AJAX

ASYNCHRONOUS JAVASCRIPT AND XML

COMMUNICATION ENTRE JAVASCRIPT ET XML
DE FAÇON ASYNCHRONE

PRÉSENTATION ET UTILITÉ

AJAX S'APPUIE SUR LES TECHNOLOGIES SUIVANTES :

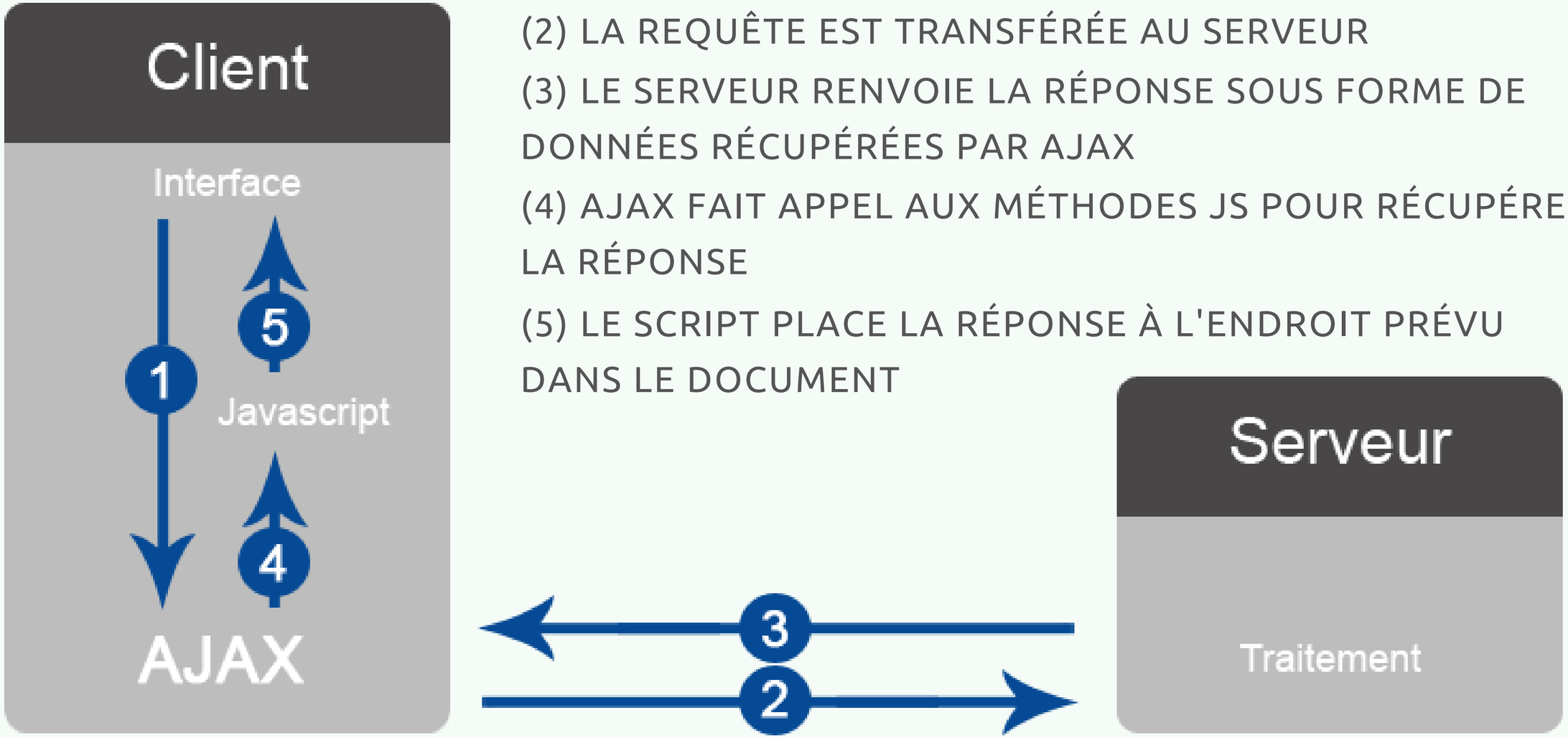
- **Javascript** et le **DOM** pour le traitement des données
- **XML** ou **JSON** pour l'extraction des données (le contenu à afficher)
- **HTML** et **CSS** pour la présentation des données

L'AJAX est un ensemble de technologies visant à effectuer des transferts de données.

APPLICATIONS COURANTES D'AJAX :

- Mise à jour d'une (ou plusieurs) partie de la page web sans recharger entièrement la page
- Suggestion automatique lors de la saisi d'un champ
- Sauvegarde automatique d'un texte dans la base de données lors de la saisie
- Upload d'un fichier sur le serveur tout en visualisant l'état de progression du chargement

FONCTIONNEMENT



(1) LE CLIENT DÉCLENCHE UN ÉVÉNEMENT

(2) LA REQUÊTE EST TRANSFÉRÉE AU SERVEUR

(3) LE SERVEUR RENVOIE LA RÉPONSE SOUS FORME DE DONNÉES RÉCUPÉRÉES PAR AJAX

(4) AJAX FAIT APPEL AUX MÉTHODES JS POUR RÉCUPÉRER LA RÉPONSE

(5) LE SCRIPT PLACE LA RÉPONSE À L'ENDROIT PRÉVU DANS LE DOCUMENT

LES FORMATS DE DONNÉES

XML (eXtensible Markup Language)

Il permet de stocker les données dans un langage de balisage semblable au HTML (avec des balises personnalisées). Il est très pratique pour stocker de nombreuses données ayant besoin d'être formatées.

JSON (JavaScript Object Notation)

Il a pour particularité de segmenter les données dans un objet JavaScript, il est très avantageux pour de petits transferts de données segmentées et est de plus en plus utilisé.

Et aussi le format texte et HTML

UTILISATION AVEC JQUERY

La méthode .ajax()

```
2
3  $("#more_com").click(function(){
4
5      $.ajax({
6          url : 'more_com.php',
7          type : 'GET',
8          dataType : 'html',
9          success : function(code_html, statut){
10             $(code_html).appendTo("#commentaires"); // On passe code_html à jQuery() qui va nous créer
   l'arbre DOM !
11         },
12
13         error : function(resultat, statut, erreur){
14
15         },
16
17         complete : function(resultat, statut){
18
19         }
20
21     });
22
23 });
24
```

UTILISATION AVEC JAVASCRIPT

L'objet XMLHttpRequest

Méthode GET

```
// Requête HTTP
httpRequest = new XMLHttpRequest();


// Nom de la fonction qui va traiter la réponse du serveur

httpRequest.onreadystatechange = function () {
    // Instructions
};

// Lancement de la requête lorsque la réponse est reçue
// Méthodes .open() et .send()

httpRequest.open('GET', 'http://www.example.org/some.file', true);
httpRequest.send();
```

Méthode POST

```
httpRequest.open("POST", "cours.php", true);
// La méthode setRequestHeader() a 2 arguments : nom de l'entête et valeur attribuée
httpRequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
// Supposons que l'on veut envoyer au serveur les paramètres :
// 'cours' et 'niveau' qui ont comme valeurs respectives 'AJAX' et 'Moyen'
httpRequest.send("cours=AJAX&niveau=Moyen");
```

UTILISATION AVEC JAVASCRIPT

L'attribut readyState

```
6  httpRequest.onreadystatechange = function () {
7      if (httpRequest.readyState === 4) {
8          // Requête terminée et réponse prête (correspond à la valeur readyState = 4)
9      } else {
10         // Requête pas encore prête
11         // Les autres valeurs readyState :
12         // 0 = Requête non initialisée
13         // 1 = Requête en cours de chargement
14         // 2 = Requête reçue par le serveur
15         // 3 = Réponse en cours de traitement
16     }
17 };
```

UTILISATION AVEC JAVASCRIPT

L'attribut status

```
6   httpRequest.onreadystatechange = function () {
7       if (httpRequest.status === 200) {
8           // Le document a été trouvé à l'emplacement désigné et est accessible pour l'ouverture
9       } else {
10          // Il y a eu un problème avec la requête
11          // Exemples d'erreurs :
12          // 404 = Document introuvable
13          // 500 = Erreur interne du serveur
14          // 403 = Accès refusé
15      }
16  };
```

UTILISATION AVEC JAVASCRIPT

Exemple d'affichage des données

```
6  httpRequest.onreadystatechange = function () {
7      if (httpRequest.readyState === 4 && httpRequest.status === 200) {
8          // Tout s'est bien passé
9          document.getElementById("maDiv").innerHTML = httpRequest.responseText;
10     };
11 }
```