

# Initiation à la programmation (PHP)

## Partie 0 - Programme

Nous vous proposerons chaque jour de la lecture, des activités/exercices et des ressources. Tout est décrit dans les parties correspondantes.

Cette première partie décrit ce que nous attendons de vous aujourd'hui.

### 1 - Soyez curieux !

Cf. partie "*Lecture*".

### 2 - Faites les activités dans l'ordre et mettez en pratique

Et n'hésitez à aller plus loin **si et seulement si** vous en avez le temps et que vous avancez bien.

Vous allez acquérir des connaissances au fil de votre avancement dans les activités. Utilisez-les pour résoudre un maximum de problèmes parmi les séries d'exercices que nous vous proposons.

### 3 - Elargissez vos réseaux sociaux et soyez curieux (bis) !

N'oubliez pas de partager vos découvertes avec les autres via Slack !

## Partie 1 - Lecture

### Comprendre en quoi consiste la programmation ! (3 min vidéo)

Courte vidéo d'introduction à un cours en ligne. Toujours utile. Hélas, le reste des vidéos est cher.

<https://www.video2brain.com/fr/tuto/en-quoi-consiste-la-programmation>

## Partie 2 - Activités

### PHP : les bases

PHP est un outil puissant, il est utilisé pour le développement des sites, partout dans le monde. Grâce à Php un site devient dynamique, c'est à dire que les

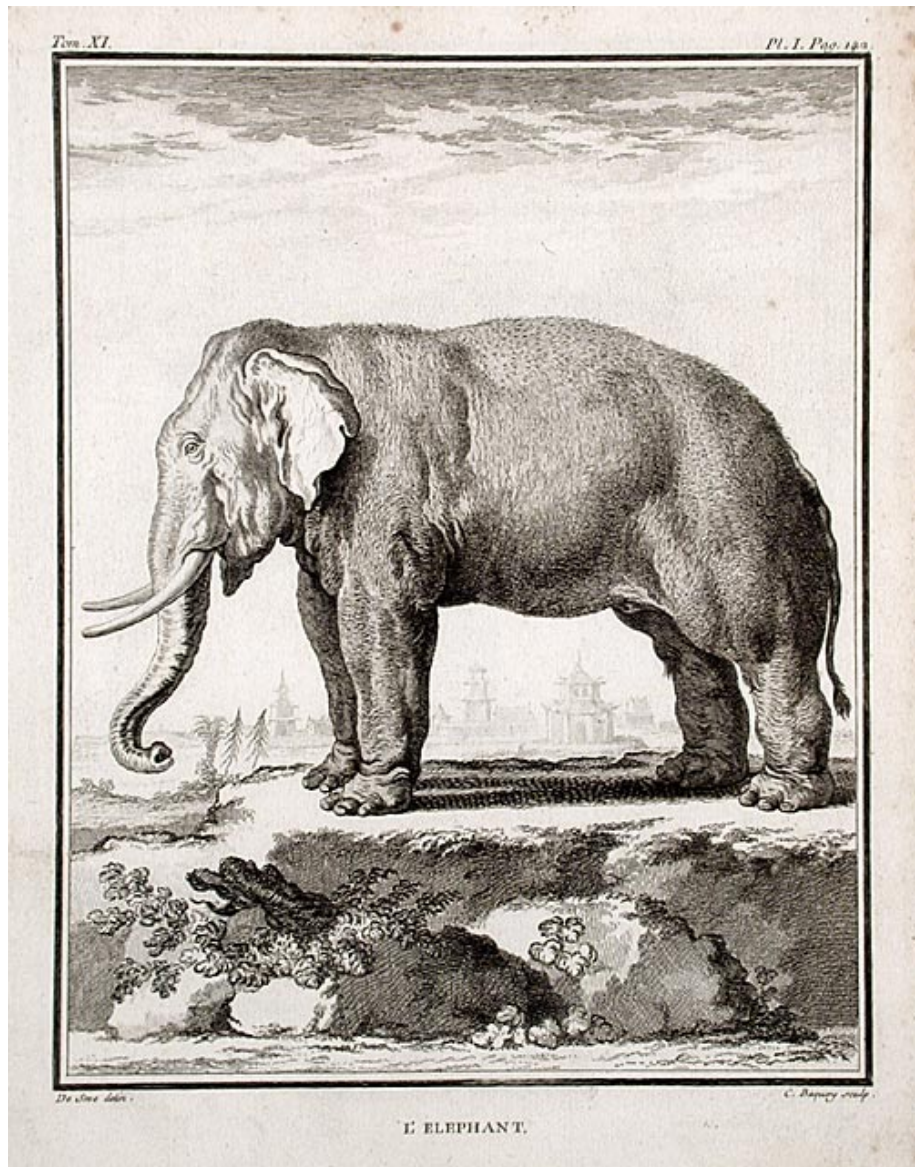


Figure 1: The Elephant

pages peuvent être générées selon l'internaute, la langue, la langue etc.

Après les 2 premières parties du cours : *Partie 1 - Les bases de PHP* et *Partie 2 - Transmettre des données de page en page*, vous réaliserez au minimum 2 exercices dans les catégories “*Introduction PHP*”, “*Les boucles*”, “*Les tableaux*” et “*blocs et tri*”.

Après la *Partie 3 - Stocker des informations dans une base de données*, faites les exercices de la catégorie “*PHP/MySQL*”

(Pour les exercices voir la *Partie 3 - Exercices qui se trouve dans ce document*).

<https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql>

## PHP / MySQL

Le réel intérêt d'un langage de type **back-end** comme PHP est de gérer les données (stocker, restituer, transformer, etc.).

Par exemple, pour un formulaire de contact pour un site internet il serait bien de stocker les informations préalablement rentrées quelque part.

Afin de pouvoir stocker et restituer les données on utilise une Base De Données.

<https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/presentation-des-bases-de-donnees-2>

## Partie 3 - Exercices

### A faire après la partie “*Partie 2 - Transmettre des données de page en page*”

Pour débiter, utilisez une page d'affichage qui récupère les données rentrées par l'utilisateur et une page de traitement de données.

### Catégorie : Introduction PHP

#### Exercice 1 - Calcul du panier

Calculer le prix total d'un panier. Celui-ci comportera 3 produits différents. Pour chaque produit, laissez la possibilité d'entrer le **prix** (hors taxe), le **taux de tva** (en pourcentage), et la **quantité**.

## Exercice 2 - Comparateur de prix

Comparer le prix 4 produits et afficher le nom du produit le moins cher et le nom du produit le plus cher. Donnez la possibilité de rentrer le **nom** des produits et leur **prix**.

## Exercice 3 - Le compte en banque

Vous devez afficher l'état de votre compte en banque. S'il est **négatif**, vous devez affiché "**Déficitaire**", s'il est à 0, vous devez affiché "**En équilibre**" et s'il est **positif** vous devez affiché "**Crédité**". Laissez la possibilité d'entrer l'état du compte.

## Exercice 4 - Nombre de jours dans un mois

Afficher le nombre de jours dans un mois. Les mois sont choisis à partir d'une **select box**. On considère dans cet exercice que le mois de février comprend toujours 28 jours.

## Exercice 5 - Valider une date

Validez la date d'anniversaire rentrée par un utilisateur. une date valide doit être de la forme "jj/mm/yyyy" autrement dit "15/09/2019".

## Exercice 6 - Le stationnement alternatif

Dans une rue où se pratique le stationnement alternatif, du 1 au 15 du mois, on se gare du côté des maisons ayant un numéro **impair**, et le reste du mois, on se gare de l'autre côté. Laissez la possibilité de choisir la **date** et le **numéro de la maison** devant laquelle vous vous êtes garé, puis affichez si vous êtes bien garé ou non.

## Catégorie : Les boucles

### Exercice 1 - Afficher les n premiers chiffres

L'utilisateur entre 2 nombres qui correspondent au **début** et à la **fin** d'une plage de nombres.

Vous devez afficher tous les nombres de cette plage en respectant les règles ci-dessous :

- Les nombres doivent être affichés sous forme de liste

- Les nombres paires sont en bleu
- Les nombres impaires sont en orange
- Les nombres qui sont des carrés parfaits sont en gras

*Note* : La plage de nombres doit être obligatoirement des nombres positifs (supérieur à 0)

## Exercice 2 - Maximum de nombres

Affichez 10 champs de texte. L'utilisateur les remplira avec des nombres de son choix. Vous devez afficher le nombre le plus grand.

## Exercice 3 - Génération de suites

Sur une plage de nombres de 0 à 100, affichez les suites ci-dessous dans des blocs différents :

a) Une suite de nombre avec *“Le pas croissant”*

1, 2, 4, 7, 11, 16, ...

b) Une suite de nombre dite *“La boiteuse”*

1, 2, 4, 5, 7, 8, 10, 11, ...

c) La suite de Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

## Exercice 4 - Numérologie

Convertir votre nom et votre prénom en un nombre sur le principe de la numérologie, chaque lettre a un poids. Exemple :

A = 1, B = 2, C = 3, etc..

Afficher le poids de votre prénom et le poids de votre nom. Par exemple si vous vous appelez Bob, cela donne :

B = 2, O = 15, B = 2 donc le poids est  $2 + 15 + 2 = 19$

## Exercice 5 - Jeu de la fourchette

Simuler le jeu de la fourchette. Ce jeu consiste à essayer de découvrir un nombre quelconque compris entre 1 et 100 inclus, tiré au sort par l'ordinateur (la primitive `rand(min, max)` retourne un entier entre `min` et `max`). L'utilisateur a droit à huit essais maximum. À chaque essai, vous devrez afficher un message indicatif « **nombre donné trop petit** » ou « **nombre donné trop grand** »

». En conclusion, soit « **bravo, vous avez trouvé en *[nombre]* essai(s)** » soit « **désolé, le nombre était *[valeur]*** ».

## Catégorie : Les tableaux

### Exercice 1 - Somme

Écrire un algorithme qui calcule et affiche la **somme** des entiers d'un tableau.

### Exercice 2 - Maximum et minimum

Écrire un un algorithme qui affiche le **plus grand** et le **plus petit** nombre contenu dans un tableau d'entiers.

### Exercice 3 - Nombre d'éléments d'un tableau

Écrire un un algorithme qui affiche le nombre d'éléments qu'il y a dans un tableau.

### Exercice 4 - Plus grand écart

Écrire un algorithme qui calcule le **plus grand écart** entre deux entiers consécutifs dans un tableau.

### Exercice 5 - Tableau ordonné ?

Écrire un un algorithme qui affiche “**vrai**” si un tableau d'entiers est ordonné (strictement) croissant sur les valeurs, ou “**faux**” si ce n'est pas le cas.

### Exercice 6 - Occurence des chiffres

Écrire un un algorithme qui affiche pour chacun des chiffres le nombre de fois qu'il apparait dans un nombre. Ainsi, pour le nombre 10502851125, l'affichage mentionnera que le chiffre 0 apparait 2 fois, 1 apparait 3 fois, 2 apparait 2 fois, 5 apparait 3 fois et 8 apparait une fois (l'affichage ne mentionnera donc pas les chiffres qui n'apparaissent pas).

## “Blocs et tri”

### Exercice 1 - Le chiffrement de César

Depuis l’antiquité, les hommes politiques, les militaires, les hommes d’affaires cherchent à garder secret les messages importants qu’ils doivent envoyer. L’empereur César utilisait une technique (on dit un chiffrement) qui porte à présent son nom : remplacer chaque lettre du message par la lettre qui se situe  $k$  position plus loin dans l’alphabet (cycliquement).

Exemple : si  $k$  vaut 2, alors le texte clair "CESAR" devient "EGUCT" lorsqu’il est chiffré et le texte "ZUT" devient "BWV".

Bien sûr, il faut que l’expéditeur du message et le récepteur se soient mis d’accord sur la valeur de  $k$ .

Vous devez permettre à l’utilisateur de rentrer un texte pour être crypté, de un texte crypté pour être décrypté et de pouvoir choisir la valeur  $k$ .

### Exercice 2 - Validité d’une date

Reprendre l’algorithme de validation d’une date développé dans la partie “*Les structures de contrôle*” et le rendre modulaire. C’est-à-dire que vous aurez plusieurs méthodes dans votre code :

- Une fonction qui vérifie la validité du jour. Elle recevra en paramètre le jour et retournera un booléen (“vrai” si le jour est valide, sinon “faux”) ;
- Une fonction qui vérifie la validité du mois (exprimé en nombre, de 1 à 12). Elle recevra en paramètre le mois et retournera un booléen (“vrai” si le mois est valide, sinon “faux”) ;
- Une fonction qui vérifie la validité d’une année. Elle recevra en paramètre l’année et retournera un booléen (“vrai” si l’année est valide, sinon “faux”) ;
- Une fonction qui utilise les trois précédentes pour vérifier qu’une date est bien valide. Cette fonction recevra en paramètre le jour, le mois et l’année, chacun sous la forme d’un entier.

Vous afficherez “vrai” si la date encodée par l’utilisateur est valide, sinon vous afficherez “faux”.

Attention ! On ne tiendra pas compte des années bissextiles dans cet exercice.

### Exercice 3 - Le stationnement alternatif avec plaque d’immatriculation

Reprenez l’exercice “*Le stationnement alternatif*”. Maintenant, en plus des contraintes précédentes, pour pouvoir se garer une voiture doit avoir un numéro pair du 1 au 15 et un numéro impair le reste du mois. Ajouter un champ pour

entrer le numéro de plaque d'immatriculation. Nous prendrons en compte que les nouvelles plaques d'immatriculation. ex: AA 555 ZZ

## Catégorie : PHP/MySQL

Créez une base de données **concessionnaire** à l'aide de phpmyadmin.

### Exercice 1 - Identité

Dans la base de données de **concessionnaire**, créez la table **utilisateur**. Cette table aura comme champs : **id**, **identifiant**, **mot\_de\_passe**.

Ajoutez des utilisateurs dans la table en passant par l'interface de **phpmyadmin**.

Créez un formulaire pour permettre à l'utilisateur de rentrer son **identifiant** et son **mot de passe**.

Créez une autre page qui vérifie si le couple **identifiant** et **mot de passe**, rentré par l'utilisateur, est valide.

Afficher le message **“Bienvenue”** si l'utilisateur est valide et **“Mot de passe ou identifiant incorrectes”** dans le cas contraire.

### Exercice 2 - Le concessionnaire

Dans la base de données **concessionnaire**, créer une table nommée **véhicule**. Cette table aura comme champs : **id**, **nom**, **couleur**, **marque**, **consommation**, **puissance**, **prix**.

Nous allons interagir avec ces bases de données.

Tout d'abord, créez formulaire pour pouvoir ajouter un véhicule dans la base de données. Créez une page qui affiche, dans un tableau toutes les informations des véhicules.

Mettre en place la fonctionnalité pour permettre la suppression d'un véhicule.

### *[Pour les courageux]* Challenge - Réorganisation de répertoire

Pour ce dernier exercice, on vous donne peu de consignes, peu de pistes, c'est à vous de vous débrouiller pour trouver une solution qui fonctionne et construire un code propre, lisible, modulaire et réutilisable.

Écrire un programme qui réorganise les éléments d'un répertoire de votre machine. Il devra ranger les fichiers par extensions dans un même dossier. Par exemple tous les fichiers **.mp3** seront rangés dans le dossier **“mp3”**, tous les fichiers **.odt** seront rangés dans le dossier **“odt”**



## Ressources

Documentation de PHP : <http://php.net>

Pdo Cheat Cheets <http://www.mustbebuilt.co.uk/2012/10/16/pdo-cheatsheet/>

### La portée d'une variable :

Les variables ont différentes portées. Chaque variable existe dans un contexte.

<http://php.net/manual/fr/language.variables.scope.php>

Afin de pouvoir faire la relation entre des *tables* il faut utiliser le système de *clé primaire* et de *clé étrangère* :

<https://openclassrooms.com/courses/administrez-vos-bases-de-donnees-avec-mysql/cles-primaires-et-etrangeres>