

TP et Labo de la semaine 9 + solutions

Séance de démonstration #9, IFT1015

- 1) Un palindrome est un texte, tel que "zeus a ete a suiez", qui peut se lire de la même façon de gauche à droite et de droite à gauche. Codez une fonction `palindrome(texte)` qui retourne vrai si et seulement si le texte en paramètre est un palindrome. Utilisez la méthode `charAt`.

Solution :

```
var palindrome = function (texte) {  
  
    for (var i=0; i<texte.length/2; i++) {  
        if (texte.charAt(i) !== texte.charAt(texte.length-i-1)) {  
            return false;  
        }  
    }  
  
    return true;  
};
```

- 3) Sachant que dans le jeu de caractère Unicode, les caractères 0 à 9 ont les codes 48 à 57, codez une fonction `valeurNum(texte)` qui prend en paramètre un texte contenant des chiffres décimaux, et qui retourne comme résultat le nombre qui correspond à cette séquence de chiffres.
C'est-à-dire l'appel `valeurNum("123")` doit retourner 123. Évitez d'utiliser les conversions automatiques des opérateurs arithmétiques. Utilisez la méthode `charCodeAt`.

Solution :

```
var valeurNum = function (texte) {  
  
    var n = 0;  
  
    for (var i=0; i<texte.length; i++) {  
        n = n*10 + texte.charCodeAt(i) - 48;  
    }  
  
    return n;  
};
```

4) Codez une fonction `majuscule(texte)` qui prend en paramètre un texte et qui retourne un texte avec les mêmes caractères, sauf que le premier caractère, si c'est une lettre minuscule, sera remplacée par la lettre majuscule

correspondante. Par exemple :

```
majuscule("") => ""
majuscule("x") => "X"
majuscule("allo") => "Allo"
majuscule("Bonjour") => "Bonjour"
```

Utilisez le truc suivant :

```
String.fromCharCode(n)
```

retourne un texte contenant un seul caractère dont le code dans le jeu de caractère Unicode est `n`. Par exemple,

```
String.fromCharCode(65) => "A"
```

Solution :

```
var majuscule = function (texte) {
  if (texte.length > 0) {
    var c = texte.charCodeAt(0);
    if (c >= 97 && c <= 122) { // entre a et z ?
      return String.fromCharCode(c-32) + texte.slice(1, texte.length);
    }
  }
  return texte;
};
```

Note : la fonction `"...".charCodeAt()` retourne `NaN` si l'index passé en paramètre est invalide (plus petit que 0 ou plus grand ou égal à la longueur de la String). On peut donc simplifier avec:

```
var majuscule = function (texte) {
  var c = texte.charCodeAt(0);
  if (c >= 97 && c <= 122) { // Faux si c = NaN
    return String.fromCharCode(c-32) + texte.slice(1, texte.length);
  }
  return texte;
};
```

5) Codez un programme qui décompose un texte en mots et affiche combien de fois apparaît chaque mot vu dans le texte.

Par exemple,

```
analyser("il a un chat noir et un chat blanc");
```

devrait imprimer :

```
il 1
a 1
un 2
chat 2
noir 1
et 1
blanc 1
```

Solution :

```
var lettre = function (car) {
    return (car >= "a" && car <= "z") ||
           (car >= "A" && car <= "Z");
};

var decouperEnMots = function (texte) {
    var resultat = [];
    var debut = 0;
    while (debut < texte.length) {
        if (lettre(texte.charAt(debut))) {
            var i = debut+1;
            while (i < texte.length && lettre(texte.charAt(i))) {
                i++;
            }
            resultat.push(texte.slice(debut, i));
            debut = i+1;
        } else {
            debut++;
        }
    }
    return resultat;
};

var frequenceMots = function (texte) {
    var mots = decouperEnMots(texte);
    var vus = []; // mots vus à date
```

```
var resultat = []; // chaque mot vu avec sa fréquence
for (var i=0; i<mots.length; i++) {
    var pos = vus.indexOf(mots[i]);
    if (pos >= 0) {
        resultat[pos].freq++;
    } else {
        vus.push(mots[i]);
        resultat.push({mot: mots[i], freq: 1});
    }
}
return resultat;
};

var imprimerFreq = function (freqMots) {
    for (var i=0; i<freqMots.length; i++) {
        print(freqMots[i].mot + " " + freqMots[i].freq);
    }
};

var analyser = function (texte) {
    imprimerFreq(frequenceMots(texte));
};

analyser("il a un chat noir et un chat blanc");
```