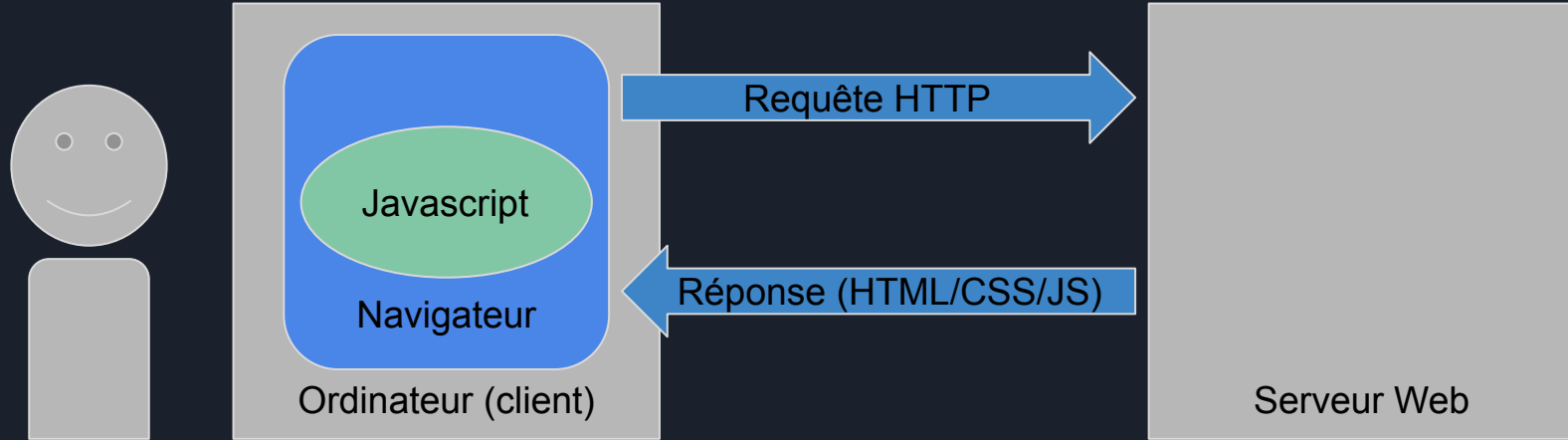




Javascript

Requêtes HTTP

Le modèle client-serveur





Le protocole HTTP

- HyperText Transfer Protocol
- Protocole de communication
- À la base du Web
- HTTPS (chiffrement TLS)
- Actuellement à la version 2



Le protocole HTTP

Composition d'une requête HTTP :

- Le verbe HTTP:
 - GET
 - POST
 - PUT
 - DELETE
 - ...
- URL
- Version
- En-tête
- Corps



Le protocole HTTP

Composition d'une réponse HTTP :

- Version
- Code de status:
 - 20x : OK
 - 30x : redirections
 - 40x : erreurs client (dont la fameuse 404)
 - 50x : erreurs serveur
 - ...
- En-tête
- Corps



JSON

- Javascript Object Notation
- Format normé de structuration de données:
 - clé - valeur
 - types de données: object, number, boolean, string, null, array
- Très proche des objets Javascript
- Utilisé dans tous les langages de programmation
- Pratique pour les échanges de données client / serveur
 - sérialisation / désérialisation d'objets
- Peut être utilisé dans le corps des requêtes et réponses HTTP



JSON: exemple

```
{
  "firstName": "Bertrand",
  "lastName": "Bon",
  "languages": [{"name": "Français", "skill": 10},
                 {"name": "Anglais", "skill": 8},
                 {"name": "Javascript", "skill": 9}],
  "car": null,
  "age": 29,
  "isSleeping": false,
  "computer": {
    "processor": "i7-6700HQ",
    "graphicCard": "970gtx",
    "RAM": "12GB"
  }
}
```



Autres formats de structuration de données

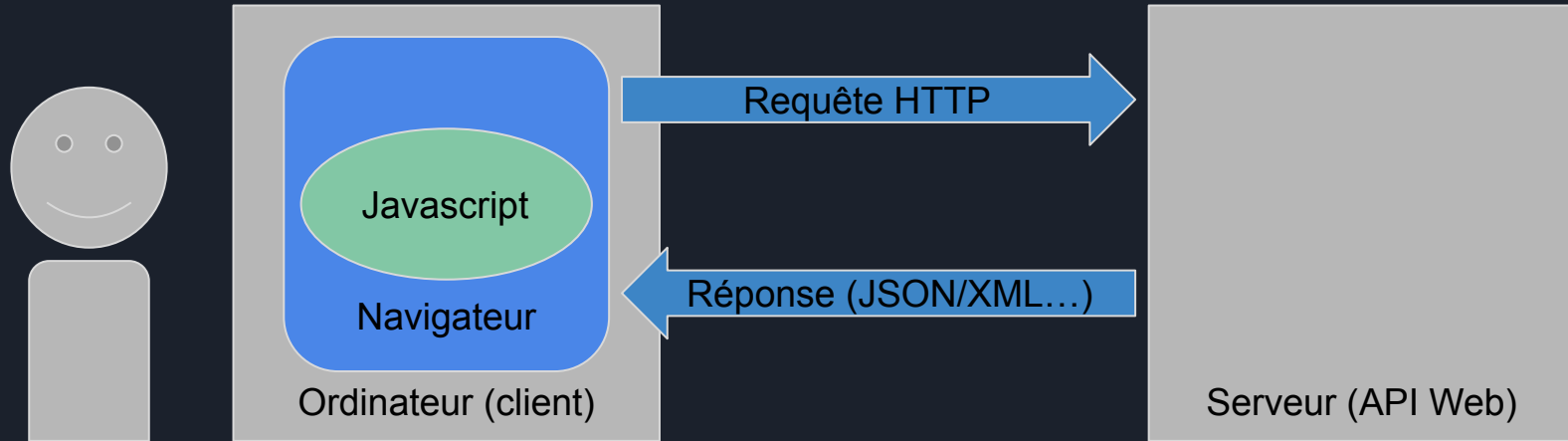
- XML
- YAML
- protobuf
- ...



API

- Application Programming Interface
- Terme générique, interface permettant à deux systèmes différents d'interagir. Un logiciel offre ses services à un autre via l'API qu'il lui présente
- Un serveur web peut présenter une API, avec laquelle il est possible d'interagir via des requêtes HTTP. Elle peut-être publique, et est dans ce cas documentée
- Exemple: l'API web de Twitter, qui permet par exemple:
 - de rechercher et récupérer des tweets pour pouvoir effectuer un traitement dans sa propre application
 - de publier des tweets
 - de gérer des campagnes de publicité sur Twitter
 - ...
 - documentation: <https://developer.twitter.com>


API Web





Javascript: Fetch

- [Documentation](#)
- Successeur de XMLHttpRequest
- Permet d'effectuer des requêtes HTTP en Javascript



● Exercice : Requêter une API web (météo)

- [Documentation de l'API utilisée](#)
- Exemple: Données météo pour Toulouse: <http://www.prevision-meteo.ch/services/json/toulouse>
- Conseil pour bien visualiser la structure: copiez le JSON dans un fichier, sauvegardez le avec l'extension .json, ouvrez le avec VSCode et utilisez l'auto-formatage

● Exercice : Requête une API web (météo)

1. Créez un fichiers `meteo.html` avec le contenu suivant:


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>API Meteo</title>
  </head>
  <body>
    <h1>API Meteo</h1>
    <form>
      <label for="city">Ville:</label><input name="city" type="text">
      <input type="submit" value="Envoyer"></form>
      <div id="display"></div>
      <script src="meteo.js"></script>
    </body>
</html>
```

● Exercice : Requêter une API web (météo)

2. Créez un fichier `meteo.js` avec le contenu suivant:

```
const url = "https://www.prevision-meteo.ch/services/json/";

function requestApi(event) {
  event.preventDefault();
  const city = document.querySelector("form input[name='city']");
  console.log(city.value);
  fetch(`${url}${city.value}`).then(response => response.json())
    .then(data => {
      console.log('Success:', data);
      /* Ajoutez ici le traitement du JSON contenu dans data ! */
    })
    .catch((error) => {
      console.error('Error:', error);
    });
}
```



● Exercice : Requêter une API web (météo)

3. Modifiez `meteo.js` pour faire en sorte d'utiliser le JSON reçu pour créer une interface présentant la météo de la ville (captures d'écran du résultat à obtenir page suivante)

● Exercice : Requêter une API web (météo)

API Meteo

Ville:

Meteo pour Toulouse, France

21.07.2020, 01:00

Conditions actuelles:



Température: 25°
Vent: 13 km/h, SO
Nuit claire

Prévisions à 4 jours:

Mercredi, 22.07.2020:



Développement nuageux, 22.07.2020:
Température minimum: 22, température maximum: 39

Judi, 23.07.2020:



Eclaircies, 23.07.2020:
Température minimum: 21, température maximum: 37

Prévisions à 4 jours:

Mercredi, 22.07.2020:



Développement nuageux, 22.07.2020:
Température minimum: 22, température maximum: 39

Judi, 23.07.2020:



Eclaircies, 23.07.2020:
Température minimum: 21, température maximum: 37

Vendredi, 24.07.2020:



Ensoleillé, 24.07.2020:
Température minimum: 19, température maximum: 33

Samedi, 25.07.2020:



Eclaircies, 25.07.2020:
Température minimum: 17, température maximum: 32

● Exercice : Requêter une API web (météo)

4. Ajoutez la météo heure par heure pour chaque jour:

Aide:

- l'attribut contenant les données de chaque heure dans le JSON commence par un chiffre (exemple: 7H00). Cela empêche d'utiliser l'opérateur point pour y accéder. À la place, vous aurez besoin de l'opérateur crochet (exemple:
`data.fcst_day_1.hourly_data[`7H00`])` ([Doc](#))
- Il y a 24 heures dans un jour. Il peut-être judicieux d'essayer d'utiliser une boucle for pour accéder aux attributs 0H00 à 23H00 (avec un compteur i de 0 à 23)

Mercredi, 22.07.2020:



Développement nuageux, 22.07.2020:

Température minimum: 22, température maximum: 39

Heure par heure:

0H00  Nuit claire

1H00  Nuit claire

2H00  Nuit claire

3H00  Nuit claire

4H00  Nuit claire

5H00  Nuit claire

6H00  Nuit claire

7H00  Ensoleillé

8H00  Ensoleillé

9H00  Ensoleillé