

# Développement d'un site web dynamique (UAA12)

Exercices : échange de données (GET/POST)

## Objets d'apprentissage :

- ✓ Elaborer un formulaire sur la base d'une structure donnée.
- ✓ Construire une page WEB dynamique à l'aide du langage PHP.
- ✓ Construire une page WEB dynamique à l'aide du langage Javascript.
- ✓ Concevoir un formulaire répondant aux exigences d'un cahier des charges.
- ✓ Dynamiser un site WEB exclusivement à l'aide du langage Javascript.
- ✓ Associer des balises HTML de formulaires à leur sémantique.

## Exercice 1 : GET et révision du JavaScript

L'objectif de cet exercice est de créer un site WEB éducatif permettant de tester ses connaissances en répondant à des quizz comportant plusieurs questions à choix multiples. Pour le rendre attractif, ce site présentera les réponses sous forme de boutons qui se déplacent sur l'écran.

Du côté JavaScript, tu te chargeras de programmer les « boutons volants ». La réponse sélectionnée par l'utilisateur sera ensuite envoyée au serveur PHP en utilisant la méthode GET.

Du côté PHP, tu te chargeras de vérifier si la réponse est correcte ou pas.

### Étape 1

Du côté HTML, crée une page comportant :

- un paragraphe
- suivi d'un div vide de 600px de large sur 400px de haut
- lui-même suivi d'un paragraphe indiquant « Cliquez sur la réponse correcte ! »

Prévois un identifiant pour le premier paragraphe et pour le div. Utilise les règles CSS pour donner une bordure au div, pour préciser ses dimensions et pour le mettre en position *relative* (nécessaire pour déplacer les boutons à l'intérieur).

### Étape 2

Ajoute un script JavaScript qui se déclenchera une fois la page chargée.

### Étape 3

Ajoute au script JavaScript les instructions nécessaires pour insérer quatre boutons au div.

Voici les étapes à suivre pour chaque bouton :

1. crée l'élément `<button>` via `document.createElement1`
2. ajoute-lui la classe CSS `boutonVolant` via la propriété `className2`
3. ajoute-lui un identifiant via la propriété `id3`
4. ajoute l'élément dans le div via `appendChild`

Modifie la définition de la classe CSS `boutonVolant` pour qu'elle impose une largeur minimale de 60px et une hauteur minimale de 30px. Fais également en sorte que les boutons aient une position absolue (**position: absolute**).

Observe le résultat : les boutons doivent être tous superposés dans le coin supérieur gauche du div.

---

1 [https://www.w3schools.com/jsref/met\\_document\\_createelement.asp](https://www.w3schools.com/jsref/met_document_createelement.asp)

2 [https://www.w3schools.com/jsref/prop\\_html\\_classname.asp](https://www.w3schools.com/jsref/prop_html_classname.asp)

3 [https://www.w3schools.com/jsref/prop\\_html\\_id.asp](https://www.w3schools.com/jsref/prop_html_id.asp)

## Étape 4

Occupe-toi maintenant du positionnement des boutons.

La position d'un bouton est définie par deux coordonnées (`posX`, `posY`) et son déplacement est déterminé par deux vitesses (`vitesseX`, `vitesseY`).

Les valeurs des coordonnées et de la vitesse seront générées aléatoirement lors de leur création :

- La position en x du coin supérieur gauche du bouton sera un entier déterminé aléatoirement et compris entre 0 et `600 - clientWidth` (étant donné que le div fait 600px de large, il faut soustraire la largeur du bouton pour éviter qu'il ne déborde en dehors du div)
- La position en y du coin supérieur gauche du bouton sera un entier déterminé aléatoirement et compris entre 0 et `400 - clientHeight` (étant donné que le div fait 400px de hauteur, il faut soustraire la hauteur du bouton pour éviter qu'il ne déborde en dehors du div)
- Les vitesses en x et y seront un entier déterminé aléatoirement et compris entre -5 et 5

Voici l'instruction permettant de générer un nombre aléatoire entre deux entiers min et max (inclus) :

```
Math.floor(Math.random() * (max - min + 1)) + min;
```

Les coordonnées et la vitesse de chaque bouton seront enregistrées dans un tableau à deux dimensions. La clé est l'identifiant du bouton et les valeurs sont ses coordonnées et sa vitesse.

Voici un exemple du contenu de ce tableau 2D :

	pos X	pos Y	vitesse X	vitesse Y
1	30	50	2	-1
2	20	10	1	1
3	50	50	-5	2

Voici quelques instructions utiles pour mettre en place ton tableau 2D :

```
// crée un tableau vide
let tabPos = [];
// crée un tableau vide pour la clé "1" (ce qui donne un tableau 2D)
tabPos[1] = [];
// ajoute la valeur 30 pour la clé "posX" dans le tableau vide de la clé "1"
tabPos[1]["posX"] = 30;
// ajoute la valeur 50 pour la clé "posY" dans le tableau vide de la clé "1"
tabPos[1]["posY"] = 50;
```

Pour positionner précisément un bouton à l'intérieur d'un div, il faut indiquer la position via les propriétés `style.left` et `style.top` :

```
bouton.style.top = <valeur> + "px"
bouton.style.left = <valeur> + "px"
```

Attention : <valeur> est à récupérer directement depuis le tableaux des positions !

## Étape 5

Occupe-toi maintenant du déplacement des boutons.

Le mouvement est simulé ainsi : si, à un moment, le bouton est positionné en (1,4) et a une vitesse de (2,-1), le bouton se déplacera de 2px vers la droite et de 1px vers le haut ; il passera alors aux positions (3,3), (5,2), (7,1), etc.



Crée la fonction `déplacerBoutons()` qui se charge de déplacer tous les boutons présents sur l'interface :

1. Cible l'ensemble des boutons grâce à `document.getElementsByClassName(classe)`. Attention, puisque cette méthode renvoie un tableau d'éléments, tu dois parcourir l'ensemble du tableau (à l'aide d'un **for**)
2. Pour chaque bouton :
  - a) extrait sa position (`posX, posY`) et sa vitesse (`vitesseX, vitesseY`) depuis le tableau 2D grâce à son `id`
  - b) calcule sa nouvelle position selon sa vitesse
  - c) effectue le mécanisme de rebond (dans le cas où le bouton sortirait du div)
  - d) enregistre sa nouvelle position dans le tableau des positions (`tabPos`)

### Algorithme du mécanisme de rebond

```
*
nouvellePosX = posX + vitesseX
nouvellePosY = posY + vitesseY
// Cas où le bouton sort à gauche ou à droite
if (nouvellePosX < 0 OR nouvellePosX + largeur > 600)
    vitesseX = vitesseX * -1
    nouvellePosX = posX + vitesseX

// Cas où le bouton sort en haut ou en bas
if (nouvellePosY < 0 OR nouvellePosY + hauteur > 400)
    vitesseY = vitesseY * -1
    nouvellePosY = posY + vitesseY
```

## Etape 6

Fais en sorte que la fonction `déplacerBoutons()` soit exécutée toutes les 30 millisecondes. Utilise pour cela la fonction `setInterval(...)`<sup>4</sup>.

## Etape 7

Il est temps maintenant de s'occuper du premier paragraphe (celui qui affiche la question).

Génère tout d'abord deux entiers (`nombre1` et `nombre2`) entre 0 et 10 et affiche le texte « Que vaut `<nombre1> x <nombre2> ?` »

## Etape 8

Fais en sorte que chacun des boutons comporte une réponse différente, avec le texte « `<nombre1> x <nombre2> = <reponse>` » avec :

Bouton 1 : `reponse = nombre1 * nombre2` (c'est-à-dire la bonne réponse)

Bouton 2 : `reponse = nombre1 + nombre2`

Bouton 3 : `reponse = (nombre1 * nombre2) + 2`

Bouton 4 : `reponse = (nombre1 * nombre2) + 6`

Indice : utilise la propriété `innerHTML` des boutons.

Fais en sorte que, lorsqu'on clique sur chacun de ces boutons, l'utilisateur soit redirigé vers la page `verification.php` en passant via la méthode GET les arguments suivants :

- `n1 = nombre1`
- `n2 = nombre2`
- `rep = reponse`

Indice : utilise la propriété `onclick` des boutons<sup>5</sup>.

## Etape 9

Crée un fichier `verification.php` qui :

- affiche « Vous ne pouvez pas accéder à cette page » si aucune donnée n'est transmise en GET
- affiche « Réponse correcte ! » si l'utilisateur a choisi la bonne réponse
- affiche « Réponse incorrecte ! » sinon

## Etape 10 (dépassement)

Ajoute des boutons qui se déplacent aléatoirement dans l'interface (c'est à dire qui ont une vitesse qui change à chaque déplacement).

---

<sup>4</sup> [https://www.w3schools.com/jsref/met\\_win\\_setinterval.asp](https://www.w3schools.com/jsref/met_win_setinterval.asp)

<sup>5</sup> <https://stackoverflow.com/questions/15097315/change-onclick-attribute-with-javascript>

## Exercice 2 : POST

L'objectif de cet exercice est de réaliser un formulaire permettant à l'utilisateur d'entrer son nom et sa date de naissance. Ces données sont envoyées au serveur et celui-ci produira une page listant ses futurs anniversaires et affichant un certain nombre d'informations choisies par l'utilisateur.

### Etape 1

Crée un formulaire HTML comportant trois sections (balises `<fieldset>`) :

1. La première, nommée « Identification » (balise `<legend>`) contient simplement un champ textuel (balise `<input type="text">`) demandant le prénom de l'utilisateur
2. La seconde, nommée « Date de naissance », contient :
  - i. une liste déroulante (balise `<select>`) proposant à l'utilisateur de sélectionner son jour de naissance (de 1 à 31)
  - ii. des boutons radio (balise `<input type="radio">`) proposant à l'utilisateur de sélectionner son mois de naissance
  - iii. un champ numérique (balise `<input type="number">`) proposant à l'utilisateur d'entrer son année de naissance
3. La troisième, nommée « Colonnes à afficher », contient trois cases à cocher (balise `<input type="checkbox">`) proposant à l'utilisateur de choisir s'il souhaite afficher les dates et/ou les jours et/ou l'âge

Ce formulaire comporte également un bouton de soumission (balise `<input type="submit">`) permettant d'envoyer les données du formulaire au serveur lorsque l'utilisateur clique dessus.

Quelques points d'attention :

- ✓ N'oublie pas d'ajouter des étiquettes aux champs (balise `<label>`)
- ✓ La date par défaut est le 1<sup>er</sup> janvier 2000
- ✓ Choisis la méthode d'envoi POST pour le formulaire ; fais en sorte que la balise `<form>` comporte les attributs suivants : `<form action="/data.php" method="post">`

### Etape 2

Crée un fichier nommé `data.php` qui se charge simplement d'afficher les données reçues en POST. Pour cela, tu peux ajouter l'instruction suivante dans ton script :

```
<?php  
var_dump($_POST);  
?>
```

Remplis les données de ton choix dans le formulaire, clique sur le bouton, et observe le résultat. Comprends-tu l'importance de spécifier l'attribut `name` d'un champ ?

### Etape 3

Supprime l'instruction précédente. Tu vas maintenant t'attaquer à l'affichage des futurs anniversaires.

Tout d'abord, fais en sorte que la page affiche « Chère / Cher <prénom>, voici vos futurs anniversaires. »

Ensuite, propose un tableau contenant plusieurs colonnes :

1. La première contient la liste des années allant de 2023 à 2050
2. Si l'utilisateur a coché l'option « date d'anniversaire », une colonne reprend chaque date des futurs anniversaires au format JJ/MM/AAAA
3. Si l'utilisateur a coché l'option « jours d'anniversaire », une colonne reprend chaque jour des futurs anniversaires (lundi, mardi, etc.)
4. Si l'utilisateur a coché l'option « âge », une colonne indique l'âge qu'il atteindra lors de chacun de ses futurs anniversaires.

Pour le calcul du jour (colonne 3), utilise la fonction `date (...)` :

<http://php.net/manual/en/function.date.php>.

### Etape 4

Le prénom est champ obligatoire et l'année de naissance doit être comprise entre 1900 et 2023 : ajoute les contraintes nécessaire en HTML pour faire en sorte que l'utilisateur ne puisse pas soumettre le formulaire si les données sont manquantes ou incorrectes. Utilise pour cela les attributs `required`, `min` et `max`.

Teste le mécanisme de vérification en tentant de soumettre le formulaire sans indiquer de prénom ou en indiquant une année de naissance invalide.

### Etape 5

Ces attributs permettant de vérifier les données d'un formulaire sont-ils sécurisés ? Pour le savoir, rends-toi dans l'inspecteur web de ton navigateur, édite l'HTML en supprimant ces attributs et tente une nouvelle fois d'envoyer le formulaire...

### Etape 6

Les attributs `required`, `min` et `max` sont des mécanismes de vérification basée sur le JavaScript : elles sont effectuées coté client. Dès lors, l'utilisateur peut à sa guise modifier le code pour outrepasser ces mécanismes.

Pour rendre ton projet plus sécurisé, il faut effectuer également la vérification du coté du serveur, c'est à dire en PHP !

À l'aide de la fonction `isset (...)`, fais en sorte que le page PHP n'affiche rien si certaines informations transmises ne sont pas correctes.

### Etape 7 (dépassement)

Fais en sorte d'afficher le message d'erreur « Les erreurs suivantes doivent être corrigées : » suivi des messages suivants (en fonction des cas) dans une liste (balises `<ul>` et `<li>`):

- lorsque le prénom est manquant : « Le prénom est obligatoire. »
- lorsque l'année de naissance est trop grande : « L'année de naissance doit être avant 2023. »
- lorsque l'année de naissance est trop petite : « L'année de naissance doit être après 1900. »



Affiche la / les erreur(s) en gras dans un cadre rouge.

## Références

Les présents exercices ont été élaborés à l'aide des ressources suivantes :

- Forbes, A. (2013). The Joy of PHP: A Beginner's Guide to Programming Interactive Web Sites.
- Vaswani, V. (2021). PHP A Beginner's Guide.
- Cours de « Développement d'applications WEB » (2018), Hénallux