Les classes

La programmation orientée objet

Boris Rose
Concepteur Logiciel

20 juillet 2024

Résumé

Nous verrons ici ce qu'est une classe, ses attributs, ses méthodes

0.1 Définition d'une classe

On peut dire qu'une classe est un exemple , elle a d'ailleurs un prototype. C'est un objet prototypé qui peut donner son prototype à d'autres objets que l'on appelle des instances de cette classe.

```
// on est dans le fichier home-container.js

//on définit la classe HomeContainer

class HomeContainer {

    constructor(onNavigate){
        this.onNavigate = onNavigate
    }

}
```

Explications le mot clé class permet de définir une classe en l'occurrence je l'appelle HomeContainer

la première chose que je mets dedans c'est une fonction particulière que l'on appelle le constructeur c'est grâce à cette fonction que je vais pouvoir instancier la classe c'est-à-dire transmettre son prototype (ses attributs et ses méthodes) à d'autres objects

Dans mon exemple je dis que lors de l'appelle du constructeur,autrement dit lors de l'instanciation de classe, ou encore quand je vais vouloir transmettre son prototype à un objet, je veux qu'on lui passe un argument qui va être la valeur donné au paramètre onNavigate.

Une fois que la valeur est passée au constructeur elle va l'associer à l'instance avec :

```
this.onNavigate = onNavigate
```

this fait référence à l'environnement présent à savoir la classe. Dit autrement this.onNavigate va permettre de créer un attribut à la classe et donc de pouvoir l'utilisateur n'importe où dans l'environnement que représente la classe autrement dit dans d'autres méthodes de cette classe.

```
// on est dans le fichier home-container.js

//on définit la classe HomeContainer

class HomeContainer {
```

```
    constructor(onNavigate){
        this.onNavigate = onNavigate
        const homeMainButton = document.getElementById("
home-main-button")
        homeMainButton.addEventListener("click", this.
onClick)
    }

    static showClassName(){
        /*
            ceci est une méthode de classe, elle est appelé
e sur la classe et non pas sur l'instance
        */
        console.log("Home Container")
    }

    onClick(){
       // ceci est une méthode d'instance elle est appelée
sur l'instance autrement dit sur l'objet qui est l'instance
 de cette classe

       onNavigate("#login")
    }


}

export default HomeContainer
```

0.2 Méthodes de classe et Méthodes d'instance

J'ai ajouté de nombreuses choses. Tout d'abord dans le constructeur en plus de définir un attribut onNavigate, j'ai aussi défini une variable homeMainButton (camel case h—M—B—) qui contient l'élément HTML qui a pour id "home-main-button"

Une fois que la variable homeMainButton est définie comme un élément HTML je peux brancher un écouteur d'événement sur elle qui écoute l'évé-

nement click.

Branche cela veut dire appeler sur ladite variable la méthode des éléments HTML , addEventListener()

Cette méthode prend deux arguments le premier est le nom de l'événement que l'on souhaite écouter. En l'occurence c'est le click sur le bouton et en second argument la fonction qui va gérer l'événement autrement dit qui va s'exécuter au moment où l'événement survient. On appelle cette fonction un gestionnaire d'événement et vu qu'elle est passée comme argument à une fonction addEventListener() on l'appelle une callback (une fonction de rappel).

Donc le gestionnaire est une callback et une fonction de première classe.

Or le gestionnaire que j'ai passé à la méthode addEventListener est une méthode d'instance de ma classe HomeContainer

Donc je peux y faire référence avec le this.onClick

Attention ! Comme vous pouvez le voir je n'ai pas exécuter la fonction en la passant en argument j'ai juste passé sa définition onClick. Car c'est seulement au moment de la survenance de l'événement click que je veux qu'elle soit exécutée.

0.3 Appel du constructeur de la classe HomeContainer

```
/*

On est dans le fichier router.js

Ne pas oublier d'appeler la fonction qui est tout en bas en
   export default dans le fichier main.js qui est le point d'
   entrée de votre application

Ppremière chose à mettre les imports

*/

import homeView from "./src/js/views/home-view/home-view.js"
import HomeContainer from "./src/js/containers/home-container/
   home-container.js"
```

```
window.onNavigate = (h) => {
    navigateToPage(h)
}



/*

    on a vu l'événement click des boutons et bien l'événement
   de modification de l'url contenu dans la barre d'adresse du
    navigateur s'appelle popstate

    Je veux qu'au moment où l'url est modifiée que cela exécute
    le gestionnaire autrement dit la fonction fléchée => que j
   'ai assigné comme valeur à la propriété onpopstate de l'
   objet window
*/


window.onpopstate = () => {
    navigateToPage(window.location.hash)
}




// je définis une fonction qui réinitialise ce qui est contenu
   entre les balises de l'élément HTML ayant comme id root

function resetRoot(){
    const root = document.getElementById("root")
    root.innerHTML = ""
    return root
}



// je définis une fonction qui va mettre à jour dans le
   navigateur l'historique de navigation en modifiant
```

```
   visiblement l'url qui se trouve dans la barre d'adresse

function updateHistoryState(h){
    window.history.pushState({}, "", window.location.pathname +
    h)
}


// je définis une fonction qui modifie l'interface utilisateur
   en fonction de h
function navigateToPage (h) {
    updateHistoryState(h)
    const root = resetRoot()
    switch(h){
        case "":
            // ce cas représente l'état du hash au démarrage de
    l'application
            // vu qu'on est sur la page racine ne window.
   location.hash = ""

            root.innerHTML += homeView()

            /*
                ci-dessus
                création de l'interface utilisateur autrement
   dit apparaît à l'écran la valeur retournée par la fonction
   homeView() et interprétée en HTML par innerHTML

                Une fois que l'interface est à l'écran , pré
   sente sur le rendu du Navigateur

                je veux que logique métier de la page soit impl
   émentée autrement dit je veux que les choses qui sont pré
   sentes sur l'interface fonctionnent

                Or la logique métier de la page est géré dans
   la classe HomeContainer

                Je vais donc instancier cette classe : le mot
   clé new suivi de l'appel au constructeur de la classe
   suffit à faire cela
```

```
            */

            new HomeContainer(window.onNavigate)
            break;
        case "#login":
            break;
        default:
            break;

    }

}

export default function(){
    navigateToPage(window.location.hash)
}
```