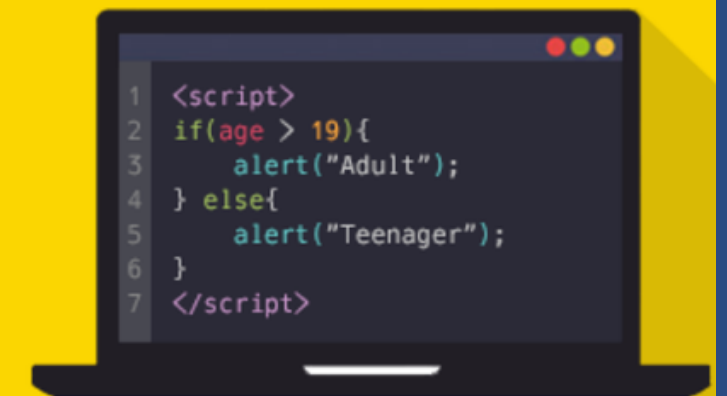


Javascript

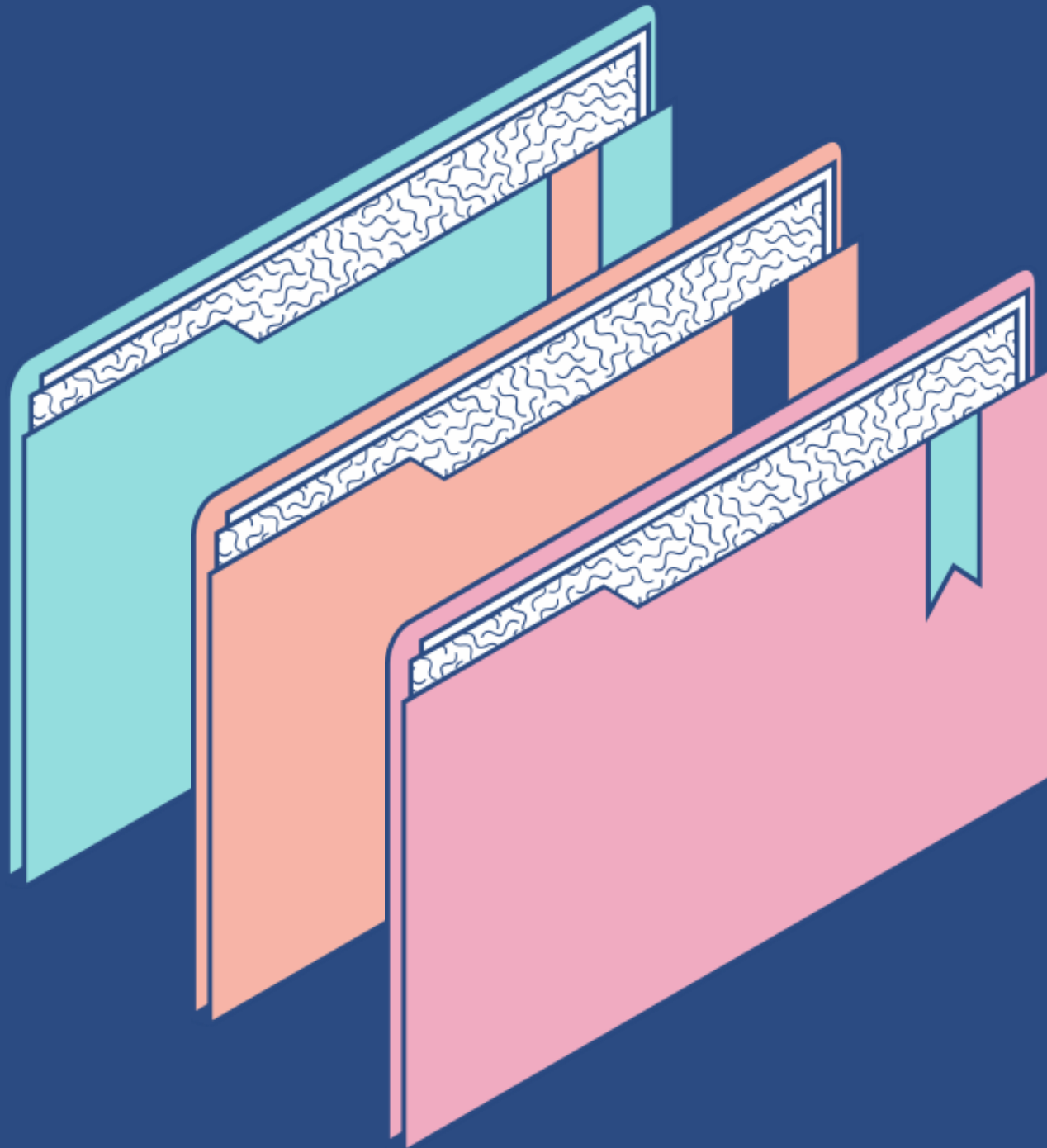


JavaScript



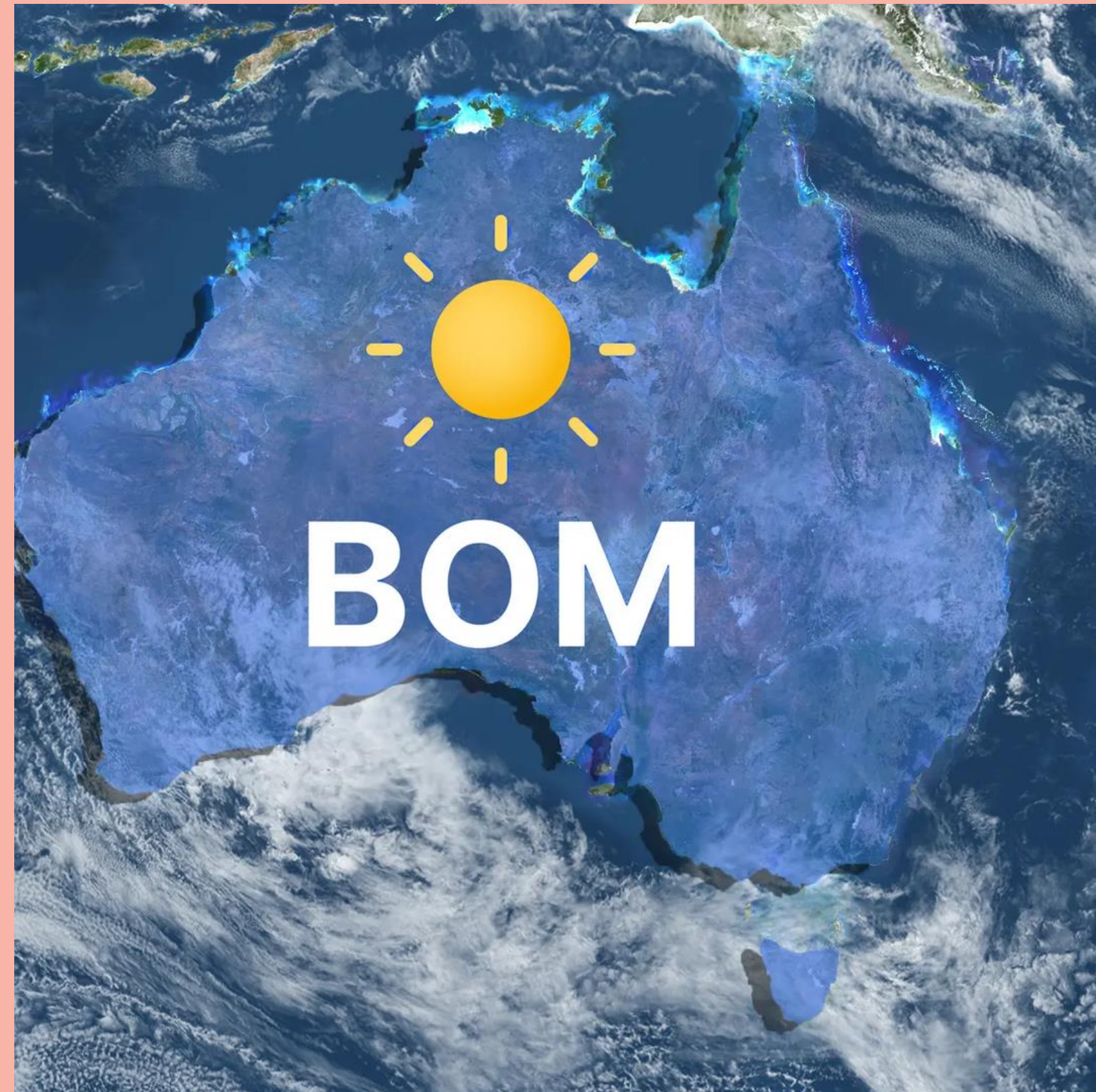
SOMMAIRE

- Présentation du JS
- Variables
- Structures de contrôle
- Fonctions
- POO en JS
- Valeurs primitives
- Manipulation du BOM
- Manipulation du DOM
- Fonctions avancées
- Stockage de données persistantes
- Canvas
- Asynchrone



Manipulation du BOM

92





Présentation des API

Une API est une interface (ensemble de code par laquelle un logiciel fourni des services à des clients)

L'API peut être comparée à une commande de voiture.

Quand on appui sur l'accélérateur, une grande quantité de mécanismes (cachés) se mettent en place, pour vous livrer directement le résultat obtenu (accélération de la voiture).

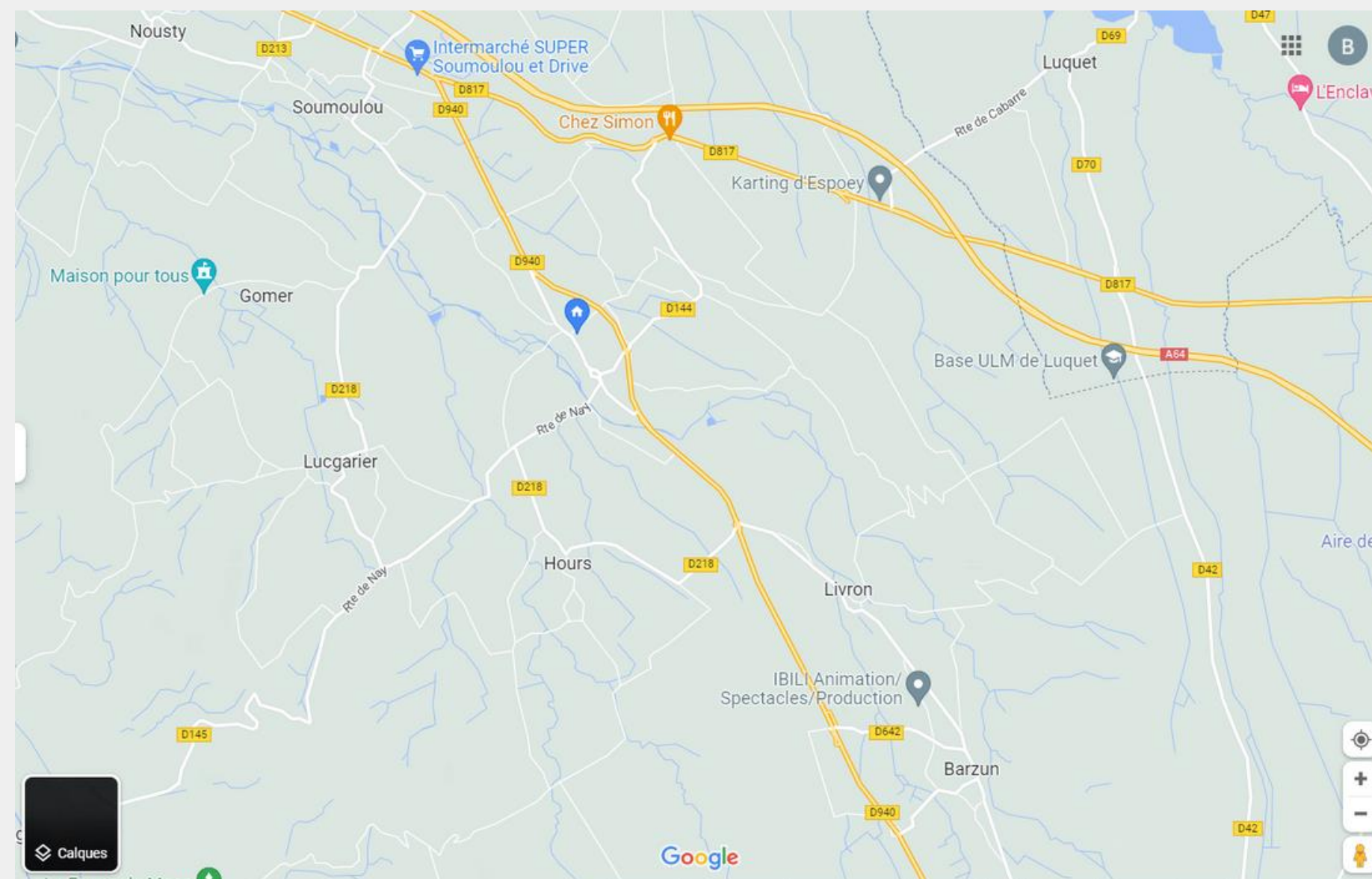


Présentation des API

Les API JS se classent en 2 grandes catégories :

- Les API intégrées au navigateurs, utilisables immédiatement (que nous allons étudier par la suite)
- Les API externes, proposées développées par des entreprises dans le but de simplifier l'utilisation de leurs services (API Google Map, Twitter, Youtube ...)

Présentation des API





Présentation du BOM (Browser Object Model)

Le BOM est une API renvoyant un objet Window, mise en place par tous les navigateur, et fournissant une multitude de services utiles.

Cet objet Window possède lui même plusieurs objets (enfants) :

- L'objet Navigator qui représente l'état du navigateur et qu'on va utiliser avec l'API Geolocation ;
 - L'objet History qui permet de manipuler l'historique de navigation du navigateur
 - L'objet Location qui fournit des informations relatives à l'URL de la page courante ;
- L'objet Screen qui nous permet d'examiner les propriétés de l'écran qui affiche la fenêtre courante ;
 - L'objet Document et le DOM dans son ensemble que nous étudierons en détail dans la suite.



Objet Window

L'objet Window représente la fenêtre du navigateur.

Il possède de nombreuses propriétés utiles, notamment :

- `innerHeight` / `innerWidth` qui retournent respectivement la hauteur et largeur de la partie visible de la fenêtre de navigation
- `outerHeight` / `outerWidth` qui retournent respectivement la hauteur et la largeur complète de la fenêtre de navigation (avec options du navigateur, barre d'URL etc...)

```
document.getElementById('p1').innerHTML =  
  'Taille de la fenêtre (ext) : ' + window.outerWidth + '*' + window.outerHeight;  
  
document.getElementById('p2').innerHTML =  
  'Taille de la fenêtre (int) : ' + window.innerWidth + '*' + window.innerHeight;
```




Objet Window

L'objet Window possède également de nombreuses méthodes, notamment :

- `open(x,y)`, permettant d'ouvrir la page x dans une seule nouvelle fenêtre/onglet/iframe.

On pourra ajouter l'option `_blank` pour qu'une nouvelle fenêtre s'ouvre à chaque appel de `open(x)`.

On pourra également ajouter les spécificités y de la fenêtre à ouvrir.

- `close()`, permettant de fermer cette page

```
let b1 = document.getElementById('b1'); //On accède au bouton #b1
let winSize = 'width=500, height=500';

fenetre = window.open('https://www.codepen.io/', '', winSize);
fenetre.close();
```



Objet Window

L'objet Window possède également de nombreuses méthodes, notamment :

- `resizeBy(x,y)` : permet de soustraire une largeur x et une hauteur y à la fenêtre ouverte par `open()`
- `resizeTo(x,y)` : permet de redimensionner la fenêtre ouverte par `open` à la largeur x et hauteur y

```
}  
function resizeWindowBy(){  
    fenetre.resizeBy(200, 100);  
}  
function resizeWindowTo(){  
    fenetre.resizeTo(960, 720);  
}
```

Certains navigateurs bloquent ces fonctionnalités par raison de sécurité. Il est tout de même possible de tester en retirant l'URL de `open()`



Objet Window

Enfin, l'objet Window possède des méthodes très utiles pour afficher des alerts, boîtes de dialogue et confirmation de fenêtre :

- `alert(x)` : Affiche une alerte contenant le text x.
- `prompt()` : Permet à l'utilisateur de saisir un texte dans une boîte de dialogue. Return la saisie.
- `confirm(x)` : permet de demander une confirmation à l'utilisateur. Return true or false.

```
if(confirm("En savoir plus ?")){  
    alert("warning");  
    prompt("compris ?");  
}
```



Exercice d'application

Création Window

Créez une fonction

Ajoutez un message de confirmation "voulez vous ouvrir une nouvelle fenêtre ?"

En cas de validation de cette confirmation, la fonction ouvrira une nouvelle fenêtre de 700px de hauteur de 700px de largeur.

Cette fenêtre ouvrira la page de l'adrrar : <https://www.adrrar-formation.com/>

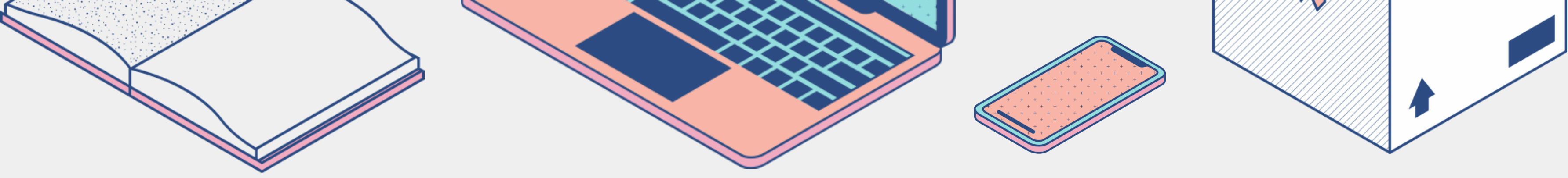
Objet Navigator

L'objet Navigator (propriété de Window), donne accès aux informations sur le navigateur de vos visiteurs (langue, activation des cookies, modèle ...).

Il possède de nombreuses propriétés intéressantes, notamment :

- language : retourne la langue définie dans le navigateur
- geolocation : retourne un objet Geolocation définissant la localisation de l'utilisateur
- cookieEnabled : détermine si les cookies sont autorisés ou non, Return true or false
- platform : retourne la plateforme utilisée par le navigateur

```
'Langue du navigateur : ' + navigator.language
```



Objet Geolocation

L'objet Geolocation (enfant de Navigator) permet de géolocaliser l'appareil de l'utilisateur.

Cet objet possède 3 méthodes utiles, disponibles uniquement en HTTPS par raison de sécurité :

- `getCurrentPosition()` : Return un objet `Position` contenant la position actuelle de l'utilisateur.
- `watchPosition()` : Permet de définir une fonction qui sera appelée dès que l'appareil change de position. Return un id utilisable par la méthode `clearWatch`.
- `clearWatch(id)` : permet de supprimer la surveillance mise en place par `watchPosition()`.



Objet Geolocation

L'objet Position (retourné par `getCurrentPosition()`) possède 2 propriétés :

- `coords`, qui retourne un objet `Coordinates` avec les coordonnées de l'appareil.
 - `timestamp`, qui représente le moment où la position a été récupérée.



Objet Geolocation

Enfin, l'interface Coordinates (Propriété de Position), implémente les propriétés suivantes :

- latitude qui représente la latitude de l'appareil ;
- longitude qui représente la longitude de l'appareil ;
- altitude qui représente l'altitude de l'appareil ;
- accuracy qui représente le degré de précision (en mètres) des valeurs renvoyées par les propriétés latitude et longitude ;
- heading qui représente la direction dans laquelle l'appareil se déplace (degrés par rapport au Nord)
 - speed qui représente la vitesse de déplacement de l'appareil (en m/s)

Objet Geolocation

```
function coordonnees(pos) {  
    let crd = pos.coords;  
  
    let latitude = crd.latitude;  
    let longitude = crd.longitude;  
  
    document.getElementById('p1').innerHTML= 'Latitude : ' + latitude.toFixed(2);  
    document.getElementById('p2').innerHTML= 'Longitude : ' + longitude.toFixed(2);  
}  
  
navigator.geolocation.getCurrentPosition(coordonnees);
```

file:/// wants to

Know your location

Block

Allow

Latitude : 43.13

Longitude : 5.93



Exercice d'application Objet Geolocation

Créez une fonction affichant la distance(différence de longitude et de latitude) et la différence d'altitude entre votre position et le sommet de la tour eiffel



Objet History

L'objet History permet de manipuler l'historique du navigateur des visiteurs POUR LA SESSION COURANTE.

Il ne s'agit pas de l'ensemble de l'historique, mais uniquement des pages visitées au sein de l'onglet dans lequel la page est ouverte.

Certains navigateurs bloquent l'exécution de cet objet



Objet History

Propriétés et méthodes utiles :

- Length() : Return le nombre d'éléments dans l'historique de l'onglet.
- go(x) qui permet de charger la page ayant une position x dans l'historique.
 - back() qui permet de charger la page précédente dans l'historique
 - forward() qui permet de charger la page suivante dans l'historique.

```
function hgo(){  
    history.go(-2);  
}  
function hback(){  
    history.back();  
}  
function hforward(){  
    history.forward();  
}
```



Objet Location

L'interface Location fourni des informations relatives à l'URL de notre page.

On peut y accéder à partir des interfaces Window ou Document (Window.location / Document.location)



Objet Location

L'objet Location possède de nombreuses propriétés, notamment :

- hash, qui retourne la partie ancre d'une URL si l'URL en possède une ;
- search, qui retourne la partie recherche de l'URL si l'URL en possède une ;
 - pathname, qui retourne le chemin de l'URL précédé par un / ;
 - href, qui retourne l'URL complète ;
 - hostname, qui retourne le nom de l'hôte ;
 - port, qui retourne le port de l'URL ;
 - protocole, qui retourne le protocole de l'URL ;
 - host, qui retourne le nom de l'hôte et le port relatif à l'URL ;
- origin, qui retourne le nom de l'hôte, le port et le protocole de l'URL.

Objet Location

L'objet Location possède 3 méthodes intéressantes :

- La méthode `assign()` qui va charger une ressource à partir d'une URL passée en argument ;
 - La méthode `reload()` qui va recharger le document actuel ;
- La méthode `replace()` qui va remplacer le document actuel par un autre disponible à l'URL fournie en argument.

```
function recharge(){
    location.reload();
}
function assigne(){
    location.assign('https://www.pierre-giraud.com');
}
function remplace(){
    location.replace('https://www.pierre-giraud.com');
}
```



Exercice d'application

Objet Location

Créez une fonction stockant les éléments suivants au sein d'un tableau :

- Nom d'hôte
 - Port
- protocole
- URL complète

Affichez votre tableau, puis proposez à votre utilisateur d'être redirigé.

S'il accepte, il sera redirigé sur la page de l'Adrar.



Objet Screen

L'objet Screen donne les informations sur l'écran de vos visiteurs (taille, résolution ...)

L'objet Screen possède 6 propriétés intéressantes :

- width : retourne la largeur totale de l'écran ;
- availWidth : retourne la largeur de l'écran moins celle de la barre de tâches ;
- height : retourne la hauteur totale de l'écran ;
- availHeight : retourne la hauteur de l'écran moins celle de la barre de tâches ;
- colorDepth : retourne la profondeur de la palette de couleur de l'écran en bits ;
- pixelDepth : retourne la résolution de l'écran en bits par pixel.



Objet Screen

```
'Dimensions totales de l\'écran : ' + screen.width + '*' + screen.height +  
'<br>Surface disponible : ' + screen.availWidth + '*' + screen.availHeight +  
'<br>Palette de couleur : ' + screen.colorDepth +  
'<br>Résolution : ' + screen.pixelDepth;
```

Dimensions totales de l'écran : 1920*1080
Surface disponible : 1920*1057
Palette de couleur : 24
Résolution : 24