TP 7: Classes

Bases de la POO: classes JavaScript

Programmation Orientée Objet: classes, instances et `this`

Une classe est un modèle d'objet. Elle peut être instanciée, c'est à dire qu'on crée un objet (appelé *instance*) selon ce modèle.

La modèle d'une classe consiste à assurer que chaque objet instance de cette classe aura les mêmes:

- propriétés; (*cf chapitre sur les types avancés*)
- et méthodes: des fonctions qui s'appliquent à une instance donnée.

À noter que:

- chaque instance d'une classe aura les mêmes propriétés, mais la valeur de celles-ci pourra être différente pour chaque instance;
- chaque instance d'une classe aura les mêmes méthodes, mais l'exécution de la fonction correspondante ne s'appliquera qu'à l'instance sur laquelle elle aura été appelée.

Comme nous allons le voir dans la suite du cours, les classes sont très utilisées pour manipuler la structure de pages Web. Notamment pour intégrer plusieurs instances d'un même composant sur une même page.

Comment instancier une classe en JavaScript/ES6

En guise d'exemple, supposons qu'on veuille intégrer un composant permettant d'afficher une galerie d'images sur notre page Web.

Supposons que ce composant soit défini par une classe nommée `Galerie`.

Comme pour toute classe, on peut instancier une `Galerie` en appelant son constructeur avec le mot clé `new`. Le constructeur de cette classe prend un paramètre: `conteneur`, une portion de la page Web dans lequel la galerie s'intègrera.

Enfin, supposons que la classe `Galerie` fournisse des méthodes qui seront rattachées à chaque instance de cette classe:

- `ajouterImage()` permet de spécifier l'URL d'une image qui sera à afficher dans cette galerie,
- et `regenerer()` permet de mettre à jour l'affichage de la galerie, après y avoir ajouté des images.

Voici un exemple d'instanciation de cette classe:

```
// supposons que conteneur référence un <div> de la page
var maGalerie = new Galerie(conteneur);
maGalerie.ajouterImage('img7.jpg');
maGalerie.regenerer();
```

Le mot clé **new** permet d'instancier notre classe, et donc d'exécuter son constructeur en fournissant une valeur pour le paramètre `conteneur`. Comme pour une fonction, l'appel au constructeur retourne l'instance de `Galerie` fraichement créée.

Comment définir une classe en JavaScript/ES6

Afin de permettre l'instanciation de la classe `Galerie`, le créateur du composant a du la définir de la manière suivamte:

```
class Galerie {

  // définition du constructeur de la classe Galerie
  constructor(conteneur) {
    this.conteneur = conteneur;
    this.urlImages = [];
  }

  // cette méthode permet d'ajouter une image à cette galerie
  ajouterImage(url) {
    this.urlImages.push(url);
  }

  // cette méthode permet de générer et d'afficher cette galerie dans la page
  regenerer() {
    var html = '';
    // génération des éléments <img> dans le conteneur
    for (var i = 0; i < this.urlImages.length; i++) {
      html = html + '<img src="' + this.urlImages[i] + '" class="hidden">;';
    }
    this.conteneur.innerHTML == html;
  }

}
```

À noter:

- le paramètre `conteneur` du constructeur de la classe a été affecté comme propriétés d'un certain objet `this`. (nous allons expliquer ça plus bas)

Usage de `this`

Quand on mentionne `this` dans la définition d'une méthode, ce mot clé représente l'instance depuis laquelle la méthode a été appelée.

Par exemple:

```
class Article {
  constructor(titre) {
    this.titre = titre;
  }
```

```
  getTitre() {
    return this.titre; // this === article1 ou article2, dans notre exemple
  }
}

var article1 = new Article('Trump élu président');
var article2 = new Article('Macron se présente');

article1.getTitre(); // => retourne 'Trump élu président'
article2.getTitre(); // => retourne 'Macron se présente'
```

À noter qu'en JavaScript, `this` est en fait utilisable depuis toute fonction, qu'elle soit ou pas définie dans une classe. Il faut retenir que l'usage de classes permet à l'interpréteur JavaScript d'affecter automatiquement à `this` l'instance sur laquelle s'exécute chaque méthode.

Exercice: Création de classe

--> http://marijnhaverbeke.nl/talks/es6_falsyvalues2015/exercises/#Point