Programmation Web L3 Miage – Info M2 IMDS

Exercices sur le cours 1

Étudiant

Un étudiant est un objet avec la propriété `nom` qui contient son nom et la propriété `prénom` qui contient son prénom.

Écrivez la fonction `etudiantNomPrenom(nom, prenom)` qui renvoie un étudiant de nom le nom passé en paramètre en majuscules et de prénom celui passé en paramètre avec la première lettre en majuscule et les suivantes en minuscules (les méthodes `toLowerCase()` et `toUpperCase()` des chaînes de caractères renvoie la chaîne sur laquelle elles sont appelées respectivement en minuscules et en majuscules).

Écrivez la fonction `etudiantFormaté(e)` qui renvoie la chaîne de caractères `NOM : nom étudiant Prénom : prénom étudiant` correspondant à l'étudiant e (p.ex. `NOM : TALUS Prénom : Jean`).

Écrivez la procédure de test `testEtudiant` qui déclare les variables `nom` valant `talus` et `prenom` valant `jean`, crée directement un objet étudiant à partir de ces deux variable et l'affiche formaté sur la console, puis crée un étudiant avec `etudiantNomPrenom` et l'affiche formaté sur la console.

Compteur

On veut créer une classe pour représenter un compteur.

Écrivez la « classe » `Compteur` : son constructeur prend la valeur entière du compteur en paramètre (0 par défaut) et sa valeur limite (100 par défaut) qu'ils stocke dans l'objet (dans `_valeur` et `_limite`), sa méthode `valeur()` renvoie la valeur, sa méthode `limiteAtteinte()` renvoie vrai si la limite du compteur est atteinte, faux sinon et sa méthode `augmente()` augmente la valeur de 1 si la limite n'est pas atteinte.

Écrivez la procédure de test `testCompteur1` qui crée un compteur de valeur 5 et de limite 12 et qui dix fois l'augmente et affiche sa valeur dans la console. Testez.

Ajoutez dans la classe la fonction statique `aPartirObjet(objc)` qui crée et renvoie un compteur à partir de l'objet `objc` qui contient une propriété valeur contenant la valeur du compteur et une propriété limite qui contient la limite du compteur.

Écrivez la procédure de test `testCompteur2` qui fait comme `testCompteur1` mais en utilisant `aPartirObjet` pour créer le compteur.

Solution étudiant

```
function etudiantNomPrenom(nom, prenom)  {
  return {
    nom : nom.toUpperCase(),
    prenom: prenom[0].toUpperCase() +
prenom.slice(1).toLowerCase()
  };
}

function etudiantFormaté(e) {
  return `NOM : ${e.nom} Prénom : ${e.prenom}`
}

function testEtudiant() {
  const nom = 'talus';
  const prenom = 'jean';
  const e1 = { nom, prenom };
  console.log(etudiantFormaté(e1));
  const e2 = etudiantNomPrenom(nom, prenom);
  console.log(etudiantFormaté(e2));
}

testEtudiant();
```

Solution compteur

```
class Compteur {
  constructor(valeur = 0, limite = 100) {
    this._valeur = valeur;
    this._limite = limite;
  }

  valeur() {
    return this._valeur;
  }

  limiteAtteinte() {
    return this._valeur === this._limite;
  }

  augmente() {
    if (!this.limiteAtteinte()) {
      ++this._valeur;
    }
  }

  static aPartirObjet(objc) {
    return new Compteur(objc.valeur, objc.limite);
  }
}

function etudiantFormaté(e) {
  return `NOM : ${e.nom} Prénom : ${e.prenom}`
}

function testCompteur1() {
  const c = new Compteur(5, 12);
  for (let i=0; i<10; ++i) {
    c.augmente();
    console.log(c.valeur());
  }
}

function testCompteur2() {
  const c = Compteur.aPartirObjet({ valeur:5, limite: 12});
  for (let i=0; i<10; ++i) {
    c.augmente();
    console.log(c.valeur());
  }
}

testCompteur1();

testCompteur2();
```