MODULE FRONT-END

JAVASCRIPT –SUR BASE DU TABLEAU DE POST IT

Objectifs du cours

Compréhension et maitrise du web client

Réalisation d’un jeu PacMan

Avantages de la programmation orientée objet

L'orienté objet remplace le procédural dans les grands programmes car il présente de multiples avantages:

-Facilité de compréhension (Permet de regrouper toutes informations sur un *objet* dans le code : si mon programme gère des voitures, l'objet du même nom contiendra la marque de celle-ci, sa vitesse ainsi que sa couleur, etc...)

- L'encapsulation (la programmation orientée objet permet la protection de l'information contenue dans un objet. Les données ne sont pas accessibles directement par l'utilisateur, celui-ci devant passer par les méthodes publiques.).

-La modularité du code (On peut généralement récupérer 80 % du code d'un projet pour le réutiliser sur un projet similaire contrairement à la programmation procédurale. De plus les objets permettent d'éviter la création de code redondant, permettant donc un gain de temps et donc d'argent).

Les classes

Déclaration d'une classe : class maClasse {}

Membre d'une classe : monMembre = 5;

Méthode d'une classe : maFonction() {}

Getter : get monMembre() {}

Setter : set monMembre(i) {}

Constructeur : constructor() {}

Gestion du moimême : this

Les classes- exemple

```
class Voiture
{
   Marque;
   Modele;
   Couleur;
   constructor(marque, modele, couleur) {
      this.Marque=marque;
      this.Modele=modele;
      this.Couleur=couleur;
   }
     roule() {
        return "Vroum vroum";
     }
     get Marque() {
        return "Voiture de marque "+this.Marque;
       }
     set Marque(laMarque) {
         this.Marque=laMarque;
     }
     get Modele() {
        return "Voiture de modele "+this.Modele;
       }
     set Modele(leModele) {
         this.Modele = leModele
     }
}
```

Principes du développement Orienté Objet

L'isolation :

Par principe on n'utilisera pas de variable globale directement dans une classe. On la passera en paramètre d'une fonction. Cela permet de réutiliser la classe dans un futur programme et d'éviter les bugs dans celui en cours.

Une bonne habitude est également de créer un fichier par classe, ce n'est pas obligatoire mais recommandé.

L'instanciation :

Une classe n'est qu'un modèle contenant les variables (membres) et les fonctions (méthodes). Une classe servira de coquille pour créer plusieurs objets qui se ressemblent.

Ex d'instanciation : var personne1 = new Personne('Bob');

Principes de l'Orienté Objet- suite

L'héritage :

Vous pouvez faire hériter une classe d'une classe parent, exemple une classe chien enfant d'une classe animal.

Une classe enfant hérite par principe de l'ensemble des membres et méthodes de la classe parent.

Une classe enfant n'a pas besoin de constructeur, la classe parent construit l'objet.

Une classe enfant peut ajouter de nouveaux membres et de nouvelles méthodes à la classe parent.

Le polymorphisme :

Lorsqu'une classe redéfini un membre ou une méthode de la classe parent, on parle de polymorphisme pour ce faire le membre ou la méthode modifié doit avoir le même nom que dans la classe parent.

Pour exécuter une méthode de la classe parent, vous pouvez utiliser le mot clé super.methodeParent(…)

Pour redéfinir un constructeur dans la classe enfant vous devez absolument appeler le constructeur parent via la méthode super() à l'intérieur de votre constructeur

Méthodologie Kaban avec GitHub

Sur GitHub, vous pouvez gérer de la gestion de projet Agile grace à la méthode Kaban.

Le principe : une todo-list de post-it à placer de a faire, vers en-cours puis fait.

Pour réaliser une gestion de projet Kaban sur votre dépôt cliquez sur l'onglet projet puis new project.

Donnez un nom, une description et comme template, je vous propose basic Kaban.

Pour le projet post-it, je vous propose les étapes suivantes :