



Algorithmique Préparation

« La plus grande attention doit être portée à la compréhension du problème, faute de quoi l'algorithme n'a aucune chance d'être correct ». Denis Lapoire

« Une petite impatience ruine un grand projet. »
Citation de Confucius ; Livre des sentences - VIe s. av. J.-C.

J'écoute et j'oublie.
Je lis et je retiens.
Je fais et j'apprends.
(Proverbe chinois)



TABLE DES MATIÈRES

1 - Présentation du document.....	4
2 - Prémisses.....	5
2.1 - Comment faire une omelette en cassant des œufs ?.....	5
2.2 - Définition d'un algorithme.....	7
2.3 - Instructions de base.....	7
3 - Algorithmique et JavaScript.....	8
3.1 - Notions de JavaScript.....	9
3.2 - Débogage.....	10
3.3 - Les variables.....	12
3.4 - Quelques opérateurs JavaScript.....	13
4 - Exemples d'algorithmes sans formulaire.....	14
4.1 - Modèle de page sans formulaire de saisie.....	15
4.2 - L'affectation.....	16
4.2.1 - Objectif : afficher dans un <label> la valeur d'une variable.....	16
4.2.2 - Syntaxes.....	17
4.2.3 - Code.....	17
4.3 - La structure séquentielle et les nombres.....	18
4.3.1 - Définition.....	18
4.3.2 - Objectif.....	18
4.3.3 - Code.....	18
4.3.4 - Exercices.....	19
4.4 - Les chaînes de caractères et la concaténation.....	20
4.4.1 - Définition.....	20
4.4.2 - Objectif.....	20
4.4.3 - Syntaxes.....	21
4.4.4 - Code.....	21
4.5 - Le IF pour les conditions.....	22
4.5.1 - Objectif.....	22
4.5.2 - Syntaxe.....	22
4.5.3 - Code.....	23
4.5.4 - Exercice.....	23
4.6 - La boucle FOR.....	24
4.6.1 - Objectif.....	24
4.6.2 - Syntaxe.....	24
4.6.3 - Code.....	25
4.6.4 - Exercices.....	26
4.7 - Les Tableaux ordinaux à 1 dimension.....	27
4.7.1 - Définition.....	27
4.7.2 - Objectif.....	27
4.7.3 - Syntaxes.....	28

4.7.4 - Un exemple.....	29
4.7.5 - Exercices.....	30
5 - Exemples d'algorithmes avec un formulaire de saisie.....	31
5.1 - Modèle de page avec un formulaire de saisie.....	32
5.2 - L'affectation.....	34
5.2.1 - Objectif.....	34
5.2.2 - Exercices.....	36
5.3 - Le IF pour les conditions.....	37
5.3.1 - Objectif.....	37
5.3.2 - Un exemple : majeur ou mineur ?.....	38
5.3.3 - IF imbriqués et condition complexe.....	40
5.3.4 - Exercice.....	42
6 - Annexes.....	43
6.1 - Les opérateurs logiques.....	43
7 - Bibliographie et Webographie.....	44
7.1 - Bibliographie.....	44
7.2 - Webographie.....	44

1 - PRÉSENTATION DU DOCUMENT

Ce document est une préparation à un cours d'algorithmique Initiation qui lui même sera suivi d'un cours d'algorithmique Avancé. C'est une pré-initiation !

L'accent est mis sur les trois structures de base d'un algorithme.

Les connaissances informatiques requises sont de connaître **Windows** ou **Linux** en mode graphique ou **iOS**, plus précisément la manipulation d'un **explorateur de fichiers**, la connaissance de base d'**HTML**, la manipulation d'un **navigateur** et d'un **éditeur de textes ASCII**.

Les exemples seront codés en JavaScript (pas en Java).

JavaScript est le langage des navigateurs qui permet de dynamiser les pages statiques HTML.

Ceci n'est pas un cours JavaScript.

Pratique de ce document.

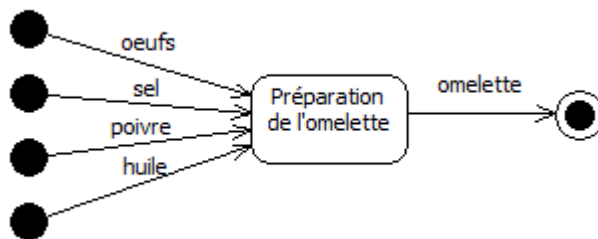
Vous devez :

créer un seul dossier,
créer un fichier .html et pour chaque exemple copier/coller le code du support de cours,
créer un fichier .js et pour chaque exemple copier/coller le code du support de cours,
tester chaque fichier avec un navigateur (clic droit sur le fichier .html / Ouvrir avec / votre navigateur préféré).

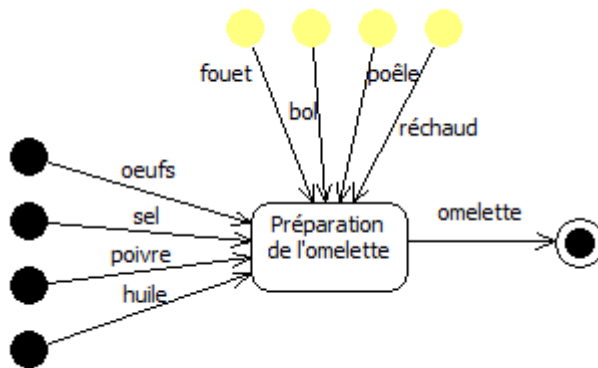
2 - PRÉMISSES

2.1 - COMMENT FAIRE UNE OMELETTE EN CASSANT DES ŒUFS ?

Pour obtenir une omelette il faut des œufs, du sel, du poivre et de l'huile d'olive (des ingrédients).



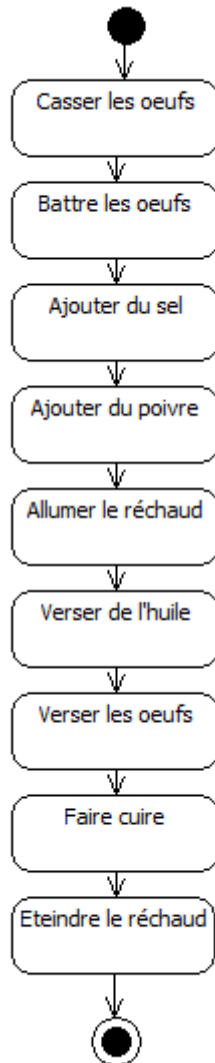
Mais il faut aussi un bol ou une assiette creuse, une fourchette ou un fouet, une poêle, un réchaud, etc (des ustensiles).



Les ustensiles informatiques sont les variables et les instructions.

et aussi .../...

une recette (un mode opératoire, un ordonnancement).



En somme pour obtenir un PLAT il faut des INGREDIENTS, des USTENSILES et un MODE OPERATOIRE.



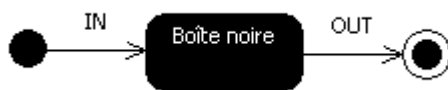
Dans un algorithme c'est la même chose : pour obtenir un ou plusieurs OUTPUTS il faut un ou plusieurs INPUTS et un mode opératoire.

2.2 - DÉFINITION D'UN ALGORITHME

Un algorithme est un procédé de calcul; il décrit une **succession d'opérations** à exécuter dans un certain ordre et sous certaines conditions pour passer des données de base (les inputs) aux résultats (les outputs). Il correspond à un processus, découpé en étapes dont l'enchaînement permet d'arriver à un but fixé.

Cette définition convient au **paradigme de la programmation impérative** (structurée et procédurale). Le paradigme de la programmation impérative est le suivant : on s'intéresse aux actions.

Écrire un algorithme c'est blanchir une boîte noire. C'est « découvrir » les actions à faire exécuter pour passer des entrées aux sorties.



A côté des instructions il faut manipuler des **variables**.

Une variable est un espace mémoire, de taille plus ou moins importante, pour stocker temporairement des valeurs utiles aux calculs. Les contenus des variables peuvent être des textes, des numériques, des dates, etc.

2.3 - INSTRUCTIONS DE BASE

Les instructions de base de la programmation impérative sont :

- ✓ la déclaration des variables,
- ✓ l'affectation : assigner une valeur ou le résultat d'une expression à une variable,
- ✓ le branchement conditionnel : exécuter une opération si une condition prédéfinie est remplie,
- ✓ le branchement sans condition : appeler une procédure ou une fonction,
- ✓ l'itération : exécuter plusieurs fois la même opération.

3 - ALGORITHMIQUE ET JAVASCRIPT

3.1 - NOTIONS DE JAVASCRIPT

Les commentaires sont entre `/*` et `*/` ou après `//`.

Toute instruction JS est terminée par un `;` (point-virgule).

Les noms des variables, des constantes, des instructions, des fonctions sont sensibles à la casse c'est-à-dire que les variables `NomDuClient` et `nomduclient` sont deux variables différentes. Ceci n'est pas vrai dans tous les langages.

Les variables, les constantes et les fonctions ne sont pas typées.

Une variable est déclarée après le mot **var**.

Elle peut être initialisée lors de la déclaration.

Une variable globale (déclarée hors d'une fonction) est utilisable hors et dans une fonction.

Une variable locale (déclarée dans une fonction) n'est utilisable que dans la fonction où elle est déclarée.

if est utilisé pour les conditionnelles. La condition est entre parenthèses **()**.

for et **while** sont utilisés pour les boucles. Comme pour le **if**, les parenthèses ouvrent et ferment la condition.

document.getElementById("element") est utilisé pour pointer vers un élément HTML identifié.

`element.onclick = nomDeFonction;` est utilisé pour associer un élément et l'événement clic à une fonction.

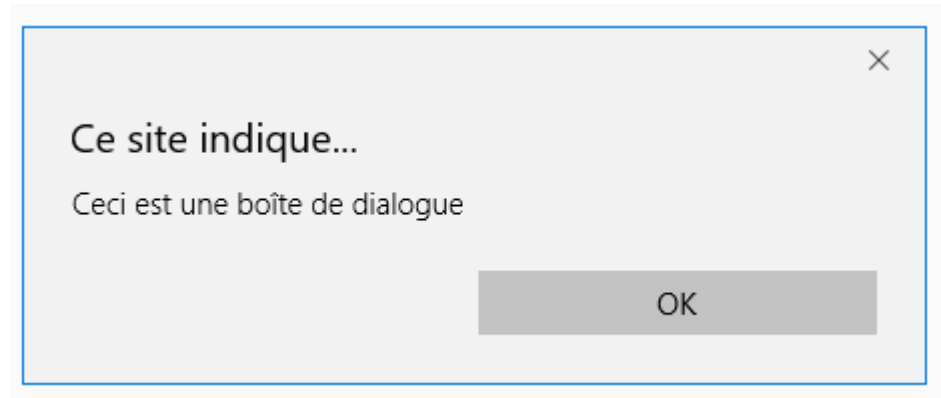
`element.value` est utilisé pour affecter ou récupérer la valeur d'un élément HTML qui possède l'attribut `value`.

`element.innerHTML` est utilisé pour affecter ou récupérer la valeur d'un élément HTML qui ne possède pas l'attribut `value`.

Cf les annexes pour plus de détails.

3.2 - DÉBOGAGE

alert(variable); cette instruction est utilisée pour afficher le contenu d'une variable ou d'un texte dans une boîte de dialogue.



Nom du fichier : Alerte.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Alerte</title>

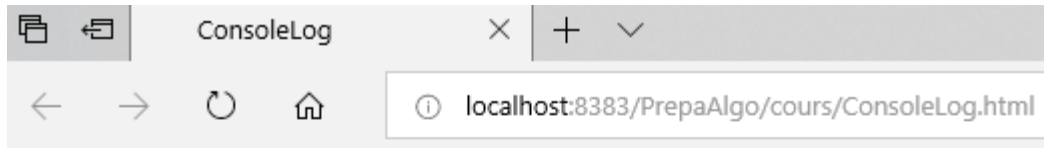
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

  </head>
  <body>
    <h3>Alerte</h3>

    <script>
      // CODE JavaScript
      alert("Ceci est une boîte de dialogue");
    </script>

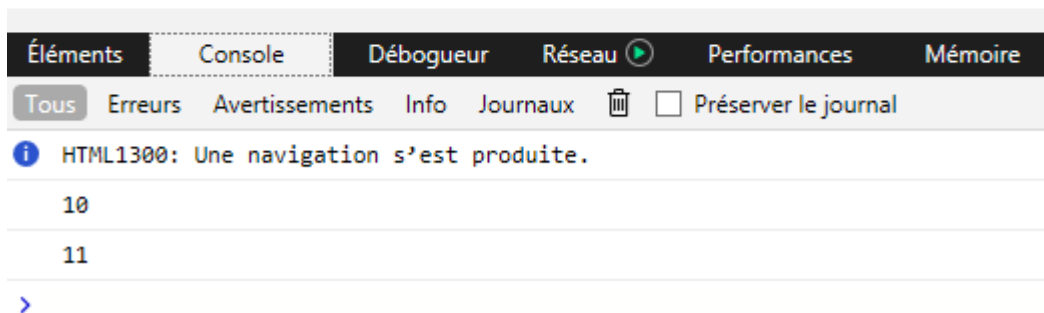
  </body>
</html>
```

console.log(variable); cette instruction est utilisée pour afficher le contenu d'une variable à la console du navigateur (F12 pour la plupart des navigateurs et ensuite l'onglet « Console »).



ConsoleLog

Ouvrez la console avec F12 puis sélectionnez l'onglet "Console"



Nom du fichier : ConsoleLog.html

```
<!DOCTYPE html>

<html lang="fr">
  <head>
    <title>ConsoleLog</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

  </head>

  <body>
    <h3>ConsoleLog</h3>
    <p>Ouvrez la console avec F12 puis sélectionnez l'onglet
"Console"</p>
    <script>
      // CODE JavaScript
      var i = 10;
      console.log(i);
      i++;
      console.log(i);
    </script>

  </body>
</html>
```

3.3 - LES VARIABLES

Dans cet exemple une variable est déclarée et initialisée, ensuite son contenu est affiché dans une boîte de dialogue puis modifié (on ajoute 1 à la valeur précédente), et enfin affiché à nouveau dans une boîte de dialogue.

```
var i = 10;  
alert(i);  
i++;  
alert(i);
```

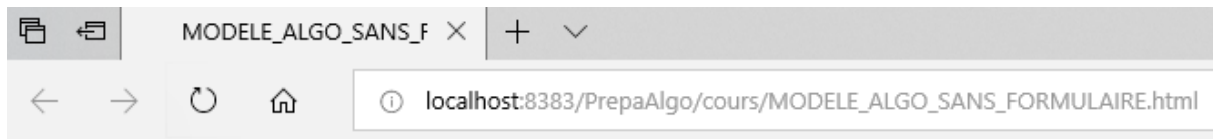
3.4 - QUELQUES OPÉRATEURS JAVASCRIPT

Catégories	Opérateurs
Affectation	=
Arithmétiques	+, -, *, /
Comparaisons	== pour l'égalité > pour supérieur à >= pour inférieur à < pour inférieur à <= pour inférieur ou égal à != pour différent de
Logiques	&& pour le ET logique pour le OU logique ! pour le non logique
Concaténation (juxtaposer deux textes)	+

Pour plus de détails cf les Annexes.

4 - EXEMPLES D'ALGORITHMES SANS FORMULAIRE

4.1 - MODÈLE DE PAGE SANS FORMULAIRE DE SAISIE



Modèle Algo sans formulaire

3

C'est-à-dire que nous utiliserons les éléments HTML suivants :
titre,
paragraphe (identifié).

```
<!DOCTYPE html>

<html lang="fr">
  <head>
    <title>MODELE_ALGO_SANS_FORMULAIRE.HTML</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

  </head>

  <body>
    <h3>Modèle Algo sans formulaire</h3>
    <p id="pAffichage"></p>
    <script>
      // CODE JavaScript
      var i = 3;
      document.getElementById("pAffichage").innerHTML = i;
    </script>

  </body>
</html>
```

Remarques :

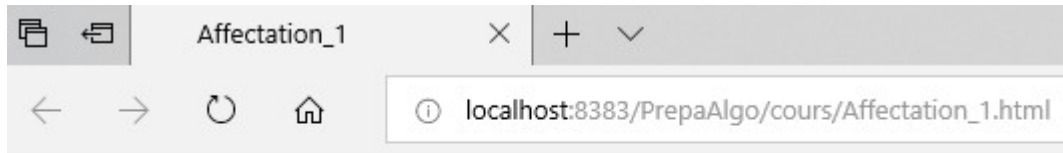
Tous les éléments qui seront manipulés par l'algorithme seront identifiés grâce à l'**attribut id**.
Le code JavaScript (l'algorithme) est stocké dans la page HTML entre la balise **<script>** et la balise **</script>**.

Il pourrait être stocké dans un fichier à part lui-même stocké dans un autre dossier que le fichier HTML.

Ces différentes possibilités seront vues en cours.

4.2 - L'AFFECTATION

4.2.1 - Objectif : afficher dans un <label> la valeur d'une variable.



Affectation_1

3

4.2.2 - Syntaxes

Déclaration d'une variable et initialisation

```
var nomDeLaVariable = valeur;
```

Affichage du contenu d'une variable dans un paragraphe

```
document.getElementById("identifiant du paragraphe").innerHTML =  
nomDeLaVariable;
```

4.2.3 - Code

```
<!DOCTYPE html>  
  
<html lang="fr">  
  <head>  
    <title>Affectation_1</title>  
  
    <meta name="viewport" content="user-scalable=no, width=device-  
width, initial-scale=1.0, maximum-scale=1.0" />  
    <meta charset="utf-8" />  
  
  </head>  
  
  <body>  
    <h3>Affectation_1</h3>  
    <p id="pAffichage"></p>  
    <script>  
      // CODE JavaScript  
      var i = 3;  
      document.getElementById("pAffichage").innerHTML = i;  
    </script>  
  </body>  
</html>
```

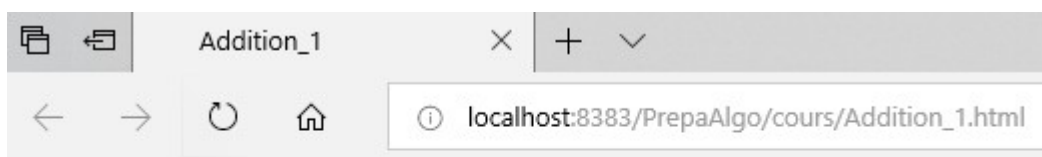
4.3 - LA STRUCTURE SÉQUENTIELLE ET LES NOMBRES

4.3.1 - Définition

La première structure de programmation est la structure séquentielle. Ce qui veut dire que **toutes** les instructions sont toujours exécutées en séquence et toujours dans le même ordre.

4.3.2 - Objectif

Additionner 2 nombres.



Addition_1

8

4.3.3 - Code

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Addition_1</title>
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>
  <body>
    <h3>Addition_1</h3>
    <p id="pAffichage"></p>
    <script>
      // Code JavaScript
      var a = 3;
      var b = 5;
      var r = a + b;
      document.getElementById("pAffichage").innerHTML = r;
    </script>
  </body>
</html>
```

Remarque : 4 instructions sont exécutées : première déclaration et initialisation, une deuxième, un calcul et un affichage.

4.3.4 - Exercices

La soustraction.
La multiplication.
La division.

4.4 - LES CHÂÎNES DE CARACTÈRES ET LA CONCATÉNATION

4.4.1 - Définition

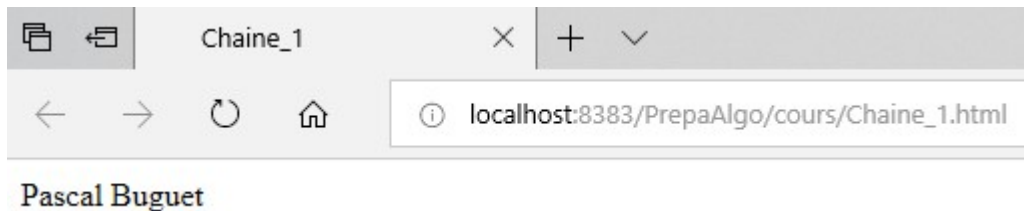
Une chaîne de caractères est un ensemble de caractères, un mot, une phrase.

Une chaîne de caractères est écrite entre des simples quotes droites (') ou des doubles quotes (").

Précédemment les valeurs de type numérique étant sans quote ni double quote.

4.4.2 - Objectif

Le prénom et le nom sont concaténés avec un espace entre les 2.



4.4.3 - Syntaxes

L'opérateur + est utilisé pour concaténer 2 chaînes de caractères.

4.4.4 - Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Chaine_1</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>

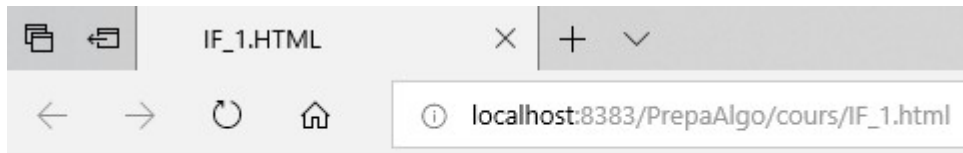
  <body>
    <label id="pAffichage"></label>
    <script>
      // Déclaration des variables
      var prenom = "Pascal"; // Chaîne initialisée
      var nom = "Buguet"; // Chaîne initialisée
      // Concaténation
      var prenomEtNom = prenom + " " + nom;
      // Affichage
      document.getElementById("pAffichage").innerHTML = prenomEtNom;
    </script>
  </body>
</html>
```

4.5 - LE IF POUR LES CONDITIONS

4.5.1 - Objectif

Explorer la structure conditionnelle.

Afficher Majeur ou Mineur en fonction de l'âge.



Majeur

4.5.2 - Syntaxe

```
if (condition)
{
    Action a1;
    Action a2;
}
[else
{
    Action b1;
    Action b2; ...
}]
```

4.5.3 - Code

```
<!DOCTYPE html>
<!-- Un IF simple :
SI la valeur est >= valeur alors ... SINON ...
-->
<html>
  <head>
    <title>IF_1.HTML</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>

  <body>
    <div>
      <p id="pAffichage"></p>
    </div>

    <script>
      var age = 17;
      // Si l'âge est supérieur ou égal à 18
      if (age >= 18) {
        document.getElementById("pAffichage").innerHTML =
"Majeur";
      }
      // Si l'âge est inférieur à 18
      else {
        document.getElementById("pAffichage").innerHTML =
"Majeur";
      }
    </script>
  </body>
</html>
```

Testez cette page HTML et ensuite modifiez la valeur et l'âge et re-testez.

4.5.4 - Exercice

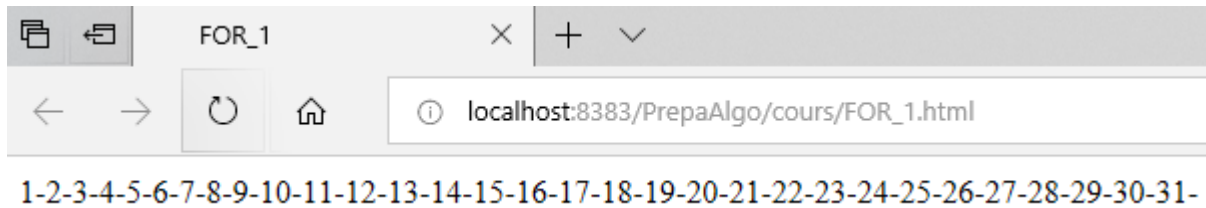
Additionnez 2 nombres. Affichez la nature du résultat : Positif ou Négatif.

4.6 - LA BOUCLE FOR

4.6.1 - Objectif

Structure itérative ie faire une boucle.

Afficher sur la même ligne des valeurs allant de 1 à 31 séparées par des traits-d'union.



4.6.2 - Syntaxe

```
for (var initialisation du compteur; condition de sortie;
    incrémentation/décrémentation du compteur)
{
    instruction 1;
    instruction 2; ...
}
```


4.6.3 - Code

On utilise la concaténation pour cumuler dans la chaîne de caractères résultante.

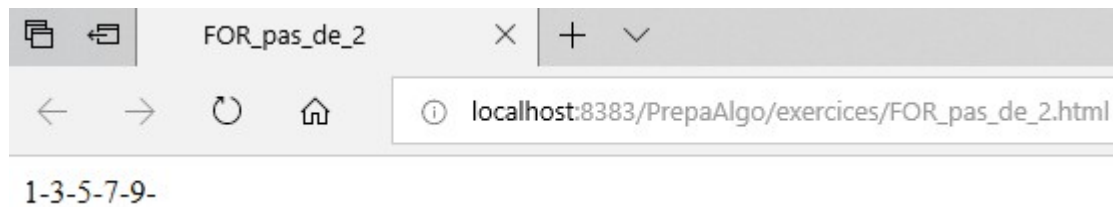
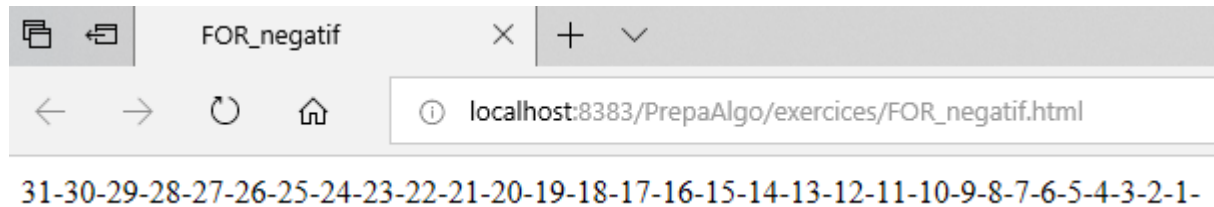
```
<!DOCTYPE html>
<html>
  <head>
    <title>FOR_1</title>
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>
  <body>
    <label id="pAffichage"></label>
    <script>
      // Déclaration des variables
      var texte = ""; // Chaîne initialisée à chaîne vide

      // Boucle de 1 a 31
      for (var i = 1; i <= 31; i++) {
        texte = texte + i + "-"; // Concatenation de la valeur du
compteur et d'un -
      }

      // Affichage
      document.getElementById("pAffichage").innerHTML = texte;
    </script>
  </body>
</html>
```

4.6.4 - Exercices

Algo	Description
For-	Les nombres de 31 à 1
ForPasDe2	Les chiffres impairs



4.7 - LES TABLEAUX ORDINAUX À 1 DIMENSION

4.7.1 - Définition

Les tableaux à indices. Le premier indice est 0.

Représentation d'un tableau 1D :

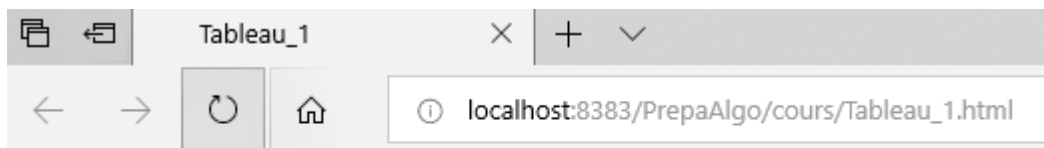
1	3	5	7	9
---	---	---	---	---

et avec les indices :

0	1	2	3	4
1	3	5	7	9

4.7.2 - Objectif

Affichez ceci :



Contenu du tableau : 1,3,5,7,9,11

Nombre d'éléments : 6

2ème élément : 3

4.7.3 - Syntaxes

Initialisation d'un tableau

```
var t = new Array(1,3,5,7,9);
```

Nombre d'éléments d'un tableau

```
var nb = t.length;
```

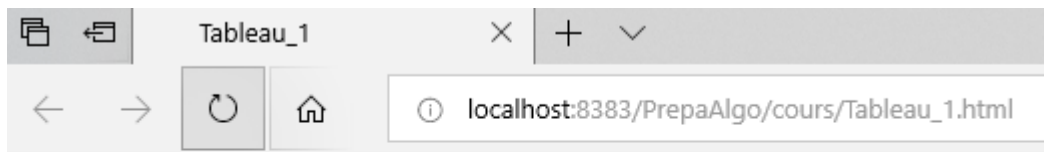
Modification de la valeur d'un élément

```
t[indice] = valeur;
```

Récupération de la valeur d'un élément

```
var valeur = t[indice];
```

4.7.4 - Un exemple



Contenu du tableau : 1,3,5,7,9,11

Nombre d'éléments : 6

2ème élément : 3

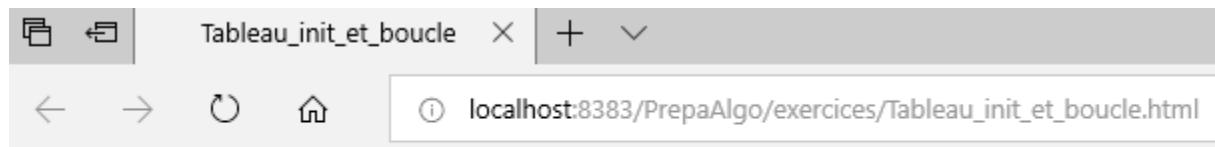
Affiche le contenu du tableau, le nombre d'éléments et le 2ème élément.
Pour afficher tout cela on concatène dans le texte d'un élément HTML.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Tableau_1</title>
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>
  <body>
    <div>
      <p id="pResultats"></p>
    </div>
    <script>
      var texte = "";
      var t = new Array(1, 3, 5, 7, 9, 11);

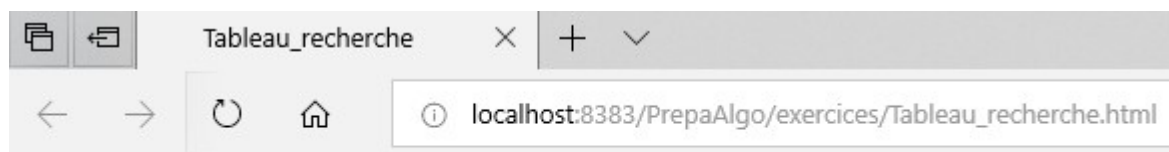
      texte = texte + "Contenu du tableau : " + t + "<br>";
      texte = texte + "Nombre d'éléments : " + t.length + "<br>";
      texte = texte + "2ème élément : " + t[1];
      document.getElementById("pResultats").innerHTML = texte;
    </script>
  </body>
</html>
```

4.7.5 - Exercices

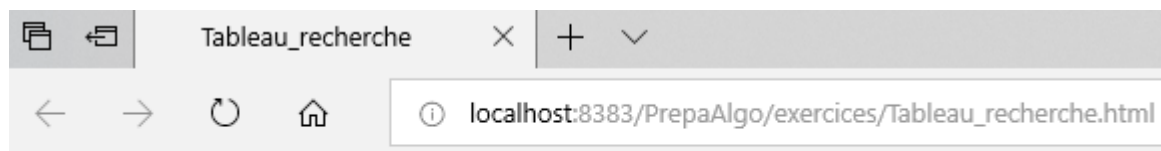
Algo	Description
Initialiser et afficher	Création d'un tableau et affichage des valeurs du tableau dans une boucle
Rechercher une valeur	Affichage de la position d'une valeur dans un tableau ; si la valeur n'est pas trouvée affichage de -1



Contenu du tableau : 1-3-5-7-9-



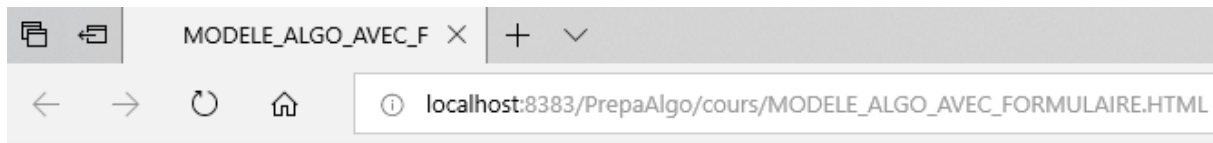
La position de 5 est 3



La position de 15 est -1

5 - EXEMPLES D'ALGORITHMES AVEC UN FORMULAIRE DE SAISIE

5.1 - MODÈLE DE PAGE AVEC UN FORMULAIRE DE SAISIE



Modele de page pour l'algo avec un formulaire

Saisissez !

C'est-à-dire que nous utiliserons les éléments HTML suivants :

titre,
label,
zone de saisie,
bouton,
paragraphe.

Le code de la page MODELE_ALGO_AVEC_FORMULAIRE.html :

```
<!DOCTYPE html>
<html>
  <head>
    <title>MODELE_ALGO_AVEC_FORMULAIRE</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  </head>
  <body>
    <h3>Modele de page pour l'algo avec un formulaire</h3>
    <div>
      <label>Saisissez !</label>
      <input type="text" id="itSaisie" />
      <button id="btValider">Valider</button>
      <p id="pResultats"></p>
    </div>
    <script>
      // Code JavaScript
      var elementDeSaisie = document.getElementById("itSaisie");
      var elementDAffichage = document.getElementById("pResultats");
      var bouton = document.getElementById("btValider");

      bouton.onclick = function () {
        // Affichage de résultats
        elementDAffichage.innerHTML = elementDeSaisie.value;
      };
    </script>
  </body>
</html>
```

Remarques :

Tous les éléments qui seront manipulés par l'algorithme seront identifiés grâce à l'**attribut id**.

On récupère dans une variable JavaScript un élément HTML avec **document.getElementById("id de l'élément")**

Le texte d'un élément de formulaire est dans l'attribut « **value** ».

Le texte d'un élément qui n'est pas un élément de formulaire est dans l'attribut « **innerHTML** ».

Le code JavaScript (l'algorithme) est stocké dans la page HTML entre la balise **<script>** et la balise **</script>**.

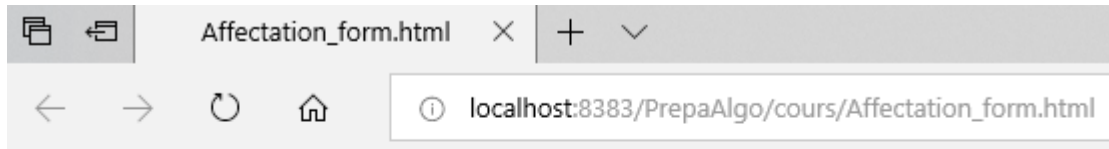
Il pourrait être stocké dans un fichier à part lui-même stocké dans un autre dossier que le fichier HTML.

Ces différentes possibilités seront vues en cours.

5.2 - L'AFFECTATION

5.2.1 - Objectif

Afficher dans un <paragraphe> une valeur saisie dans un <input>



Affectation_form

Saisissez !

Pascal

Le code JavaScript réalise ceci :

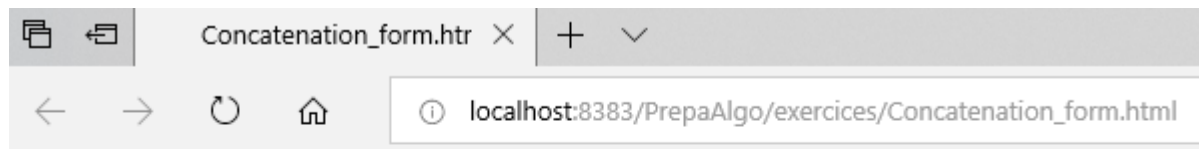
stocke dans des variables les éléments HTML utilisés dans le code,
et dans une fonction anonyme qui gère le clic
récupère la valeur saisie,
affecte à l'élément d'affichage la valeur saisie.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Affectation_form.html</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  </head>
  <body>
    <h3>Affectation_form</h3>
    <div>
      <label>Saisissez !</label>
      <input type="text" id="itSaisie" />
      <button id="btValider">Valider</button>
      <p id="pResultats"></p>
    </div>
    <script>
      // Code JavaScript
      var elementDeSaisie = document.getElementById("itSaisie");
      var elementDAffichage = document.getElementById("pResultats");
      var bouton = document.getElementById("btValider");

      // Gestion du click sur le bouton
      // Par association du click à une fonction anonyme
      bouton.onclick = function () {
        // Récupération de la valeur saisie
        var valeurSaisie = elementDeSaisie.value;
        // Affichage de résultats
        elementDAffichage.innerHTML = valeurSaisie;
      };
    </script>
  </body>
</html>
```

5.2.2 - Exercices

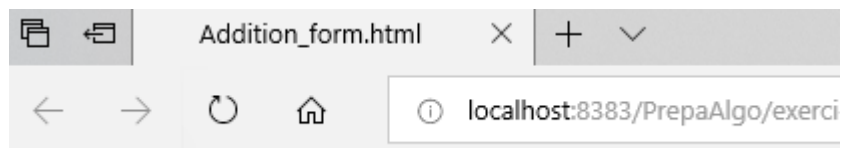
Algo	Description
Concaténation	Afficher nom et prénom saisis
Addition	Addition de 2 saisies



Concatenation_form

Prénom : Nom :

Pascal Buguet



Addition_form

A + B =

5.3 - LE IF POUR LES CONDITIONS

5.3.1 - Objectif

Explorer la structure conditionnelle.
Déjà vu.

5.3.2 - Un exemple : majeur ou mineur ?

Afficher Majeur ou Mineur en fonction de l'âge saisi.

IF

Votre âge ? Majeur

IF

Votre âge ? Mineur

```
<!DOCTYPE html>
<!-- Un IF simple :
SI la valeur saisie est >= valeur alors ... SINON ...
-->
<html>
  <head>
    <title>IF_form.HTML</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>

  <body>
    <div>
      <label>Votre âge ?</label>
      <input type="text" id="itAge" value="18" />
      <input type="button" id="btOK" value="OK" />
      <label id="lblMessage"></label>
    </div>

    <script>
      // Quand on clique sur le bouton btOK
      var itAge = document.getElementById("itAge");
      var btOK = document.getElementById("btOK");
      var lblMessage = document.getElementById("lblMessage");
      btOK.onclick = function () {
        // On recupere la valeur saisie dans la variable nommee
age
        var age = itAge.value;
        // On declare et on initialise a chaine vide la variable
nommee etat

        var etat = "";
        // Si l'age est superieur ou egal a 18
        if (age >= 18) {
          etat = "Majeur";
        }
        // Si l'age est inferieur a 18
        else {
          etat = "Mineur";
        }
        // On affiche l'etat
        lblMessage.innerHTML = etat;
      };
    </script>
  </body>
</html>
```

5.3.3 - IF imbriqués et condition complexe

La même chose mais l'âge doit être compris entre 0 et 130. Voir les conditions complexes dans les annexes.

IFsImbriques

Votre âge ? Age étrange !!!

Le code HTML est le même.

```
<!DOCTYPE html>
<html>
  <head>
    <title>IF_form.HTML</title>

    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />
  </head>
  <body>
    <div>
      <label>Votre âge ?</label>
      <input type="text" id="itAge" value="18" />
      <input type="button" id="btOK" value="OK" />
      <label id="lblMessage"></label>
    </div>
    <script>
      // Quand on clique sur le bouton btOK
      var itAge = document.getElementById("itAge");
      var btOK = document.getElementById("btOK");
      var lblMessage = document.getElementById("lblMessage");
      btOK.onclick = function () {
        // On récupère la valeur saisie dans la variable nommée
age
        var age = itAge.value;
        // On déclare et on initialise la chaîne vide la variable
nommée etat
        var etat = "";
        if (age >= 0 && age <= 130) {
          if (age >= 18) {
            etat = "Majeur";
          } else {
            etat = "Mineur";
          }
        } else {
          etat = "Age étrange !!!";
        }
        lblMessage.innerHTML = etat;
      };
    </script>
  </body>
</html>
```

5.3.4 - Exercice

Algo	Description
If else imbriqués	Contrôles de la saisie de l'âge ... suite... L'âge doit être saisi

6 - ANNEXES

6.1 - LES OPÉRATEURS LOGIQUES

Opérateur logique	Description
&&	Et logique
	Ou logique
!	La négation logique

Table de vérités

C1	C2	AND	OR	NOT C1
Vrai	Vrai	Vrai	Vrai	Faux
Vrai	Faux	Faux	Vrai	Faux
Faux	Vrai	Faux	Vrai	Vrai
Faux	Faux	Faux	Faux	Vrai

AND : Le résultat est VRAI si les opérandes C1 et C2 sont VRAI.

OR : Le résultat est VRAI si un des opérandes C1 et C2 est VRAI.

NOT : Le résultat est VRAI l'opérande C1 FAUX.

Exemples :

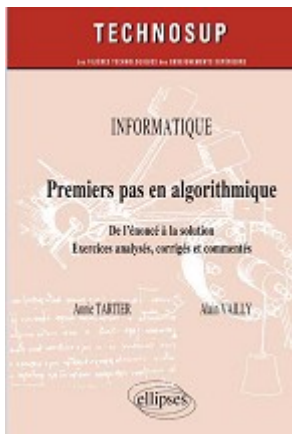
AND : pour être authentifié le pseudo et mot de passe doivent être valides.

OR : pour afficher les produits des catégories Eaux et Sodas il faut utiliser l'opérateur OR.

NOT : pour afficher les villes hors de France il faut utiliser l'opérateur NOT.

7 - BIBLIOGRAPHIE ET WEBOGRAPHIE

7.1 - BIBLIOGRAPHIE



Tartier Annie, Vailly Alain « **INFORMATIQUE - Premier pas en algorithmique** - De l'énoncé à la solution, approche par l'expérimentation »
Exercices analysés, corrigés et commentés (niveau A)
ISBN : 9782340001251
Editions Ellipses
25.00€

7.2 - WEBOGRAPHIE

<http://pise.info/algo/codage.htm>