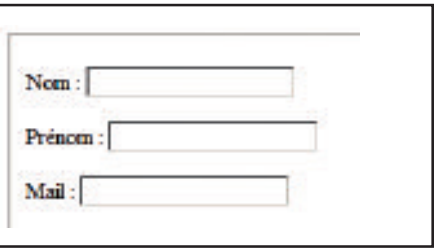
Les formulaires sont omniprésents dans les pages web.
Saisir un mot clé dans un moteur de recherche, s'inscrire dans un groupe de discussion, ou passer une commande en ligne ne sont possibles que grâce aux formulaires. Les informations saisies peuvent être envoyées :
- à une adresse courriel (e-mail),
- à un programme de traitement, écrit par exemple en PHP, suivi ou non d'un stockage dans une base de données.

Les éléments constitutifs d'un formulaire doivent être contenus entre les balises <form> </form> ; cette balise comporte deux attributs importants :
- L'attribut **action** : il indique l'action à exécuter lorsque l'on clique sur le bouton "envoyer". Ce sera en général un script programmé dans un langage donné (php, perl, asp…) ; ce script s'exécute sur le serveur qui héberge la page du formulaire quand on fait clique sur "envoyer".
- L'attribut **method** : il permet de définir la méthode de transfert des données vers le serveur. Les deux valeurs possibles sont **GET** et **POST**.

Voir http://www.j0k3r.net/article/php-formulaires-methode-post-ou-get
Les deux moyens de transfert utilisent des tableaux. Get permet le passage dans l'URL mais est limité à 255 caractères. Post évite la mise en évidence des paramètres.

À l'intérieur des balises "form" on peut insérer des éléments interactifs qui peuvent être :
- une balise input, autofermante (ou qui ne nécessite pas de balise fermante) qui peut contenir un champ de saisie (type = "text"), des boutons radio (type = "radio"), des cases à cocher (type = "checkbox"), un bouton de soumission (type = "submit") ;
- une balise textarea : une zone de saisie multiligne ;
- une balise select : une liste à choix multiples.

```
<form action="adresse de fichier de traitement" method="get">
<fieldset>
<p>Nom : <input name="Nom" type="text" /></p>
<p>Prénom : <input name="prenom" type="text" /></p>
<p>Mail : <input name="mail" type="text" /></p>
<input type="submit" name="Envoyer" />
</fieldset>
</form>
```



FONCTIONS

Le langage php supporte le concept de fonctions.
Une fonction est un petit programme, écrit par l'utilisateur, assurant des opérations et qui rend, au programme qui l'utilise, un résultat. En appelant la fonction logarithme, l'utilisateur lui transmet une valeur numérique (un paramètre) et la fonction renvoie la valeur du logarithme de cette valeur si cela est possible ou un message d'erreur.
Une fonction, dans le cadre de la bioinformatique, peut rendre le DNA complémentaire du DNA transmis (le paramètre). En reprenant les méthodes vues précédemment, la fonction peut s'écrire :

```
function DNAcomplement($seq)
        {
                $regles_complementarite = array(A=>T,T=>A,C=>G,G=>C);
                $complement =strtr($sequence, $regles_complementarite);
                        return $complement;
        }
```

Elle pourrait aussi s'écrire :

```
function DNAcomplement ($z, $option)
        {
                        $dna="";
                        for ($i=0; $i<strlen($z); $i++)
                                {
                                        $a=substr($z,$i,1);
                                        if ($a= ="A") {  if ($option= =1) {$c="T";} else {$c="U";};};
                                        if ($a= ="C") {$c="G";};
                                        if ($a= ="G") {$c="C";};
                                        if ($a= ="T") {$c="A";};
                                        $dna=$dna.$c;
                                };
        return $dna;
        }
```

Le programme principal peut s'écrire :

```
<?php
$sequence=$_POST ['sequence'];                    récupération de la séquence dans la variable $sequence
$sequence_complementaire= DNAcomplement($sequence);   transformation de toutes les lettres en majuscules.
echo "La séquence source est : $sequence <br /> ";
echo "La séquence complémentaire est : $sequence_complementaire";
?>
```

L'intérêt des fonctions est multiple :
- rendre le programme plus lisible,
- simplifier l'écriture,
- rendre possible la division du travail entre plusieurs personnes, chacune assurant l'écriture de certaines fonctions, donc faciliter la maintenance,
- permettre l'enrichissement d'une bibliothèque de fonctions réutilisables dans d'autres applications, et intégrables dans le langage utilisé lui-même.

Il est évidemment nécessaire de bien définir les limites imposées aux fonctions. Dans le cas de la séquence complémentaire décrite ci-dessus, aucune vérification n'est faite sur la chaîne transmise en paramètre : elle peut contenir des lettres ne représentant pas de bases du DNA, notamment des U ! Il est possible d'intégrer une vérification selon le cahier des charges fixé. La comparaison de deux méthodes utilisées dans les deux fonctions montre que plusieurs voies sont possibles pour résoudre un même problème. Le choix de la méthode utilisée peut être lié à la performance, en terme de vitesse. Remarquer que l'ajout d'un paramètre permet d'utiliser cette fonction, sensée rendre un DNA, pour fabriquer un RNA complémentaire de la séquence…

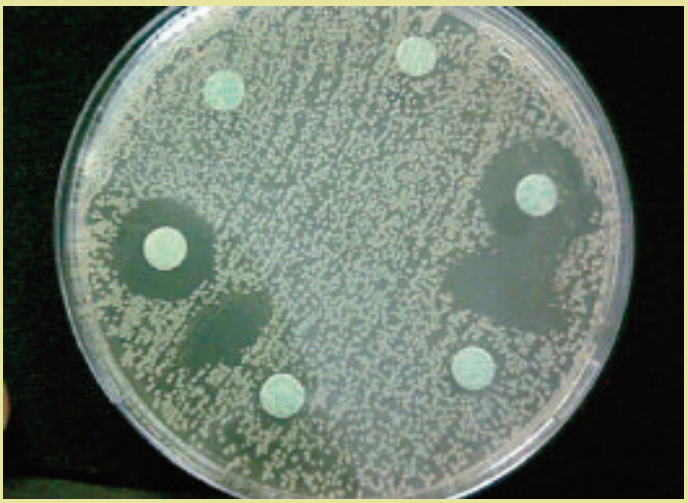
4. GÉRER UNE BASE DE DONNÉES AVEC PHP…

Gérer un site internet, et afficher des pages peut se faire de façon simple avec HTML et ses liens.
Il est toutefois possible de faire autrement : l'ensemble des pages peut s'intégrer comme éléments d'une base de données et être gérés comme tels par php.
Au lieu de faire un lien vers staphylococcus.html, on affiche la page (le fichier) 69 de la base de données.

[http://www.techmicrobio.eu/index.php?option=com_content&view=article&id=69:staphylococcaceae&catid=35:systematique-bacterienne&Itemid=90]

un prochain numero de **L'OPÉRON** developpera cet aspect

L'ymagier



"Mise en évidence d'une bétalactamase à spectre élargi chez *Klebsiella pneumoniae* : synergie entre l'acide clavulanique et une céphalosporine de 3ème génération formant une inhibition en forme de bouchon de champagne. Milieu de Mueller Hinton"
(Photo : Claire MOTTET, Lycée Jean Perrin, Rezé)