

## PROJET 2

---

À REMETTRE, AU PLUS TARD, LE JOUR DE L'EXAMEN  
MARDI, LE 19 AVRIL 2016 À 13:00:00

### 1. Énoncé du problème

Le but ultime de ce projet est de créer un calendrier perpétuel pour l'année 1900 jusqu'à l'année 2199.

Ce projet est réparti sur 2 pages Web distinctes :

- La page **selectionne-date.htm** qui sert à sélectionner et à valider une date;
- La page **affiche-calendrier.htm** qui sert à afficher le calendrier en fonction d'une date transmise à la page Web.

Ici, **selectionne-date.htm** doit transmettre la date qui a été sélectionnée par l'utilisateur à **affiche-calendrier.htm**.

### 2. Avant de commencer le projet

Créez un dossier nommé **Projet2** puis copiez les fichiers suivants (les six premiers fichiers sont situés sur le lecteur P).

- **selectionne-date.htm**
- **affiche-calendrier.htm**
- **logoCGG.jpg**
- **styleProjet2.css**
- **libUtilitaires.js**
- **libValidationBaseER.js**
- **date.js** (celui que vous avez fait dans le laboratoire 11)

Sur les pages Web **selectionne-date.htm** et **affiche-calendrier.htm**, remplacez toutes les occurrences de **VotreNom** par votre vrai nom.

### 3-1. Projet 2 – 1<sup>ère</sup> partie – Résumé (lire les pages 2 à 5 avant de commencer cette partie)

Pour cette première partie de projet, on vous demande de programmer la page Web **selectionne-date.htm** dont le rôle est de sélectionner et de valider une date dans l'intervalle du 1<sup>er</sup> janvier 1900 au 31 décembre 2199.

- La date doit être saisie par l'utilisateur à partir de quatre listes déroulantes soit :

*ddlJour* = 1 à 31

*ddlMois* = 1 à 12 (Janvier à Décembre)

*ddlSiecle* = 19 à 21

*ddlAnnee* = 00 à 99 (00 à 99)

Dans un premier temps, lors du chargement initial de la page Web, c'est la date du jour qui se reflète dans les différentes listes déroulantes. Par exemple, voici ce qu'on devrait voir si la date du jour est le 8 avril 2016.



The image shows a horizontal row of UI elements. From left to right: a button labeled 'ACTUALISER', a dropdown menu showing '8', another dropdown menu showing 'Avril', a third dropdown menu showing '20', a fourth dropdown menu showing '16', and a button labeled 'CALENDRIER'.

**Vendredi, le 8 avril 2016**

- Le rôle du bouton **CALENDRIER**, apparaissant à la droite des listes déroulantes est de transmettre la date sélectionnée à la page Web **selectionne-date.htm**.

Le bouton **CALENDRIER** ne doit s'afficher que si la date est valide. (Notez que c'est le cas dans l'exemple ci-dessus, puisque cette date reflète celle du jour courant !)

- Le rôle du bouton **ACTUALISER** apparaissant à la gauche des listes déroulantes est de recharger la page Web avec les valeurs initiales.

### 3-2. Projet 2 – 1<sup>ère</sup> partie – Travail à effectuer

**IMPORTANT** : Vous devez TOUJOURS UTILISER les fonctions qui ont déjà été programmées dans les librairies **libUtilitaires.js** et **date.js** à chaque fois que cela s'avère utile. Ne réinventez pas la roue!!!

1. Création des quatre listes déroulantes et du bouton **CALENDRIER**.

Ces listes déroulantes ont déjà été créées en partie. Vous n'avez qu'à programmer les options de la liste (à l'aide de **document.write**). Lire les commentaires dans le XHTML pour savoir ce que vous devez programmer.

```
<select id="ddlJour" name="ddlJour" class="" onchange="afficheBoutonCalendrierEtDateSelectionnee()">
<select id="ddlMois" name="ddlMois" class="" onchange="afficheBoutonCalendrierEtDateSelectionnee()">
<select id="ddlSiecle" name="ddlSiecle" class="" onchange="afficheBoutonCalendrierEtDateSelectionnee()">
<select id="ddlAnnee" name="ddlAnnee" class="" onchange="afficheBoutonCalendrierEtDateSelectionnee()">
```

2. Programmation de la fonction *initialiseAvecDateCourante()* qui (1) récupère la date du jour (jour, mois et année), puis (2) initialise les quatre listes déroulantes à partir de la date qui a été récupérée. Cette fonction est appelée au chargement de la page Web (*onload*).

```
function initialiseAvecDateCourante()
```

Par exemple, si la date du jour est le 8 avril 2016.



**Note** : Vous allez devoir modifier la programmation de cette fonction plus tard.

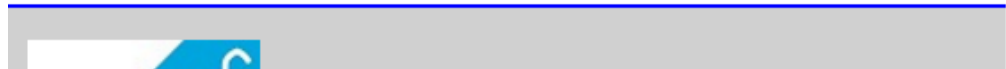
3. À chaque fois que l'utilisateur modifie le jour, le mois, le siècle ou l'année (événement *onchange*), il y a exécution de la fonction *afficheBoutonCalendrierEtDateSelectionnee()* qui (1) récupère le contenu des quatre listes déroulantes, (2) construit la date correspondante et la valide, puis en fonction du résultat de la validation, (3) masque ou affiche le bouton CALENDRIER et masque ou affiche la date en format « littéral long » (la 1<sup>ère</sup> lettre du mois doit être en minuscule).

```
function afficheBoutonCalendrierEtDateSelectionnee()
```

Par exemple, si l'utilisateur sélectionne le 28 février 2001, le bouton CALENDRIER doit s'afficher et la date sélectionnée doit s'afficher en format « littéral long » (car cette date est valide).



**Mercredi, le 28 février 2001**



Par exemple, si l'utilisateur sélectionne le 29 février 2001, le bouton CALENDRIER doit être masqué ainsi que la date en format « littéral long » (car cette date n'est pas valide).




4. À chaque fois que l'utilisateur clique sur le bouton CALENDRIER, cette page Web doit transmettre les données du formulaire de manière automatique à la page Web **affiche-calendrier.htm**. Cela doit être programmé dans la fonction *chargePageAffichageCalendrier()*.

```
function chargePageAffichageCalendrier()
```

5. À chaque fois que l'utilisateur clique sur le bouton ACTUALISER, cette page Web doit se recharger. Cela doit être programmé dans la fonction *actualiseCalendrier()*.

```
function actualiseCalendrier()
```

### 3-3. Contraintes

- Vos identificateurs doivent être significatifs. Il est impératif de respecter la convention de noms. Vos chaînes de caractères doivent être entre apostrophes. Vous pouvez perdre jusqu'à 20 pts si vous ne respectez pas ces contraintes.
- Vous ne devez pas modifier ce qui a déjà été tapé. Vous devez modifier seulement ce qui est demandé. Vous pouvez perdre 20 pts si vous ne respectez pas cette contrainte.
- Par contre, vous pouvez modifier le style dans le fichier **styleProjet2.css** (couleurs, polices de caractères, mise en page etc.). Allez-y mollo!!!
- Vous pouvez également ajouter toutes les fonctions que vous voulez (à la condition que celles-ci n'aient pas déjà été programmées dans **date.js** et dans **libUtilitaires.js**).
- Vous devez TOUJOURS UTILISER les fonctions qui ont déjà été programmées dans **libUtilitaires.js** et **date.js**.
  - Par exemple, pour aller chercher la date du jour, vous devez appeler les fonctions *recupereJourActuel*, *recupereMoisActuel* et *recupereAnneeActuelle*.
  - Par exemple, pour valider la date, vous devez appeler la fonction *dateValide*.
  - Par exemple, pour aller chercher la valeur d'une liste déroulante ou pour la modifier, vous devez utiliser la fonction *b*.
  - Par exemple, pour affecter un sélecteur à une balise, vous devez utiliser la fonction *s*.

Vous pouvez perdre jusqu'à 30 pts si vous ne respectez pas cette contrainte.

#### 4-1. Projet 2 – 2<sup>ième</sup> partie – Résumé (lire les pages 6 à 11 avant de commencer cette partie)

Pour cette deuxième partie de projet, on vous demande de programmer la page Web **affiche-calendrier.htm** dont le rôle est d’afficher le calendrier du mois à partir de la date qui a été transmise (par la page Web **selectionne-date.htm**).

Par exemple, voici le calendrier du mois d’avril 2016 si la date transmise est le 8 avril 2016.

Le bouton **NOUVELLE DATE** sert à sélectionner une nouvelle date.

Les cinq (5) autres boutons ont été prévus pour modifier le calendrier affiché.

**Avril 2016**

Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

**IMPORTANT** : Vous devez TOUJOURS UTILISER les fonctions qui ont déjà été programmées dans les librairies **libUtilitaires.js** et **date.js** à chaque fois que cela s’avère utile. Ne réinventez pas la roue!!!

## 4-2. Projet 2 – 2<sup>ième</sup> partie - Travail à effectuer

1. Programmation de la fonction *recupereDateEtConstruitCalendrier()*.

```
function recupereDateEtConstruitCalendrier()
```

Cette fonction est appelée au chargement de la page Web (*onload*).

Cette fonction doit :

- 1- Récupérer la date qui a été transmise à la page Web.
- 2- Si aucune date n'a été transmise, cette fonction doit charger la page Web **selectionne-date.htm**.
- 3- Afficher le mois (en lettres) (dans *lblMois*) ainsi que l'année (dans *lblAnnee*) sur la page Web (par exemple : **Janvier 1900**). La 1<sup>ère</sup> lettre du mois doit être en majuscule.
- 4- Masquer le bouton << ANNÉE PRÉCÉDENTE s'il s'agit du calendrier de l'année **1900**.
- 5- Masquer le bouton ANNÉE SUIVANTE >> s'il s'agit du calendrier de l'année **2199**.
- 6- Masquer le bouton << MOIS PRÉCÉDENT s'il s'agit du calendrier du mois de **Janvier 1900**.
- 7- Masquer le bouton MOIS SUIVANT >> s'il s'agit du calendrier du mois de **Décembre 2199**.
- 8- Construire le calendrier du mois et l'afficher (dans *lblCalendrierCourant*) sur la page Web.

NOUVELLE DATE		MOIS SUIVANT >>					
AUJOURD'HUI		ANNÉE SUIVANTE >>					
Janvier 1900							
Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	
0	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	31	1	2	3	

<< MOIS PRÉCÉDENT		NOUVELLE DATE		MOIS SUIVANT >>			
<< ANNÉE PRÉCÉDENTE		AUJOURD'HUI					
Novembre 2199							
Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	

Le calendrier du mois doit être construit dynamiquement dans un flux de données (utilisez les balises `<table>`/`</table>`, `<tr>`/`</tr>` et `<td>`/`</td>`).

Pour chacune des cellules du calendrier, le sélecteur de mise en forme doit être `sCellDate`.

En plus, pour chacune des cellules de la 1<sup>ère</sup> ligne du calendrier, le sélecteur de mise en forme doit être `sCellEnteteColonne`.

En plus, pour la cellule qui contient le jour courant, le sélecteur de mise en forme doit être `sCellJourCourant`.

En plus, pour la cellule qui contient un jour d'un autre mois, le sélecteur de mise en forme doit être `sCellAutreJour`.

**Juillet 2016**

Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

**Attention** : Dans votre tableau XHTML, vous ne devez pas afficher de lignes vides inutiles.

2. Programmation de la fonction `chargePageSelectionneDate()`.

```
function chargePageSelectionneDate()
```

Cette fonction est appelée lorsque l'utilisateur clique sur le bouton NOUVELLE DATE.

Cette fonction doit recharger la page Web **selectionne-date.htm**.



### 3. Programmation de la fonction *nouveauCalendrier(binMois, intSens)*.

```
function nouveauCalendrier(binMois, intSens)
```

Cette fonction est appelée lorsque l'utilisateur clique sur le bouton << **MOIS PRÉCÉDENT** ou sur le bouton **MOIS SUIVANT** >> ou sur le bouton << **ANNÉE PRÉCÉDENTE** ou sur le bouton **ANNÉE SUIVANTE** >> ou sur le bouton **AUJOURD'HUI**.

Lorsque l'utilisateur clique sur le bouton << **ANNÉE PRÉCÉDENTE**, *binMois* vaut *false* et *intSens* vaut -1 et lorsqu'il clique sur le bouton **ANNÉE SUIVANTE** >>, *binMois* vaut *false* et *intSens* vaut 1.

Cette fonction doit charger la même page Web mais en lui **transmettant manuellement** la date de l'année précédente ou de l'année suivante selon le cas.

Par exemple, si la date qui a été transmise est le 7 mars 2015, alors cette fonction doit charger la même page Web mais en lui transmettant la date du 7 mars 2014 s'il clique sur le bouton << **ANNÉE PRÉCÉDENTE** ou en lui transmettant la date du 7 mars 2016 s'il clique sur le bouton **ANNÉE SUIVANTE** >>.

**Attention** : Si la date est impossible le jour transmis doit être 1.

Par exemple, si la date qui a été transmise est le 29 février 2000, alors cette fonction doit charger la même page Web mais en lui transmettant la date du 1<sup>er</sup> février 2001 s'il clique sur le bouton **ANNÉE SUIVANTE** >>.

Lorsque l'utilisateur clique sur le bouton << **MOIS PRÉCÉDENT**, *binMois* vaut *true* et *intSens* vaut -1 et lorsqu'il clique sur le bouton **MOIS SUIVANT** >>, *binMois* vaut *true* et *intSens* vaut 1.

Cette fonction doit charger la même page Web mais en lui **transmettant manuellement** la date du mois précédent ou du mois suivant selon le cas.

Par exemple, si la date qui a été transmise est le 7 mars 2015, alors cette fonction doit charger la même page Web mais en lui transmettant la date du 7 février 2015 s'il clique sur le bouton << **MOIS PRÉCÉDENT** ou en lui transmettant la date du 7 avril 2015 s'il clique sur le bouton **MOIS SUIVANT** >>.

Par exemple, si la date qui a été transmise est le 18 janvier 2000, alors cette fonction doit charger la même page Web mais en lui transmettant la date du 18 décembre 1999 s’il clique sur le bouton << MOIS PRÉCÉDENT ou en lui transmettant la date du 18 février 2000 s’il clique sur le bouton MOIS SUIVANT >>.

**Attention** : Si la date est impossible, le jour transmis doit être 1.

Par exemple, si la date qui a été transmise est le 30 janvier 2000, alors cette fonction doit charger la même page Web mais en lui transmettant la date du 1<sup>er</sup> février 2000 s’il clique sur le bouton MOIS SUIVANT >>.

Lorsque l’utilisateur clique sur le bouton AUJOURD’HUI, *intSens* vaut 0.

Cette fonction doit charger la même page Web mais en lui transmettant manuellement la date d’aujourd’hui.

#### 4. Modification de la fonction *chargePageSelectionneDate()*.

```
function chargePageSelectionneDate()
```

Cette fonction doit charger la page Web **selectionne-date.htm** mais, cette fois-ci, en lui retransmettant manuellement la date qui a été transmise à la page Web **affiche-calendrier.htm**.

Par exemple, si le calendrier affiché est le calendrier suivant, alors la page Web **affiche-calendrier.htm** doit transmettre manuellement, à la page Web **selectionne-date.htm**, la date du 28 août 2002.

Août 2002						
Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

5. Modification de la fonction *initialiseAvecDateCourante()* (sur la page Web **selectionne-date.htm**)

Si aucune date n'a été transmise à la page Web **selectionne-date.htm**, cette fonction doit (1) récupérer la date du jour (jour, mois et année) puis (2) initialiser les quatre listes déroulantes à partir de cette dernière.

Si une date a été transmise à la page Web **selectionne-date.htm**, cette fonction doit (1) récupérer la date transmise puis (2) initialiser les quatre listes déroulantes à partir de cette dernière.

Par exemple, si la date transmise est celle du 28 août 2002, voici ce qui doit être affichée sur la page Web.

ACTUALISER	28 ▼	Août ▼	20 ▼	02 ▼	CALENDRIER
------------	------	--------	------	------	------------

**Mercredi, le 28 août 2002**

### 4-3. Contraintes

- Vos identificateurs doivent être significatifs. Il est impératif de respecter la convention de noms. Vos chaînes de caractères doivent être entre apostrophes. Vous pouvez perdre jusqu'à 20 pts si vous ne respectez pas ces contraintes.
- Vous ne devez pas modifier ce qui a déjà été tapé. Vous devez modifier seulement ce qui est demandé. Vous pouvez perdre 20 pts si vous ne respectez pas cette contrainte.
- Par contre, vous pouvez modifier le style dans le fichier **styleProjet2.css** (couleurs, polices de caractères, mise en page etc.). Allez-y mollo!!!
- Vous pouvez également ajouter toutes les fonctions que vous voulez (à la condition que celles-ci n'aient pas déjà été programmées dans **date.js** et dans **libUtilitaires.js**).
- Vous devez **TOUJOURS UTILISER** les fonctions qui ont déjà été programmées dans **libUtilitaires.js** et **date.js** lorsque cela s'applique. Ne réinventez pas la roue.

Vous pouvez perdre jusqu'à 30 pts si vous ne respectez pas cette contrainte.

### 5. Fichiers à copier

- Le dossier **Projet2** compressé sur *LÉA*.

## 6. Barème de correction

Ce projet sera corrigé sur 100 et comptera pour 9% de la session.

- ☐ **Bon fonctionnement du programme et validité des résultats** ..... 100 points
- ☐ **Modification du *XHTML* déjà tapé et/ou modification de la librairie libUtilitaires.js**..... -20 points
- ☐ **Absence de votre nom aux endroits appropriés** ..... jusqu'à -05 points
- ☐ **Non utilisation des fonctions déjà programmées dans date.js et/ou dans libUtilitaires.js** ..... jusqu'à -30 points
- ☐ **Projet non remis sur LÉA dans un dossier compressé** ..... -05 points
- ☐ **Présentation du code source** (identificateurs significatifs, convention des noms respectée, etc.) ... jusqu'à -20 points
- ☐ **Retard**..... -10% par jour (maximum 5 jours)
- ☐ **Plagiat partiel ou total** ..... -100 points pour tous les étudiants impliqués