



# ALGORITHMIQUE

# TABLE DES MATIÈRES

1.1 - Recherche dans un tableau.....	3
1.2 - Parcours tableau lentement.....	5
1.2.1 - Écrans.....	5
1.2.2 - Codes.....	6
1.3 - Partitionnement temporisé.....	9
1.3.1 - Écrans.....	9
1.3.2 - Codes.....	11

**1.1 - RECHERCHE DANS UN TABLEAU****Rechercher Valeur dans TO - Valeurs non uniques**Valeur recherchée dans [1,3,5,7,9,3,11] ? 

1	3	5	7	9	3	11
---	---	---	---	---	---	----

**Rechercher Valeur dans TO - Valeurs non uniques**Valeur recherchée dans [1,3,5,7,9,3,11] ? 

1	3	5	7	9	3	11
---	---	---	---	---	---	----

Position dans le tableau : 2 ! 6 !

**Rechercher Valeur dans TO - Valeurs non uniques**Valeur recherchée dans [1,3,5,7,9,3,11] ? 

1	3	5	7	9	3	11
---	---	---	---	---	---	----

Position dans le tableau : 3 !

**Rechercher Valeur dans TO - Valeurs non uniques**

Valeur recherchée dans [1,3,5,7,9,3,11] ?

55

Go

1	3	5	7	9	3	11
---	---	---	---	---	---	----

Position dans le tableau : Introuvable !

## 1.2 - PARCOURS TABLEAU LENTEMENT

### 1.2.1 - Écrans

#### **ParcoursTableauLentement**

2	1	4	3	9	6	5	7	8
---	---	---	---	---	---	---	---	---

Parcourir avec setTimeout

Parcourir avec setInterval

7

### 1.2.2 - Codes

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link href="../../css/jsAlgo.css" rel="stylesheet" />
    <title>ParcoursTableauLentement.html</title>
  </head>

  <body>

    <article id="article">
      <h1>ParcoursTableauLentement</h1>

      <table>
        <tr id="ligneTableau">
        </tr>
      </table>

      <p>
        <input type="button" value="Parcourir avec
setTimeout" id="btParcourirSetTimeout" />
      </p>
      <p>
        <input type="button" value="Parcourir avec
setInterval" id="btParcourirSetInterval" />
      </p>
      <p>
        <label id="lblMessage"></label>
      </p>
    </article>

    <script src="../../js/ParcoursTableauLentement.js"></script>
  </body>
</html>
```

```

/*
 * ParcoursTableauLentement.js
 */
var t = new Array(2, 1, 4, 3, 9, 6, 5, 7, 8);
var ligneTableau;
var btParcourirSetTimeout;
var btParcourirSetInterval;
var lblMessage;
var passages;
var oInterval;

// -----
function init() {
    ligneTableau = document.getElementById("ligneTableau");
    lblMessage = document.getElementById("lblMessage");

    btParcourirSetTimeout =
document.getElementById("btParcourirSetTimeout");
    btParcourirSetInterval =
document.getElementById("btParcourirSetInterval");

    btParcourirSetTimeout.onclick = parcourirSetTimeout;
    btParcourirSetInterval.onclick = parcourirSetInterval;
    afficher();
} /// init

// -----
function afficher() {
    var parent = ligneTableau;
    for (var i = 0; i < t.length; i++) {
        var td = document.createElement("td");
        td.setAttribute("id", "td" + i);
        parent.appendChild(td);
        td.innerHTML = t[i];
    }
} /// afficher

// -----
function parcourirSetInterval() {
    // blanchiment
    for (var k = 0; k < t.length; k++) {
        document.getElementById("td" + k).style.backgroundColor =
"white";
    }
    passages = 0;
    oInterval = setInterval(avancerSetInterval, 1000);
} /// parcourirSetInterval

// -----
function parcourirSetTimeout() {
    // blanchiment
    for (var k = 0; k < t.length; k++) {

```

```

        document.getElementById("td" + k).style.backgroundColor =
"white";
    }

    var i = 0;
    while (i < t.length) {
        console.log("i dans parcourir : " + i);
        i++;
        // setTimeout("nomDeFonction([param1, param2, ...])",
intervalle).
        setTimeout("avancerSetTimeout(" + i + ")", 1000 * i);
    } // while
} // parcourirSetTimeout

// -----
function avancerSetTimeout(i) {
    // blanchiment
    for (var k = 0; k < t.length; k++) {
        document.getElementById("td" + k).style.backgroundColor =
"white";
    }
    // colorisation
    console.log("i dans avancer : " + i);
    var cellule = "td" + i;
    console.log(cellule);
    if (i < t.length) {
        document.getElementById(cellule).style.backgroundColor =
"red";
    }
    console.log(i);
    lblMessage.innerHTML = i;
} // avancerSetTimeout

// -----
function avancerSetInterval() {
    // blanchiment
    for (var j = 0; j < t.length; j++) {
        document.getElementById("td" + j).style.backgroundColor =
"white";
    }
    // colorisation
    console.log("passages dans avancerSetInterval : " + passages);
    if (passages == t.length - 1) {
        clearInterval(oInterval);
    }
    var cellule = "td" + passages;
    console.log(cellule);
    document.getElementById(cellule).style.backgroundColor =
"red";
    lblMessage.innerHTML = passages;
    passages++;
} // avancer

// -----
window.onload = init;

```



## 1.3 - PARTITIONNEMENT TEMPORISÉ

### 1.3.1 - Écrans

#### Partitionner\_1.html

9	1	7	4	3	2	6	5	8
9	1	7	4	3	2	6	5	8

Phase 1

Phase 2

Valeur pivot : 4

Indice pivot : 3

Indice début : 0

Indice fin : 8

#### Partitionner\_1.html

9	1	7	4	3	2	6	5	8
9	1	7	8	3	2	6	5	4

Phase 1

Phase 2

Valeur pivot : 4

Indice pivot : 8

Indice début : 0

Indice fin : 8

**Partitionner\_1.html**

9	1	7	4	3	2	6	5	8
1	9	7	8	3	2	6	5	4

Phase 1

Phase 2

**Partitionner\_1.html**

9	1	7	4	3	2	6	5	8
1	3	7	8	9	2	6	5	4

Phase 1

Phase 2

**Partitionner\_1.html**

9	1	7	4	3	2	6	5	8
1	3	2	4	9	7	6	5	8

Phase 1

Phase 2

## 1.3.2 - Codes

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link href="../../css/jsAlgo.css" rel="stylesheet" />
    <title>Partitionner_1.html</title>
  </head>

  <body>

    <article id="article">
      <h1>Partitionner_1.html</h1>

      <table>
        <tr id="ligneTableauDepart">
        </tr>
      </table>

      <table>
        <tr id="ligneTableau">
        </tr>
      </table>
      <p>
        <input type="button" value="Phase 1" id="btPhase1"
/>
      </p>
      <p>
        <input type="button" value="Phase 2" id="btPhase2"
/>
      </p>

      <p>
        Valeur pivot :
        <label id="lblValeurPivot"></label>
      </p>
      <p>
        Indice pivot :
        <label id="lblIndicePivot"></label>
      </p>
      <p>
        Indice début :
        <label id="lblIndiceDebut"></label>
      </p>
      <p>
        Indice fin :
        <label id="lblIndiceFin"></label>
      </p>

    </article>

```

```
        <script src="../../js/Partitionner_1.js"></script>
    </body>
</html>
```

```

/*
 * Partitionner_1.js
 */

var ligneTableau;
var ligneTableauDepart;

var lblValeurPivot;
var lblIndiceDebut;
var lblIndiceFin;
var lblIndicePivot;

var btPhase1;
var btPhase2;

var valeurPivot = 4;
var indicePivot = 3;
var indiceDebut = 0;
var indiceFin;
var t = new Array(9, 1, 7, 4, 3, 2, 6, 5, 8);
var passages;
var j = 0;

var oDelai;

// -----
function init() {
    ligneTableauDepart =
document.getElementById("ligneTableauDepart");
    ligneTableau = document.getElementById("ligneTableau");

    lblValeurPivot = document.getElementById("lblValeurPivot");
    lblIndiceDebut = document.getElementById("lblIndiceDebut");
    lblIndiceFin = document.getElementById("lblIndiceFin");
    lblIndicePivot = document.getElementById("lblIndicePivot");

    btPhase1 = document.getElementById("btPhase1");
    btPhase2 = document.getElementById("btPhase2");

    btPhase1.onclick = mettrePivotALaFin;
    btPhase2.onclick = partitionner;

    indiceFin = t.length - 1;

    lblIndiceDebut.innerHTML = indiceDebut;
    lblIndiceFin.innerHTML = indiceFin;
    lblIndicePivot.innerHTML = indicePivot;
    lblValeurPivot.innerHTML = valeurPivot;

    afficher();
} /// init

// -----
function afficher() {

```

```

var parentDepart = ligneTableauDepart;
var parent = ligneTableau;
for (var i = 0; i < t.length; i++) {
    // Tableau de départ
    var tdDepart = document.createElement("td");
    tdDepart.setAttribute("id", "tdDepart" + i);
    parentDepart.appendChild(tdDepart);
    tdDepart.innerHTML = t[i];

    // Tableau à manipuler
    var td = document.createElement("td");
    td.setAttribute("id", "td" + i);
    parent.appendChild(td);
    td.innerHTML = t[i];
}

document.getElementById("tdDepart" +
indicePivot).style.backgroundColor = "red";
document.getElementById("td" +
indicePivot).style.backgroundColor = "red";
} /// afficher

/**
 *
 * @returns {undefined}
 */
function mettrePivotALaFin() {

    /*
    Permutation
    */
    // blanchiment
    for (var k = 0; k < t.length; k++) {
        document.getElementById("td" + k).style.backgroundColor =
"white";
    }

    var inter;
    inter = t[indiceFin];
    t[indiceFin] = valeurPivot;
    document.getElementById("td" + indiceFin).innerHTML =
valeurPivot;
    t[indicePivot] = inter;
    document.getElementById("td" + indicePivot).innerHTML = inter;

    indicePivot = t.length - 1;
    lblIndicePivot.innerHTML = indicePivot;

    document.getElementById("td" +
indicePivot).style.backgroundColor = "red";
} /// mettrePivotALaFin

/**
 *
```

```

    * @returns {undefined}
    */
function partitionner() {
    j = 0;
    // blanchiment
    for (var k = 0; k < t.length; k++) {
        document.getElementById("td" + k).style.backgroundColor =
"white";
    }
    /*

    */
    // for (var i = 0; i < t.length; i++) {
    //     oDelai = setTimeout(function() {
    //         console.log ("i dans partitionner : " + i);
    //         permuter(i);
    //     }, 1000);
    //     console.log ("i dans partitionner : " + i);
    // } /// for i
    // clearInterval(oDelai);
    passages = 0;
    oDelai = setInterval(function() {
        permuter();
    }, 500);
} /// partitionner

/**
 *
 * @returns {undefined}
 */
function permuter() {
    var cellule;
    // blanchiment
    for (var k = 0; k < t.length; k++) {
        document.getElementById("td" + k).style.backgroundColor =
"white";
    }
    console.log("passages dans permuter : " + passages);
    var inter;
    if (t[passages] < valeurPivot) {
        // Permutation
        inter = t[passages];
        t[passages] = t[j];
        cellule = "td" + passages;
        document.getElementById(cellule).innerHTML = t[passages];
        cellule = "td" + j;
        document.getElementById(cellule).innerHTML = inter;
        t[j] = inter;
        // Increment
        j++;
        console.log(j);
        // Affichage
        var cellule = "td" + j;
        document.getElementById(cellule).style.backgroundColor =
"red";
    }
}

```

```
    }
    passages++;
    if (passages == t.length) {
        clearInterval(oDelai);
        inter = t[t.length - 1]; // La valeur du pivot
        cellule = "td" + j;
        document.getElementById(cellule).innerHTML = valeurPivot;
        t[t.length - 1] = t[j];
        cellule = "td" + (t.length - 1);
        document.getElementById(cellule).innerHTML = t[j];
        t[j] = inter;
        cellule = "td" + j;
        document.getElementById(cellule).style.backgroundColor =
"red";

        for (var k = 0; k < t.length; k++) {
            console.log(t[k]);
        }
    }
} /// permuter

// -----
window.onload = init;
```