

HEIG-VD / INGÉNIERIE DES MÉDIAS / PROGRAMMATION WEB

INTRODUCTION AU JAVASCRIPT ASYNCHRONES

```
const paragraph = document.querySelector("p");
```

```
paragraph.textContent = "Sympa!";
```

```
alert("texte modifié");
```

```
paragraph.style.color = "red";
```



```
const paragraph = document.querySelector("p");
```

```
paragraph.textContent = "Sympa!";
```

```
alert("texte modifié");
```

```
paragraph.style.color = "red";
```



```
const paragraph = document.querySelector("p");
```

```
paragraph.textContent = "Sympa!";
```

```
alert("texte modifié");
```

```
paragraph.style.color = "red";
```



```
const paragraph = document.querySelector("p");
```

```
paragraph.textContent = "Sympa!";
```

```
alert("texte modifié");
```

```
paragraph.style.color = "red";
```



```
const paragraph = document.querySelector("p");
```

```
paragraph.textContent = "Sympa!";
```

```
alert("texte modifié");
```

```
paragraph.style.color = "red";
```





Le majeure partie du code écrit jusqu'à présent est synchrone:

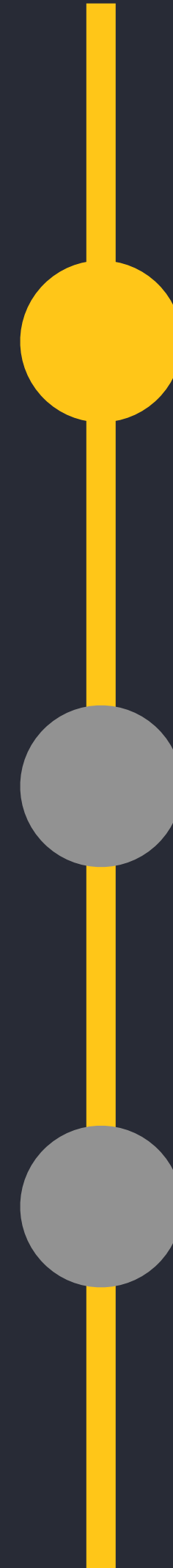
- Il est exécuté ligne par ligne
- Le moteur attend que la ligne précédente aie fini d'être exécutée avant de passer à la ligne suivante.

 **Les longues opérations bloquent l'exécution du code!**

```
const paragraph = document.querySelector("p");
```

```
setTimeout(() => {  
  paragraph.textContent = "Sympa!";  
}, 5000);
```

```
paragraph.style.color = "red";
```




```
const paragraph = document.querySelector("p");
```

```
setTimeout(() => {  
  paragraph.textContent = "Sympa!";  
}, 5000);
```

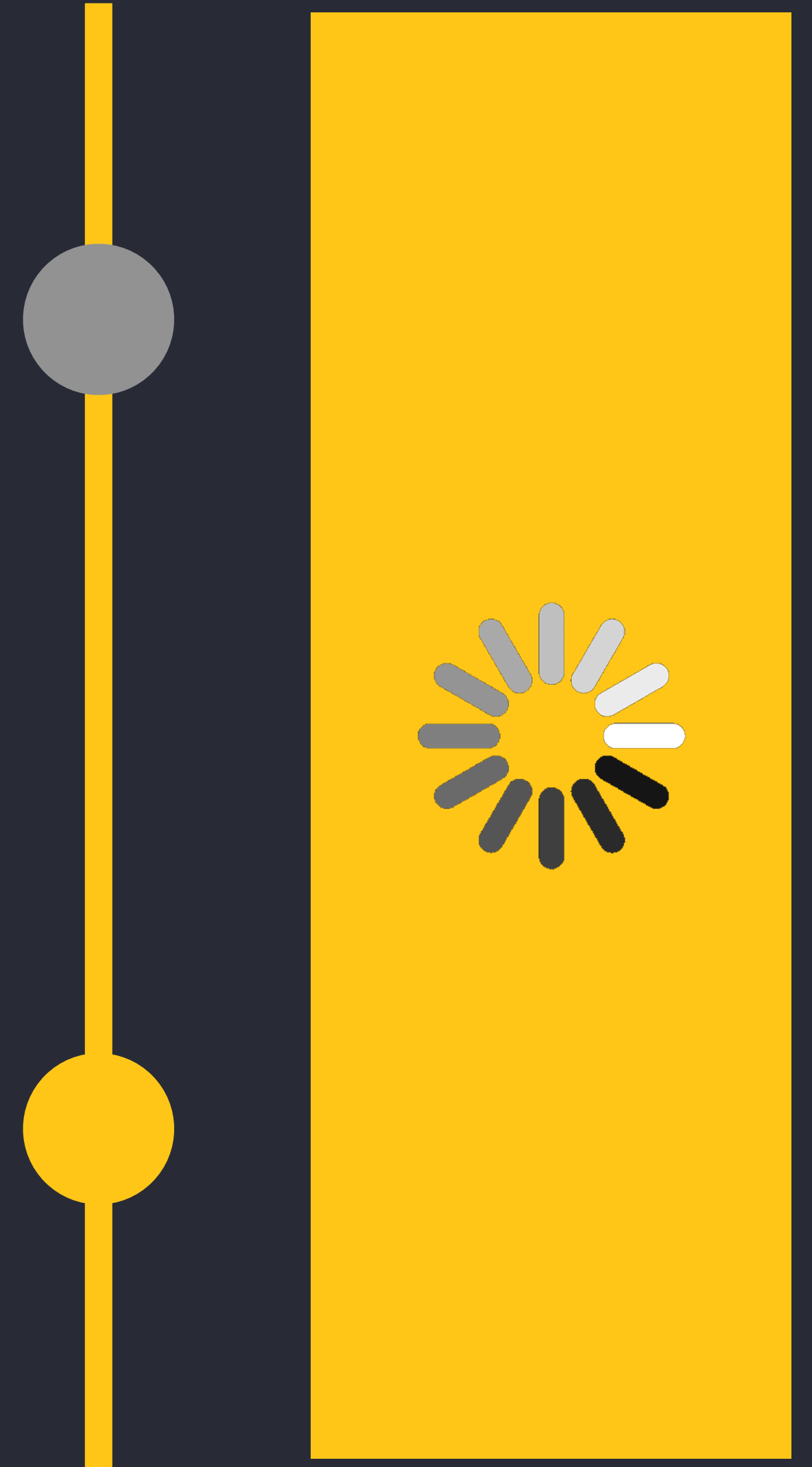
```
paragraph.style.color = "red";
```



```
const paragraph = document.querySelector("p");
```

```
setTimeout(() => {  
  paragraph.textContent = "Sympa!";  
}, 5000);
```

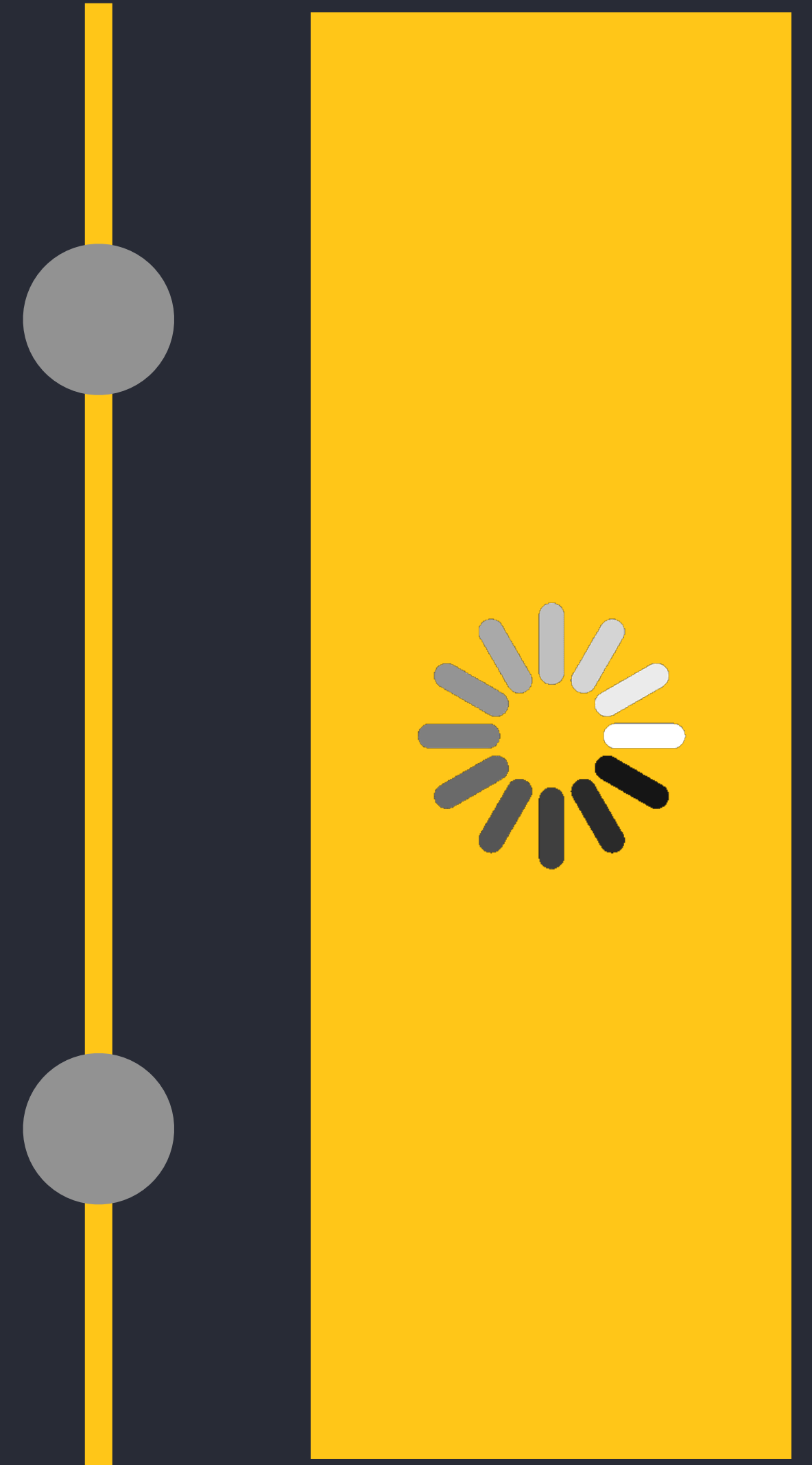
```
paragraph.style.color = "red";
```

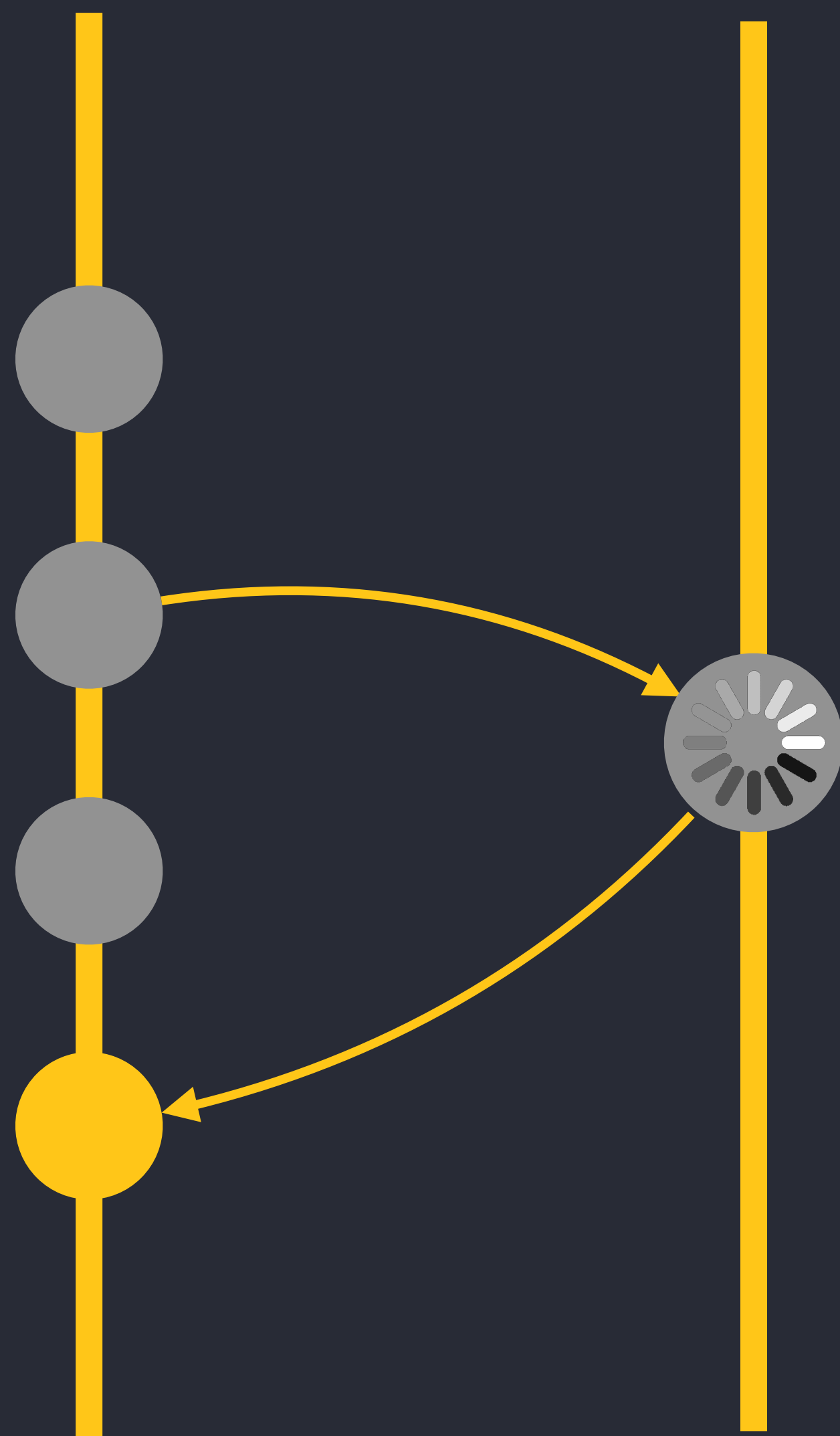


```
const paragraph = document.querySelector("p");
```

```
setTimeout(() => {  
    paragraph.textContent = "Sympa!";  
}, 5000);
```

```
paragraph.style.color = "red";
```





Du code asynchrone est exécuté une fois qu'une tâche qui tourne dans le “background” est terminée.



Le code asynchrone est non-bloquant: l'exécution du reste du code n'attend pas que la tâche asynchrone soit terminée.

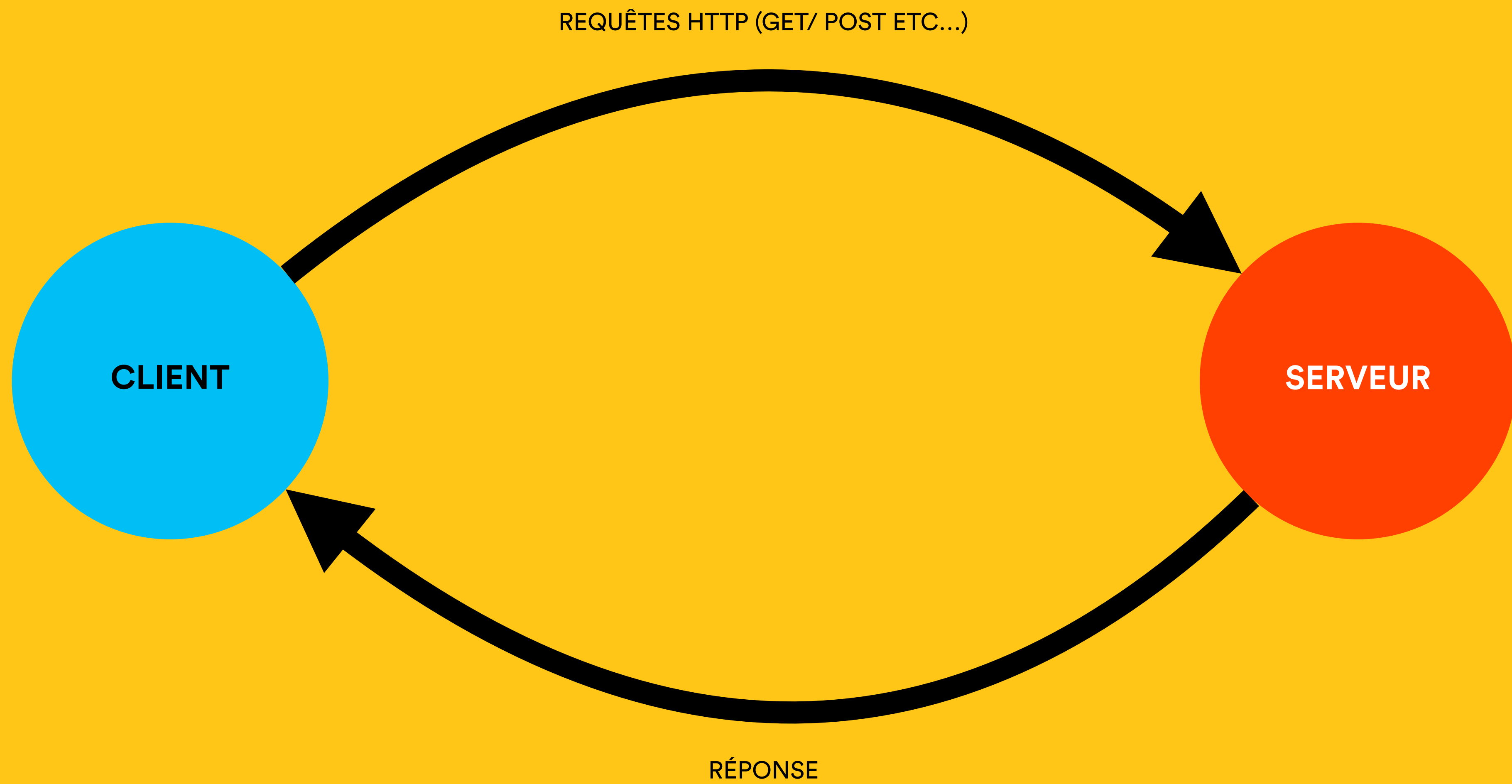
Bien que l'exemple précédent utilise un callback pour exécuter du code asynchrone, **callback \neq asynchrone.**

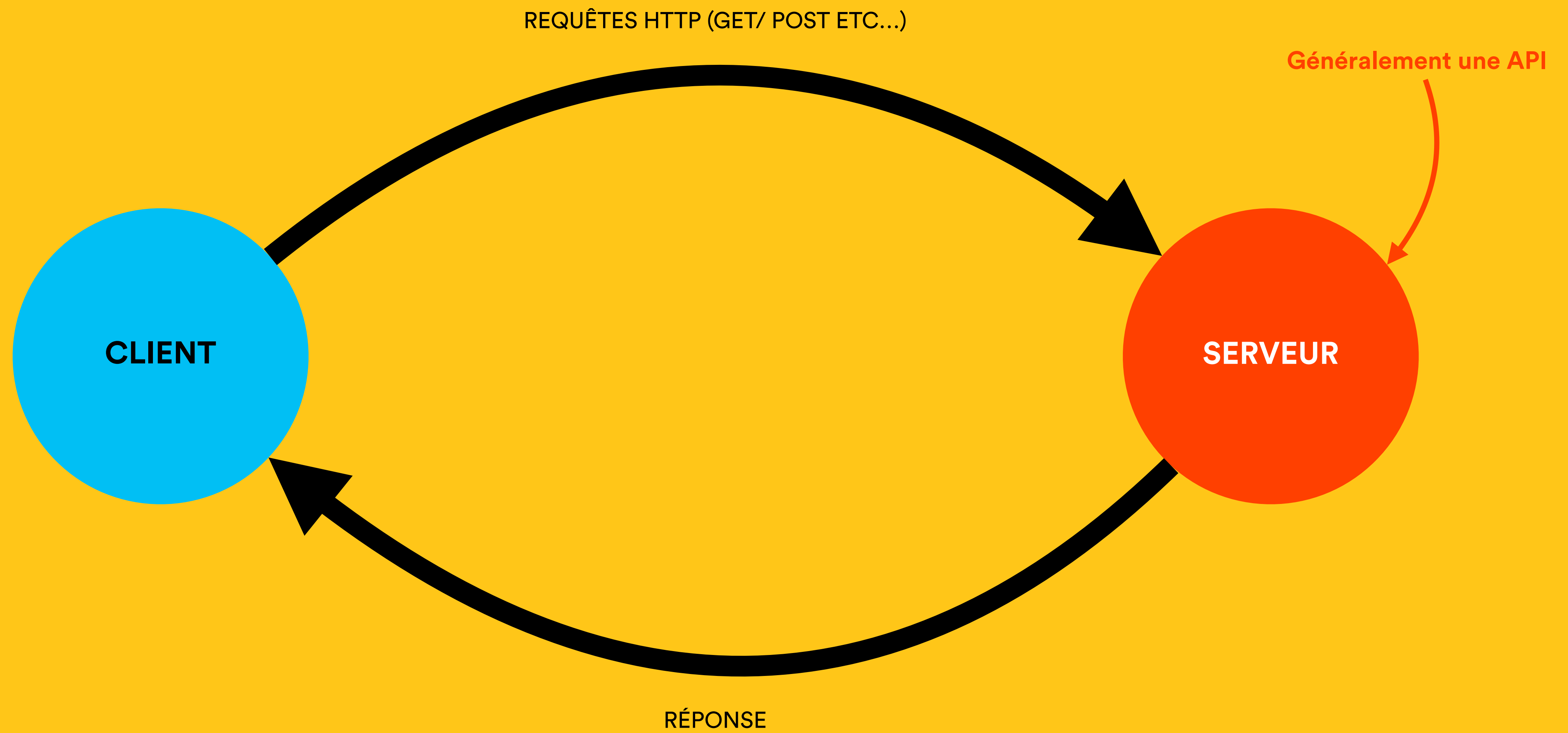




Asynchronous JavaScript and XML:

Nous permet de communiquer avec des serveurs distants de façon asynchrone. Avec des appels AJAX, nous pouvons donc requérir des données de serveurs web dynamiquement.





API

Application Programming Interface:

Un logiciel permettant d'être utilisé par un autre logiciel. Ceci permet à différentes applications de communiquer ensemble.

Il y a de nombreux types d'APIs. Certains sont "locaux": DOM API, Geolocation API.

D'autres sont "distants": Un logiciel qui reçoit des requêtes sur un serveur et renvoie les données demandées en réponse.

Vous créerez votre propre API distant dans le cours ArchiOWeb l'année prochaine.

**THERE'S AN
API FOR THAT!**

XMLHTTPREQUEST

```
const req = new XMLHttpRequest();  
req.open("GET", url);  
req.send();  
  
req.addEventListener("load", (e) => {  
    e.currentTarget.responseText;  
});
```

XMLHttpRequest() nous permet de faire des requêtes vers un serveur distant, sans avoir besoin de rafraîchir notre page.

On peut attendre la réponse en ajoutant un Event Listener sur l'objet de requête.

Lorsque la réponse arrive, nous pouvons la traiter dans le callback.


```
1 function hell(win) {
2   // for listener purpose
3   return function() {
4     loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
5       loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
6         loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
7           loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
8             loadLink(win, REMOTE_SRC+'/lib/underscore.min.js', function() {
9               loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
10                loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
11                  loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
12                    loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
13                      async.eachSeries(SCRIPTS, function(src, callback) {
14                        loadScript(win, BASE_URL+src, callback);
15                      });
16                    });
17                  });
18                });
19              });
20            });
21          });
22        });
23      });
24    });
25  };
}
```

