

# Javascript – Avancé



Bailly Benjamin



# Les api

Les API, qu'elles soient publiques ou non, existent toutes pour la même raison. Elles facilitent les nouveaux développements en s'appuyant sur une application existante.

Par exemple: l'API Twitter vous permet d'afficher vos derniers tweets sur votre site Web.

Elle fournit un ensemble de constructions que vous pouvez utiliser pour interroger le service Twitter et qui renvoi alors les informations demandées.

# Ajax

L'objectif de l'AJAX est de faire en sorte qu'une page web communique avec le serveur web sans que cela n'engendre un rechargement de la page.

C'est d'ailleurs pour cette raison que le langage préféré de l'AJAX est le JavaScript.

C'est via des codes JavaScript que la connexion entre la page web et le serveur s'établit.

Je vais vous donner un exemple de fonctionnement script ajax -> API.

# Tp – 28 Api météo

Doc de l'api : <https://openweathermap.org/current> (pour l'utiliser il faut s'inscrire)

Lisons la doc ensemble et décortiquons l'url de l'api pour la requête Ajax et l'utilisation de l'API

<https://api.openweathermap.org/data/2.5/weather?q=montpellier&appid=37eeec8daf48240d150557b608d1ee76&units=metric>

Q=montpellier pour la ville

Appid = pour la clé de l'api

Units = pour l'unité de mesure (ici celsius)

# Tp – 28 Api météo

## Étape – 1

- Construire la requête AJAX à l'aide du code que l'on a vu lors de l'utilisation de l'API bitcoin.
- Il faut vraiment comprendre l'utilisation de l'api météo et surtout faire passer les bonnes infos dans l'url. (concaténation).
- Testez avec des « console.log » et ne passez pas à l'étape suivante temps que vous n'arrivez pas à faire fonctionner votre requête.
- Pensez à stocker la requête dans une fonction.

# Tp – 28 Api météo

## Étape – 2

- Maintenant que vous arrivez à récupérer la météo pour une ville, je vous laisse changer la température.
- Ainsi que le nom de la ville.

# Tp – 28 Api météo

## Étape – 3

- Ajoutez un évènement au bouton « changer de ville » afin de pouvoir choisir une ville et la stocker à l'aide d'une boîte de dialogue (prompt).
- Rajouter la librairie AOS pour mettre une animation sur le bouton « changer de ville »
- Afficher la date et l'heure au format français dans la console.
  
- Conseil :
- En cas d'erreur sur la requête api c'est que la ville n'existe pas donc vous pouvez dans ce cas obliger l'utilisateur à rentrer encore une ville tant que c'est faux.
- Il y a plusieurs façons de faire, la plus simple reste à passer en paramètre de la fonction la ville rentrée par l'utilisateur.
  
- EXO BONUS, pour ceux qui ont finis :
- Pour gérer les paramètres de votre API comme la clé, la ville, l'unité et l'url essayez de créer une classe myApiMeteo
- EXO BONUS BONUS
- Gérez tout cela avec async et await



# Tp – 28 Api météo

Je vous envoie le fichier JS pour la correction.



# Tp – 29 Regex

Je vous envoie un pdf pour les consignes, Renseignez vous sur :

Ce qu'est un regex.  
Comment ça marche.  
La syntaxe.

Liens utiles :

- <https://regexr.com/>
- <https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/918834-memento-des-expressions-regulieres>

Pour la correction, je vous envoie le projet au format ZIP

# LocalStorage et SessionStorage

« sessionStorage » mémorise les données sur la durée d'une session de navigation, et sa portée est limitée à la fenêtre ou l'onglet actif.

« localStorage » mémorise les données sans limite de durée de vie. Contrairement à sessionStorage, elles ne sont pas effacées lors de la fermeture d'un onglet ou du navigateur.

**setItem(clé,valeur)** : stocke une paire clé/valeur

**getItem(clé)** : retourne la valeur associée à une clé

**removeItem(clé)** : supprime la paire clé/valeur en indiquant le nom de la clé

**key(index)**: retourne la clé stockée à l'index spécifié

**clear()**: efface toutes les paires

# LocalStorage et SessionStorage

Exemple : Essayez ce code, vous verrez que l'utilisateur sera demandé une seule fois.

```
if(localStorage.getItem("user")){  
    document.body.append(`Bonjour ${localStorage.getItem("user")}`);  
}  
else{  
    let user = prompt("Quel est votre pseudo ? ");  
    localStorage.setItem("user", user);  
    document.body.append(`Bonjour ${user}`);  
}
```

# LocalStorage et SessionStorage tp-30

A l'aide de localStorage Créez le code js qui permettra de sauvegarder le thème choisis de la page à l'aide du bouton (sombre, lumineux).