Json,Webstorage, API et indexedDB

Exercice 1 JSON en Javascript

1- Tester sur http://www.jsoneditoronline.org/ le texte suivant

```
{"nom":"IUT Annecy","adresse":"9 rue Arc-en-ciel","ville":"Annecy-le-Vieux","cp":74940}
```

Le chaine est au format JSON

Effectuer quelques essais avec l'interface http://www.jsoneditoronline.org

2- Récupérer le fichier www/json-js

Il y a une différence entre un texte JSON et les objets javascript

`JSON.parse()` : permet de transformer un texte JSON en objet javascript

`JSON.stringify()` : permet de transformer un objet javascript en texte JSON

3- En s'inspirant du fichier fourni (json-js), modifier le code pour :

Ajouter une instruction sur l'oject obj(sans modifier la variable text qui contient du json afin que le code postal de l'IUT soit mis à jour avec la valeur 74942

Attribution de la valeur 74942 à l'objet "cp"

```
obj.cp = 74942;
```

Ajouter un attribut "bp" à l'objet

```
obj.bp = 240;
```

Puis je le print en rajoutant à la commande print `" "+obj.bp;`

Cela donne :

```
obj.cp +" "+obj.ville+" "+obj.bp;
```

À partir d'un objet faire un format JSON on utilise la commande `JSON.stringify`

```
var texte2=JSON.stringify(obj);

alert(texte2); permet de l'afficher dans la boite alerte

document.getElementById("balisep").innerHTML = texte2 affiche dans une balise <p>
```

Ajout de la balise P au début du document html

```
<p id="balisep"></p>
```

4- Gérer les erreurs JSON si le JSON est mal formaté

Pour gérer les erreurs si le json est mal formaté
Code mal formaté:

```
var text2 = '{"nom":"IUT Annecy",adresse":"9 rue Arc-en-ciel","ville":"Annecy-le-Vieux","cp":74940}';
```

Le code permet de parser le texte et en cas d'échec de le print

```
try {
 obj=JSON.parse(text2);
} catch (e) {
 alert("pb parsing. Voir l'erreur dans la console si possible");
 console.error("Parsing error dans mon code :-( :", e);
}
```

5- Object javascript vers JSON

Tous d'abord il faut analyser le tableau ci-dessous

```
var employes = [
{"prenom":"Axel", "nom":"AIRLERYTHME"},
{"prenom":"James", "nom":"LEJAVASCRIPT"},
{"prenom":"Sam","nom": "PLAILAPROGUE"}
];
```

Si on souhaite ajouter une valeur au tableau

```
employes[5]={"prenom":"Sam","nom": "EPATE"};
```

Pour transformer le tableau en JSON dans une nouvelle variable

```
var employes = [
{"prenom":"Axel", "nom":"AIRLERYTHME"},
{"prenom":"James", "nom":"LEJAVASCRIPT"},
{"prenom":"Sam","nom": "PLAILAPROGUE"}
];
employes[5]={"prenom":"Sam","nom": "EPATE"};
var texte3=JSON.stringify(employes);
alert(texte3);
```

On utilise comme explique ci-dessus la fonction `JSON.stringify(valeur)`

On observe deux valeur `NULL` cela est du que on rajoute la valeur Sam Epate à la place numéro 5 mais il n'y a pas de valeur 4

Ensuite nous allons créer un objet

```
var myObject = {
item1: 'Texte 1',
item2: 'Texte 2'
};
```

Je transforme donc cette chaine au format JSON et je l'alerte

```
var texte4=JSON.stringify(myObject);
alert(texte4);
```

Cela correspond bien au résultat attendu

Exercice 2 JSON en php

On peut facilement passer du format JSON au format tableau ou objet php par 2 métohdes
`json_decode()` permet de transformer un texte JSON utf8 en objet ou tableau document VOIR DOC ICI.
Et retourne `NULL` si problème
`json_encode()` permet de retourner du JSON par rapport à des valeurs PHP
Et retourne `false` si problème
On peut affiner la gestion des erreurs avec les fonctions :
`json_last_error` retourne la dernière erreur JSON
`json_last_error_msg` retourne le message d'erreur lors de l'appel des fonctions

- `json_encode()`
- `json_decode()`

1- Analyse du ficher json-php.php

Avec la commande dump, sans option :

```
object(stdClass)[1]
  public 'nom' => string 'IUT Annecy' (length=10)
  public 'adresse' => string '9 rue Arc-en-ciel' (length=17)
  public 'ville' => string 'Annecy-le-Vieux' (length=15)
  public 'cp' => int 74940
```

Idem avec la commande print récursif (print_r), sans option :

```
stdClass Object
(
 [nom] => IUT Annecy
 [adresse] => 9 rue Arc-en-ciel
 [ville] => Annecy-le-Vieux
 [cp] => 74940
)
```

Avec la commande dump, AVEC option true (array) :

```
array (size=4)
  'nom' => string 'IUT Annecy' (length=10)
  'adresse' => string '9 rue Arc-en-ciel' (length=17)
  'ville' => string 'Annecy-le-Vieux' (length=15)
  'cp' => int 74940
```

Idem avec la commande print récursif (print_r), AVEC option true :

```
Array
(
  [nom] => IUT Annecy
  [adresse] => 9 rue Arc-en-ciel
  [ville] => Annecy-le-Vieux
  [cp] => 74940
)
```

2- Gérer les erreurs mal formaté

On va créer une variable mal formaté
`$text2 = '{"nom":"IUT Annecy",adresse":"9 rue Arc-en-ciel","ville":"Annecy-le-`

`Vieux","cp":74940}';`
On peut tester l'affichage des erreurs avec `print_r` et `var_dump`

```
print("<hr/><br/> Avec erreur, commande dump");
var_dump(json_decode($text2, true));
print("<br/> Idem avec la commande print récursif (print_r) ");
print("<pre>");
print_r(json_decode($text2,true));
print("</pre>");
```

On s'appercoit que l'affichage d'erreur ce fait que avec la commande `dump` et affiche `NULL`

3- Analyser la partie encodage

a.
Dans le tableau

```
$arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);
$montext= json_encode($arr);
echo $montext;
print("<br/>");
```

Cela affiche: `{"a":1,"b":2,"c":3,"d":4,"e":5}`
Pour ajouter la valeur f=> 6
`$arr += [ "f" => 6 ];`
résulat : `{"a":1,"b":2,"c":3,"d":4,"e":5,"f":6}`
b.
Dans le code ci-dessous

```
class Personne {
 public $name = "";
 public $hobbies  = "";
 public $birthdate = "";
}
$p = new Personne();
$p->name = "sachin";
$p->hobbies  = "sports";
$p->birthdate = "1-1-2000";
$textp2=json_encode($p);
echo $textp2;
```

Pour ajouter l'attribue `firstname`
j'ajoute les lignes :
`public $firstname = "";`
`$p->firstname ="Ronan";`

Exercice 3 Création d'une API

1- Analyser du code

2- Modifier le code

Paramètre GET (format)	Format de réponse
complet	{"jour":"monday","mois":"january","année":2022,"heure":10,"minutes":25,"secondes":30
date	{"jour":"monday","mois":"january","année":2022}
heure	{"heure":10,"minutes":25,"secondes":30}
Non prévu	{"erreur":"option non connue"}

Solution numéro 1

```
<?php
if (isset($_GET['complet'])){
	$jour=getdate();
	print("{"jour:"".$jour["weekday"]."","mois":".$jour["month"]."","année:"".$jour[
"year"]."","heure:"".$jour["hours"]."","minutes:"".$jour["minutes"]."","secondes:"".$jou
r["seconds"]."}");
}elseif (isset($_GET['date'])){
	$jour=getdate();
	print("le jour est ".$jour["weekday"]);
	print("{"jour:"".$jour["weekday"]."","mois":".$jour["month"]."","année:"".$jour[
"year"]."}");
}elseif (isset($_GET['heure'])){
	$jour=getdate();
	print("{"heure:"".$jour["hours"]."","minutes:"".$jour["minutes"]."","secondes:""
.$jour["seconds"]."}");
}else{
	print("{"erreur" :"opion non connue"}");
}
?>
```

Solution numéro 2

```
<?php
$jour=getdate();
$format = $_GET['format'];
switch($format){
    case $format == 'complet':
        $complet = array('jour' => $jour["weekday"], 'mois' => $jour["month"],
'annee' => $jour["year"], 'heure' => $jour["hours"], 'minute' => $jour["minutes"],
'seconde' => $jour["seconds"]);
        $complet_json= json_encode($complet);
        echo $complet_json;
        break;
    case $format == 'date':
        $date = array('jour' => $jour["weekday"], 'mois' => $jour["month"], 'annee'
=> $jour["year"]);
        $date_json= json_encode($date);
        echo $date_json;
        break;
    case $format == 'heure':
        $heure = array('heure' => $jour["hours"], 'minute' => $jour["minutes"],
'seconde' => $jour["seconds"]);
        $heure_json= json_encode($heure);
```

```
        echo $heure_json;
        break;
    default:
        $nonprevu = array('erreur' => 'option non connue');
        $nonprevus_json= json_encode($nonprevu);
        echo $nonprevus_json;
        break;
}
?>
```

Exercice 4 webstorage / indexedDB (base de données No sql)

Sur la page Webstorage.php

1-

2-Je crée un formulaire demandant un nom et lorsque j'appuie sur le boutton valider cela l'ajoute dans le local storage

```
<!DOCTYPE html>
<html>
       <head>
              <meta charset="utf-8">
       </head>
       <body>
        <label>Prénom</label>
              <input type="text" placeholder="Entrez le prénom" id="prenom">
              <button type="button" onclick="getPrenom();">Valider</button>

              <script>
                     function getPrenom() {
                 var input = document.getElementById("prenom").value;
                 alert(input); //affiche la valeur récupérer

                  // Enregistre la valeur de l'input à la clé prenom
                  localStorage.setItem('prenom', input);

                 let data = localStorage.getItem('prenom');
                 alert("Voici la valeur pour la clé prénom "+data);
              }

        </script>
       </body>
</html>
```

3-Je crée un fichier echowebstorage.php qui affiche la valeur de la clé prenom en alerte et dans la console

```
<!DOCTYPE html>
<html>
       <head>
              <meta charset="utf-8">
       </head>
       <body>
```

```
        <p id="balisep"></p>

             <script>
                let data = localStorage.getItem('prenom');

                document.getElementById("balisep").innerHTML = data;
        </script>
       </body>
</html>
```

4- Créer une autre page echostorage.html qui affiche le prénom sur changement (event) du localStorage.

```
<!DOCTYPE html>
<html>
       <head>
              <meta charset="utf-8">
       </head>
       <body>
        <p id="balisep"></p>

             <script>
                window.addEventListener('storage', function(event) {
                    if (event.key === 'prenom') {
                    console.log('The value of myItem has changed to: ' +
event.newValue)
                    document.getElementById("balisep").innerHTML = event.newValue
                }
                });

        </script>
       </body>
</html>
```