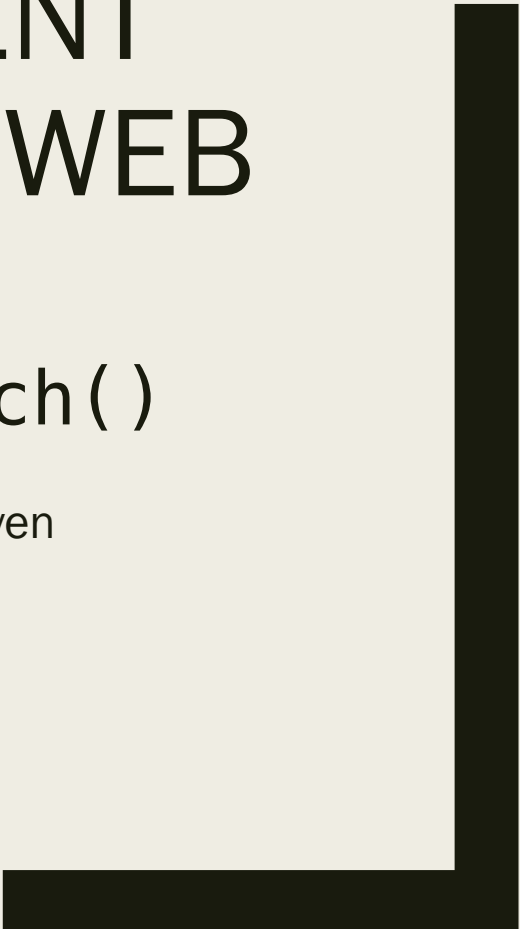




# DÉVELOPPEMENT D'APPLICATIONS WEB

## COURS 8: FONCTION `fetch()`

Enseignante: NGUYEN Thi Minh Tuyen



# Rappel: Fonction `fetch()` [1]

Nous avons parlé de la fonction `fetch()` dans le cours précédent:

- Fonction pour charger des ressources en JavaScript
- Exemple

```
fetch(pathToResource)
    .then(onResponse)
    .then(onResourceReady);
```

`onResponse`: Renvoie `response.text()` à partir de cette fonction pour obtenir la ressource sous forme de chaîne dans `onResourceReady`

`onResourceReady`: Obtient la ressource en tant que paramètre lorsqu'elle est prête

# Rappel: Fonction `fetch()` [2]

```
function onTextReady(text) {  
    // do something with text  
}  
function onResponse(response) {  
    return response.text();  
}  
fetch('images.txt')  
    .then(onResponse)  
    .then(onTextReady);
```

# Syntaxe

```
Promise<Response> fetch(entrée[, init]);
```

- **entrée** : Défini la ressource que vous voulez collecter. Cela peut être :
  - *Un USVString qui contient l'URL de la ressource à obtenir. Certains navigateurs acceptent blob: et data:.*
  - *Un objet Request.*
- **init** : facultatif, Un objet qui contient les paramètres de votre requête. Les options possibles sont :
  - *method* : *La méthode de la requête, par exemple GET ou POST.*
  - *headers* : *Les entêtes à ajouter à votre requête.*
  - *body* : *Le corps de votre requête.*
  - ....
- **Valeur retournée**: Une **Promise** qui se résoud avec un objet **Response**.
- Plus information, voir [ICI](#).

# Exemple

```
fetch(url,  
  {  
    method: 'GET',  
    headers: { 'Content-type':  
      'application/json; charset=UTF-8' ,  
      'Accept': 'application/json',  
      'Authorization': 'Bearer BQDpgu2U5YNMU-  
TbxLz7iloDm-  
LnFVqfcrbzdaFfQNb_HZrqiAazuWvbmwH4ZucLlkMByMZDI081N  
6UIwH_4LpxJC0hJexU4' }  
  })  
  .then(onResponse)  
  .then(onJsonReady);
```

# Question ?