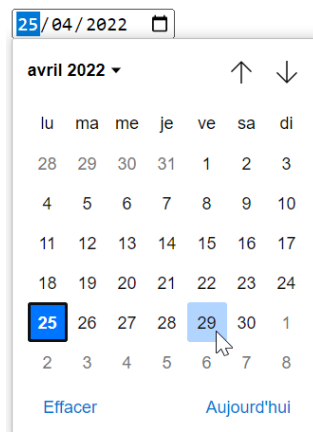


TP de synthèse

Datepicker

Ce TP consiste à ré-implémenter à la main un composant graphique appelé « Datepicker », permettant de choisir une date en la sélectionnant dans un calendrier.

Les navigateurs affichent automatiquement ce composant quand on clique sur un élément de formulaire de type date. Le style des Datepickers est propre à chaque navigateur et dépend du contexte. Par exemple, la version « bureau » du Datepicker de Chrome est illustrée ci-dessous.



L'objectif de ce TP est de faire la synthèse de tout ce que vous avez appris en JavaScript mais aussi en HTML/CSS (grilles CSS notamment).

Il est volontairement peu guidé : il permet également de faire le point sur votre capacité à lire une documentation et à créer un projet (correctement structuré) de zéro.

Compétences travaillées dans ce TP

- Structuration du code
- Lecture d'une documentation
- Algorithmique et JavaScript de base
- Programmation événementielle
- API DOM
- HTML / CSS

1 – Préliminaires

Cette partie consiste à implémenter quelques fonctions utilitaires sur les dates. Nous vous conseillons de structurer votre code comme nous l'avons vu dans les TP sur les bases du JavaScript :

- associer une fonction de test `fTest()` à chaque fonction `f()`,
- séparer le programme principal, les fonctions utilitaires et les tests.

Année bissextile

Implémentez une fonction :

`isLeap(year)`

qui renvoie `true` si l'année `year` est bissextile.

Exemples d'utilisation :

```
console.log( isLeap(2016) ); // => true
console.log( isLeap(2022) ); // => false
console.log( isLeap(2100) ); // => false
```

Nombre de jours dans un mois

Implémentez une fonction :

`getNbDays(year, month)`

qui renvoie le nombre de jours du mois `month` (entier de 1 à 12, 1 = janvier) de l'année `year`.

Exemples d'utilisation :

```
console.log( getNbDays(2022, 1) ); // => 31
console.log( getNbDays(2022, 2) ); // => 28
console.log( getNbDays(2022, 3) ); // => 31
console.log( getNbDays(2022, 4) ); // => 30

// années bissextiles
console.log( getNbDays(2016, 2) ); // => 29
console.log( getNbDays(2020, 2) ); // => 29
```

Numéro d'un jour dans la semaine

Implémentez une fonction :

`getDayInWeek(year, month, day)`

qui renvoie le numéro du jour donné en argument dans la semaine. Ce numéro doit varier entre 1 et 7, 1 correspondant à lundi, 2 à mardi, ... jusqu'à 7 pour dimanche.

Exemples d'utilisation :

```
// facilement vérifiable !
console.log( getDayInWeek(2022, 4, 25) ); // => 1 (lundi)

// quelques événements historiques
console.log( getDayInWeek(1989, 11, 9) ); // => 4 (jeudi)
console.log( getDayInWeek(1969, 7, 20) ); // => 7 (dimanche)
console.log( getDayInWeek(1945, 5, 8) ); // => 2 (mardi)
```

Aide :

On peut coder cette fonction en une ligne en s'appuyant sur le type [Date](#). Regardez bien les méthodes de ce type.

2 – Générer le calendrier

Implémentez une fonction :

```
buildCalendar(element, year, month)
```

qui génère un calendrier situé au mois `month` (entier de 1 à 12, 1 = janvier) de l'année `year` et remplace le contenu de l'élément HTML `element` par ce calendrier.

Le calendrier doit comporter les éléments suivants :

- le mois et l'année
- les noms des jours de la semaine, en commençant par lundi (possiblement abrégés)
- les numéros des jours du mois, positionnés en grille, de telle manière qu'il y ait correspondance entre le numéro et le jour de la semaine

Exemple de rendu pour `buildCalendar(calendar, 2022, 4)` :

avril 2022						
lun	mar	mer	jeu	ven	sam	dim
				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	

calendar

Conseils :

- Pour organiser les éléments, le mieux est d'utiliser une [grille CSS](#). Si vous n'êtes pas à l'aise, vous pouvez utiliser [table HTML](#) dans un premier temps.
- Ne perdez pas de temps avec le style ici : définissez les règles minimales permettant d'avoir un résultat correctement organisé. Votre priorité doit être d'avoir un calendrier **fonctionnel**. Le style (en particulier le centrage du texte) pourra être amélioré ultérieurement.

Amélioration

Ajoutez un paramètre `day` à la fonction `buildCalendar()` afin de mettre en valeur le jour indiqué dans le calendrier. Si `day` n'est pas renseigné, votre fonction ne doit pas planter, mais simplement ne rien mettre en valeur.

Exemple de rendu pour `buildCalendar(calendar, 2022, 4, 25)` :

avril 2022						
lun	mar	mer	jeu	ven	sam	dim
				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	

3 – Donner vie au calendrier

Mise en valeur au survol

Modifiez votre code pour que la case survolée par le pointeur de la souris soit mise en valeur (d'une manière différente que dans la partie précédente).

Exemple :

avril 2022						
lun	mar	mer	jeu	ven	sam	dim
				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	

Note : Ceci peut être fait en utilisant uniquement CSS.

Petites améliorations possibles :

- Ne rien faire sur des cases non cliquables (cases vides, nom des jours, ...)
- Changer l'apparence du pointeur avec la propriété CSS [cursor](#).


Sélection d'une journée

Ajoutez une balise `<input>` de type `date` à votre page.

Faites en sorte que le clic sur un jour du calendrier affecte la valeur de cette balise à ce jour.

Contrainte : il est interdit d'utiliser des variables globales.

Exemple :

29/04/2022 						
avril 2022						
lun	mar	mer	jeu	ven	sam	dim
				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	

Attention :

- la valeur d'une balise `<input>` de type « date » doit respecter un format décrit [ici](#).
- si des nombres sont stockés sous forme de chaîne de caractères, il ne faut pas oublier d'appliquer une conversion de type avant de les utiliser.

[Facultatif] Utilisation d'attributs universels

Les attributs universels `data-*` permettent de stocker des informations arbitraires dans les éléments HTML.

Par exemple, on peut stocker le jour associé à une case `cell` du calendrier dans un attribut `data-day`, que l'on peut manipuler en JavaScript par `cell.dataset.day`.

Utilisez des attributs universels pour stocker et retrouver de manière plus élégante des informations au clic.

La [documentation du MDN](#) peut vous être utile.

[Facultatif] Navigation dans les mois

- Dans l'interface graphique du calendrier, ajoutez deux éléments de navigation permettant d'aller au mois précédent et suivant
- Faites en sorte qu'un clic sur « précédent » ou « suivant » modifie le contenu du calendrier
- Exemple :

<	avril 2022						>
lun	mar	mer	jeu	ven	sam	dim	
				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		

→

<	mai 2022						>
lun	mar	mer	jeu	ven	sam	dim	
						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	

Conseil : Attention aux extrémités d'année !