

Prise en main de javascript

Benoit Favre

généré le 11 février 2021

Exercice 1 : tableaux

1. Écrire une fonction javascript `max(t)` qui renvoie le plus grand élément du tableau `t` passé en paramètre.

```
function max(t) {  
  let max_element = t[0];  
  for (let i = 1; i < t.length; i++) {  
    if (t[i] > max_element)  
      max_element = t[i];  
  }  
  return max_element;  
}
```

2. Écrire une fonction javascript `argmax(t)` qui renvoie l'indice du plus grand élément du tableau `t` passé en paramètre.

```
function argmax(t) {  
  let max_element = t[0];  
  let argmax_index = 0;  
  for (let i = 1; i < t.length; i++) {  
    if (t[i] > max_element) {  
      argmax_index = i;  
      max_element = t[i];  
    }  
  }  
  return argmax_index;  
}
```

3. Écrire une fonction javascript `median(t)` qui retourne la valeur médiane du tableau de nombres passé en paramètre.

```
function median(t) {  
  t.sort(function(a, b) {return b - a});  
  let middle = Math.floor(a.length / 2);  
  if (a.length % 2 == 0)  
    return (a[middle] + a[middle + 1]) / 2;  
  else  
    return a[middle];  
}
```

4. Écrire une fonction javascript `search(t, x)` qui renvoie l'indice de la valeur `x` dans le tableau trié `t` en utilisant la recherche par dichotomie.

```
function search(t, x) {
  let start = 0, end = t.length;
  while (start < end) {
    let middle = Math.floor((start + end) / 2);
    let value = t[middle];
    if (value < x) start = middle;
    else if (value > x) end = middle;
    else return middle;
  }
  return -1;
}
```

Exercice 2 : dictionnaires

1. Écrire une fonction javascript `sum_of_values(d)` qui renvoie la somme des valeurs du dictionnaire passé en paramètre.

```
function sum_of_values(d) {
  let sum = 0;
  for (let key in d) {
    sum += d[key];
  }
  return sum;
}
```

2. Écrire une fonction javascript `argmin(d)` qui retourne la clé associée à la plus petite valeur du dictionnaire passé en paramètre.

```
function argmin(d) {
  let min_value = null;
  let min_key = null;
  for (let key in d) {
    if (min_key == null || d[key] < min_value) {
      min_value = d[key];
      min_key = key;
    }
  }
  return min_key;
}
```

3. Écrire une fonction javascript `union(d1, d2)` qui renvoie un nouveau dictionnaire contenant tous les éléments de `d1` et de `d2`. Si une clé appartient aux deux dictionnaires, la valeur associée devra être un tableau contenant les deux valeurs.

```
function union(d1, d2) {
  let result = {};
  for (let key in d1) {
```

```

    if (key in d2) result[key] = [d1[key], d2[key]];
    else result[key] = d1[key];
  }
  for (let key in d2) {
    if (! (key in d1)) result[key] = d2[key];
  }
  return result;
}

```

4. Écrire une fonction javascript `swap(d)` qui renverse un dictionnaire dont le rôle des clés et des valeurs est renversé par rapport à celui passé en paramètre, c'est à dire que les clés deviennent les valeurs et vice-versa. Si une valeur apparaît plusieurs fois, la fonction devra afficher un message d'erreur et retourner `null`.

```

function swap(d) {
  let result = {};
  for (let key in d) {
    let value = d[key];
    if (value in result) {
      console.log("erreur");
      return null;
    } else {
      result[value] = key;
    }
  }
  return result;
}

```

Exercice 3 : chaînes de caractères

Soit une chaîne de caractères `s` contenant un texte représentant des données météo :

```

1901  1 -3
1901  2 -2
1901  3 7
...
2000 11 13
2000 12 2

```

où les lignes sont séparées par le retour chariot “\n” et les valeurs à l'intérieur d'une ligne sont séparées par des tabulations “\t”.

Écrire un programme javascript qui analyse le contenu de cette chaîne, et génère un tableau contenant des objets avec les champs `year`, `month` et `temperature` pour représenter les valeurs se trouvant sur chaque ligne.

```

[
  {year: 1901, month: 1, temperature: -3},
  {year: 1901, month: 2, temperature: -2},
  {year: 1901, month: 3, temperature: 7},
  ...
]

```

```

let data = '1901\t1\t-3\n1901\t2\t-2\n1901\t3\t7';
let table = [];
for (let line of data.split('\n')) {
  let row = line.split('\t');
  table.push({
    year: parseInt(row[0]),
    month: parseInt(row[1]),
    temperature: parseInt(row[2])
  });
}
console.log(table);

```

Exercice 4 : génération de HTML

Écrire une fonction `to_html_list(t)` qui génère une chaîne de caractère contenant une liste UL à partir d'un tableau javascript passé en paramètre.

```

function to_html_list(t) {
  let result = '<ul>';
  for (let element of t) {
    result += '<li>' + element + '</li>';
  }
  result += '</ul>';
  return result;
}

```