

# Arbres Mathématiques

## Informatique et Navigateurs Internet

Jacques Duma



Présentation de la structure d'arbre sur des exemples qui montrent où elle intervient en mathématique et en informatique.

Application à l'écriture de pages HTML pour les navigateurs Internet et exemple simple de programmation en langage JavaScript.

Un exemple très simple sera proposé et développé intégralement. Les codes sources fournis, pour la page HTML, pour la feuille de style CSS et pour le programme JavaScript devraient permettre de tester cet exemple sur toute plateforme informatique disposant d'un éditeur de texte et d'un navigateur Internet.

## Table des matières

<b>1 Structures Mathématiques</b>	<b>2</b>
1.1 Arbres	2
1.2 Expressions Mathématiques	3
1.3 Structure arborescente de l'expression	3
<b>2 Informatique : Structure d'une page HTML</b>	<b>4</b>
2.1 Code source HTML	4
2.2 Arbre de la structure DOM	5
2.3 Les Styles avec CSS	5
2.4 Modification de l'aspect visuel	6
<b>3 Représentation informatique d'une expression mathématique</b>	<b>6</b>
3.1 Codage de l'expression mathématique en HTML	7
3.2 Nouvelle structure du DOM	7
3.3 Définition des styles de présentation en CSS	8
3.4 Modification de la présentation	9
<b>4 Programmation dans un Navigateur Internet</b>	<b>9</b>
4.1 La page HTML : « <code>index.html</code> »	10
4.2 La feuille de style CSS : « <code>style.css</code> »	11
4.3 Le programme JavaScript : « <code>code.js</code> »	11
4.4 C'est simple : vous pouvez le faire!	14

# 1 Structures Mathématiques

## 1.1 Arbres

Un graphe est un ensemble de nœuds reliés par des arêtes.

Vocabulaire :

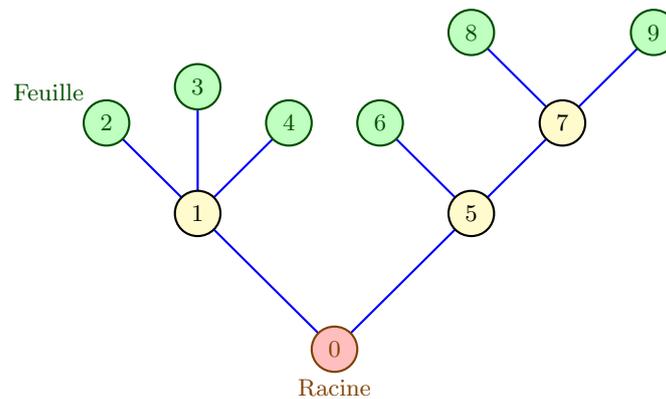
- graphe connexe : *en un seul morceau*
- cycle : *certaines sous-ensembles d'arêtes forment des boucles*

Un arbre est un graphe connexe, sans cycle

On appelle *racine* un nœud quelconque de l'arbre. Tous les autres nœuds sont reliés à la racine par un chemin unique.

### Arbre

C'est un ensemble de sommets ou nœuds (*des points*) reliés par des arêtes ou branches (*des segments*)



Propriétés :

- tous les nœuds, *SAUF la Racine*, ont un *père*
- tous les nœuds ont une *liste de fils*
- on appelle *feuilles* les nœuds dont la liste de fils est vide.
- il existe un chemin unique entre la racine et chaque nœud
- la position d'un nœud dans le graphe est donc définie de façon unique par le chemin qui va de la racine vers ce nœud

Exemples de liens père fils :

- le nœud n° 1 a pour père la Racine et pour fils les nœuds n° {2, 3, 4}
- la Feuille n° 6 a pour père le nœud n° 5 et n'a pas de fils
- la Racine n° 0 n'a pas de père mais a pour fils les nœuds n° {1, 5}

Un arbre peut être facilement exploré grâce à un algorithme simple exécuté à partir de la racine : on traite le nœud, puis ses fils de façon récursive.

## 1.2 Expressions Mathématiques

En mathématique les expressions ont aussi une structure d'arbre.  
L'évaluation se fait de manière non ambiguë en parcourant l'arbre.

### Expressions Mathématiques

#### Différentes formes d'écriture d'une expression

##### Notation usuelle

$$2 \times (a + b)$$

##### Notation fonctionnelle

$$\textit{Produit}(2, \textit{Somme}(a, b))$$

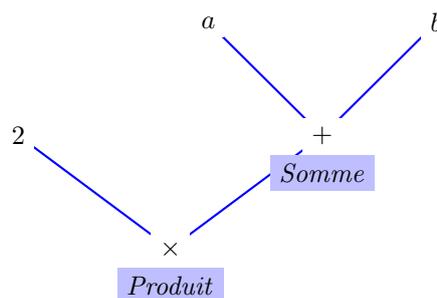
Dans les expressions usuelles les différents opérateurs peuvent être vus comme fonction de leurs arguments.

Toute expression fonctionnelle peut être vue comme un arbre dont la racine est le nom de la fonction et dont les feuilles sont les arguments.

## 1.3 Structure arborescente de l'expression

### Structure arborescente de l'expression

$$2 \times (a + b) = \textit{Produit}(2, \textit{Somme}(a, b))$$

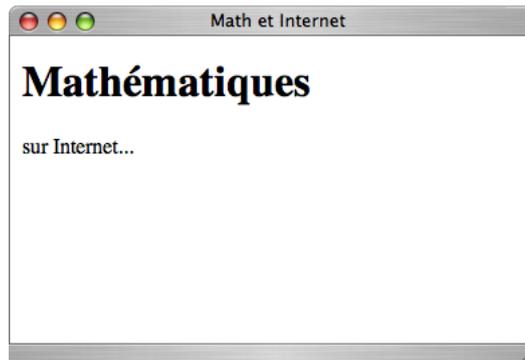


## 2 Informatique : Structure d'une page HTML

Si vous utilisez un navigateur Internet, vous chargez des pages de texte pour les consulter.

Ces pages de texte son codées en HTML (HyperText Markup Language)

### Aspect visuel



Voici un exemple très simple de page HTML.

### 2.1 Code source HTML

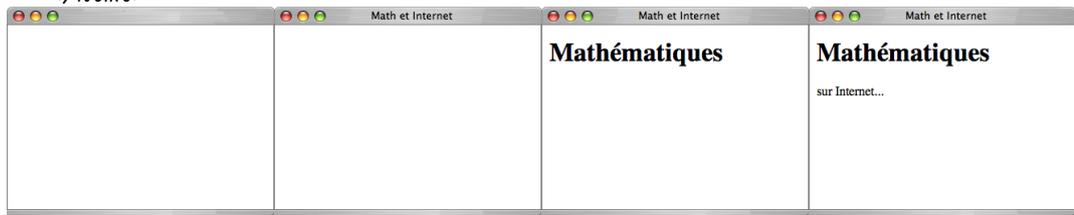
Le HTML ou *Langage à Balises pour l'Hyper-Texte*.

Il existe un certain nombre de balises pré-définies comme `<html>` qui délimite le document entier ou `<p>` qui délimite un simple paragraphe. Ces balises permettent de structurer les informations contenues dans un document.

Tous les textes du document doivent être encadrés entre une balise de début comme `<p>` et une balise de fin comme `</p>` par exemple.

#### Code source HTML : HyperText Markup Language

```
<html>
<head>
<title>Math et Internet</title>
</head>
<body>
<h1>Mathématiques</h1>
<p>sur Internet...</p>
</body>
</html>
```

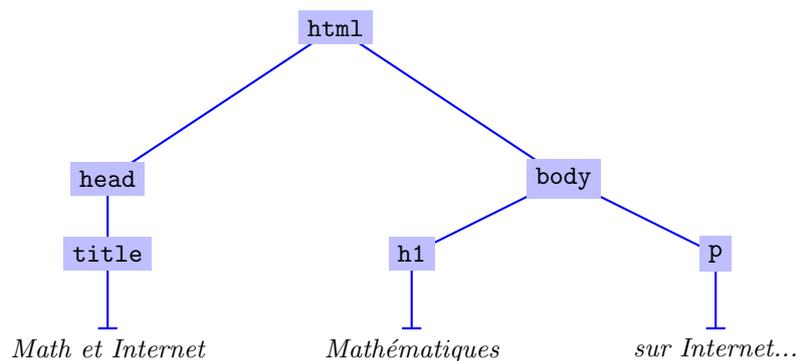


## 2.2 Arbre de la structure DOM

Ce langage permet ainsi de définir la structure de représentation des données textuelles sous forme d'arbre.

Le DOM ou *Modèle Objet du Document* permet une représentation interne de la structure d'arbre du document qui pourra être, plus tard, modifiée par programme.

### Arbre de la structure DOM : Document Object Model



## 2.3 Les Styles avec CSS

Les CSS ou *Feuilles de Style en Cascade* permettent de définir le style d'affichage des portions de texte d'un document HTML en fonction des balises qui encadrent le texte, et donc de sa position dans la structure du DOM.

### Les Styles avec CSS : Cascading Style Sheets

```
<html>
  <head>
    <title> Math et Internet </title>
    <style type="text/css">
      ... ici les définitions des styles en CSS ...
    </style>
  </head>
  <body>
    <h1> Mathématiques </h1>
    <p> sur Internet... </p>
  </body>
</html>
```

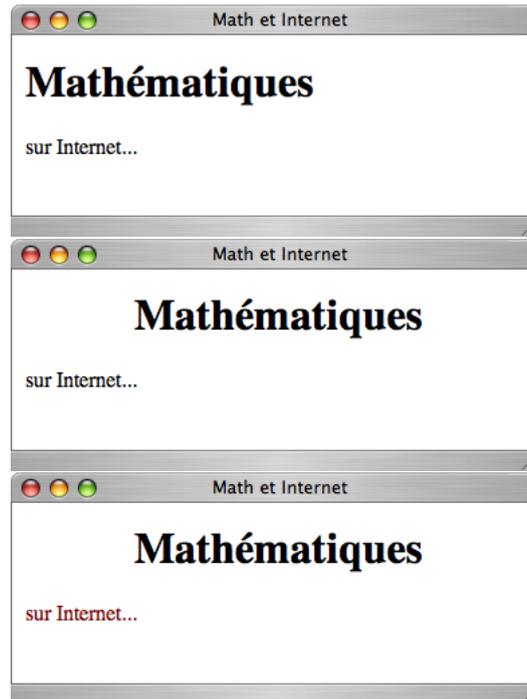
Le langage CSS permet de définir, pour chacun des éléments de la structure du document :

- la dimension ou la position
- les marges ou les espacements
- les encadrements éventuels
- la couleur des textes et des fonds
- le style des polices de caractères
- etc.

## 2.4 Modification de l'aspect visuel

### Modification de l'aspect visuel avec CSS

```
<style type="text/css">
  h1 { text-align : center; }
  p { color : maroon; }
</style>
```



Aspect original de la page HTML, à modifier ...

Nouvel aspect : centrage de l'élément `<h1>`

Aspect final : élément `<p>` en couleur

## 3 Représentation informatique d'une expression mathématique

### Représentation d'une expression mathématique

#### Objectif

- On dispose d'une formule mathématique :  $2 \times (a + b)$
- On veut représenter sa structure dans un Navigateur Internet

#### Méthode

- On va créer une description de cette formule en HTML
- On va ensuite contrôler le style de l'affichage avec CSS

### 3.1 Codage de l'expression mathématique en HTML

En HTML on dispose d'un certain nombre de balises prédéfinies comme `<html>`, `<head>`, `<body>`, `<div>`, `<p>`, `<h1>`...`<h6>`, etc.

Le texte d'un document HTML est, par défaut, affiché «au kilomètre», c'est à dire en flot continu, sans séparation.

La balise `<p>` permet de placer le texte dans un nouveau paragraphe séparé de ce qui précède par un petit espacement vertical.

La balise `<div>` permet simplement de diviser les données en blocks séparés qui sont affichées par défaut sur une nouvelle ligne.

Il y a de nombreuses façons de représenter une expression mathématique, nous allons choisir la balise `<div>` pour faire ça le plus simplement possible.

D'autres choix sont possibles et seraient tout aussi valables que celui-ci.

#### Codage de l'expression mathématique en HTML

$$2 \times (a + b) = \text{Produit}(2, \text{Somme}(a, b))$$

On décide de représenter toute expression par une `<div>` HTML

- qui contient le nom de la fonction, suivi des arguments
- ou l'argument, s'il est seul

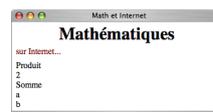
```
<div>
  Produit
  <div> 2 </div>
  <div> Somme <div> a </div> <div> b </div> </div>
</div>
```

### 3.2 Nouvelle structure du DOM

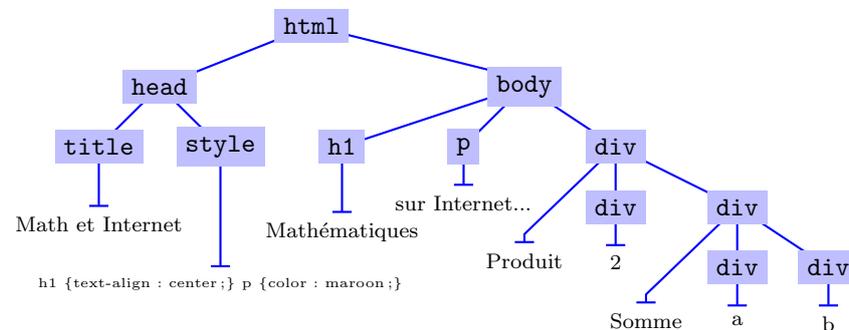
Ainsi chaque élément de l'expression mathématique sera simplement affiché en dessous du précédent, comme le veut le style par défaut de la balise `<div>`.

#### Nouvelle structure du DOM

On ajoute ce texte dans `<body>` :



... et on obtient cette nouvelle structure du DOM



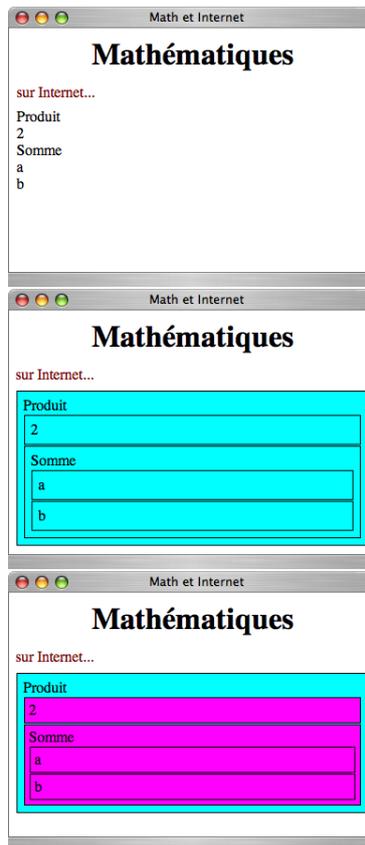
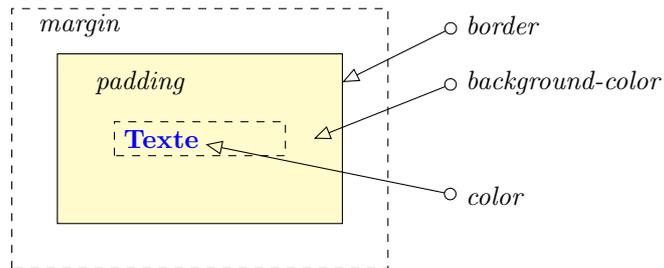
Il nous suffira alors de définir, en CSS, le mode d'affichage des balises `<div>` pour obtenir un résultat plus satisfaisant.

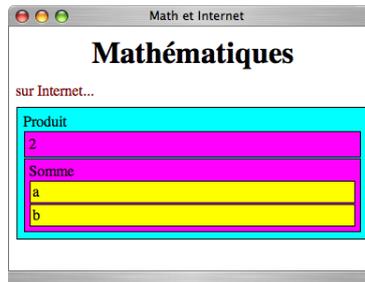
### 3.3 Définition des styles de présentation en CSS

La feuille de style CSS va permettre de modifier le mode de présentation par défaut de la balise `<div>`.

#### Définition des styles de présentation en CSS

Le texte de chaque élément de la structure du DOM est affiché dans un rectangle paramétrable :





```

<style type="text/css">
  h1 {text-align : center;}
  p {color : maroon;}
  div { margin : 1px; border : 1px black solid;
        padding : 6px; background-color : aqua; }
  div div {padding : 4px; background-color : fuchsia;}
  div div div {padding : 2px; background-color : yellow;}
</style>

```

On a ainsi défini, en cascade, les différents styles de la balise `<div>` en fonction de son niveau d'imbrication dans la structure.

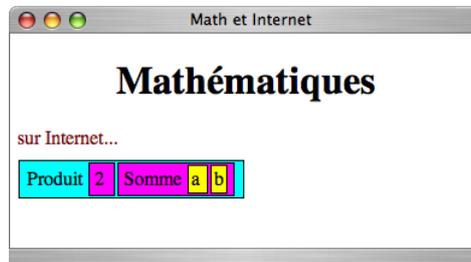
### 3.4 Modification de la présentation

Le langage CSS est assez simple, mais extrêmement riche et puissant.

Il permet des modifications de présentation impressionnantes sans toucher au contenu textuel du document HTML.

#### Modification de la présentation

On ajoute «`display : inline`» à la définition du style `<div>`



```

div { margin: 1px; border: 1px black solid;
      padding: 6px; background-color: aqua;
      display: inline; }

```

## 4 Programmation dans un Navigateur Internet

Autrefois, sur Internet, on chargeait des pages HTML simplement rédigées par des auteurs humains.

Aujourd'hui, de nombreuses pages ne sont que virtuelles, elle sont engendrées à la demande par des programmes informatiques.

Cette technique appelée DHTML pour *Dynamic HTML* permet d'adapter l'affichage des informations aux besoins du client, dynamiquement lors de la navigation sur les sites Internet.

La plupart des sites commerciaux sur Internet fonctionnent sur ce principe.

## Programmation d'un Navigateur Internet

Les pages HTML peuvent être engendrées par un programme.

### Séparation des tâches :

- La page chargée est définie en HTML
- Le style de la présentation est défini en CSS
- Ce qui est engendré par programme est codé en JavaScript

### Séparation des fichiers sources :

- Fichier : «`index.html`» la page HTML
- Fichier : «`style.css`» la feuille de style CSS
- Fichier : «`code.js`» le code JavaScript

Un autre avantage de cette technique de construction d'un site Internet est la séparation claire des différentes données et informations selon leur type.

On trouve, dans le cas le plus simple, trois fichiers différents :

- l'information : texte structuré mais sans style (HTML)
- le style : définition du mode d'affichage de l'information (CSS)
- l'interactivité : programme modifiant information ou style (JavaScript)

## 4.1 La page HTML : «`index.html`»

### La page HTML : «`index.html`»

```
<html>
<head>
  <title>Math et Internet</title>
  <link rel="stylesheet" href="style.css" type="text/css">
  <script type="text/javascript" src="code.js"></script>
</head>
<body id="page" onload="init()">
</body>
</html>
```

---

*Le corps du document est vide*

*l'entête contient un lien vers la feuille de style : «`style.css`»  
et vers le programme JavaScript : «`code.js`»*

*On définit un identificateur pour l'élément `<body>`  
... puis un appel à une fonction JavaScript*

## 4.2 La feuille de style CSS : «style.css»

### La feuille de style CSS : «style.css»

```
h1 { text-align: center; }
p { color: maroon; }

div {
  margin: 1px;
  padding: 6px;
  border: 1px black solid;
  background-color: aqua;
}
div div { padding: 4px; background-color: fuchsia; }
div div div { padding: 2px; background-color: yellow; }
```

---

On a simplement déplacé le contenu de la balise `<style>` vue dans l'exemple précédent vers le fichier `style.css`

## 4.3 Le programme JavaScript : «code.js»

### Le programme JavaScript : «code.js»

```
function init() {
  var dom_body = document.getElementById("page");
  dom_body.innerHTML = "<h1>Mathématiques</h1>";
  dom_body.innerHTML += "<p>sur Internet...</p>";
}
```

---

Le code JavaScript ne contient que la définition de `init()` appelée une seule fois, lors du chargement de la page `index.html` dans le Navigateur Internet  
*On récupère l'élément `<body>` du DOM à l'aide de `id="page"`  
La propriété `innerHTML` permet de modifier dynamiquement le contenu de l'élément `<body>`  
Ce qui affichera la page de départ originale.*

Une solution simple est d'engendrer par programme le code HTML qui représentera l'expression mathématique de façon abstraite sous forme de `<div>`.

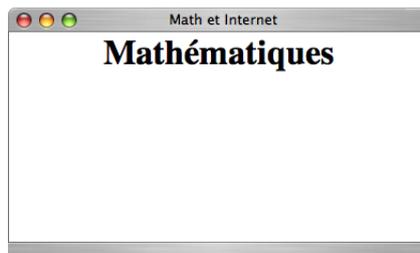
### Comment engendrer une expression mathématique ?

```
function expr(argument) {
    return "<div>" + argument + "</div>";
}

function init() {
    var dom_body = document.getElementById("page");
    dom_body.innerHTML = "<h1>Mathématiques</h1>";
    dom_body.innerHTML += expr("Somme" + expr("a") + expr("b"));
}
```

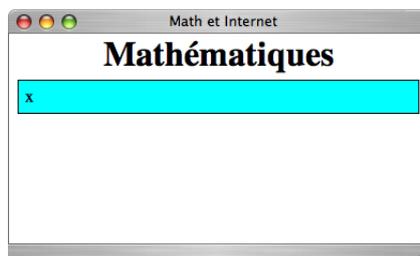
On va définir une fonction «`expr()`» qui construit l'élément `<div>` qui représente l'expression et qui devra contenir, le nom de la fonction suivi de la liste de chacun de ses arguments ... voici l'appel qui permettra par exemple de construire l'expression « $a + b$ ».

Selon la définition de `init()` on obtient :



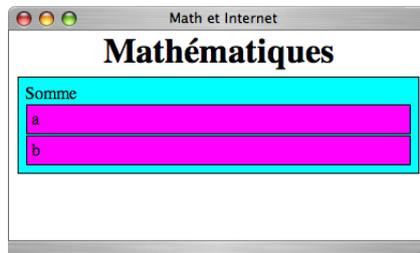
```
function init() {
    var dom_body = document.getElementById("page");
    dom_body.innerHTML = "<h1>Mathématiques</h1>";
}
```

... ou simplement l'expression  $x$



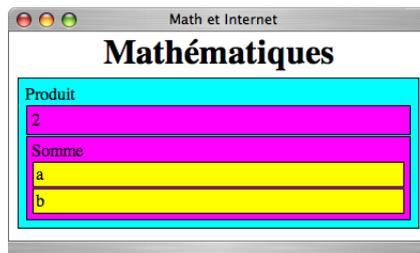
```
function init() {
    var dom_body = document.getElementById("page");
    dom_body.innerHTML = "<h1>Mathématiques</h1>";
    dom_body.innerHTML += expr("x");
}
```

... ou l'expression  $a + b$



```
function init() {  
  var dom_body = document.getElementById("page");  
  dom_body.innerHTML = "<h1>Mathématiques</h1>";  
  dom_body.innerHTML += expr("Somme" + expr("a") + expr("b"));  
}
```

... ou encore l'expression  $2 \times (a + b)$



```
function init() {  
  var dom_body = document.getElementById("page");  
  dom_body.innerHTML = "<h1>Mathématiques</h1>";  
  dom_body.innerHTML += expr("Produit" + expr("2")  
    + expr("Somme" + expr("a") + expr("b")));  
}
```

Pour simplifier, cet exemple au maximum, il est statique.

À l'ouverture de la page, le programme affiche immédiatement l'expression définie, et c'est tout.

Il est possible d'engendrer une page contenant des boutons, des menus ou champs de saisie de texte que le lecteur pourra modifier pour définir dynamiquement de nouvelles expressions mathématiques.

## 4.4 C'est simple : vous pouvez le faire !

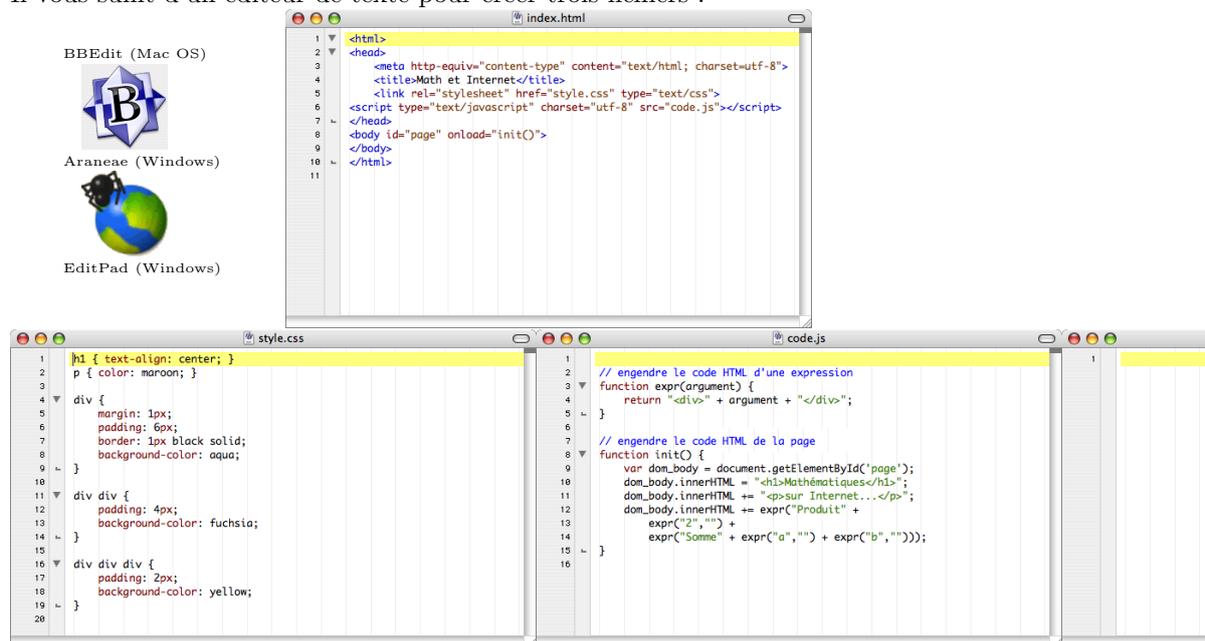
Sur un ordinateur, tout le monde dispose d'au moins un éditeur de texte et un navigateur Internet.

**Donc :** tout le monde peut réaliser le petit exemple proposé ci-dessus.

Cet exemple a volontairement été conçu pour être le plus simple possible. Ceci dans le but d'être effectivement réalisable, même par un débutant.

### C'est simple : vous pouvez le faire !

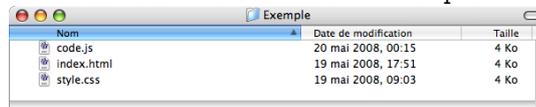
- Il vous suffit d'un éditeur de texte pour créer trois fichiers :



- «index.html» la page HTML
- «style.css» la feuille de style CSS
- «code.js» le code JavaScript

### C'est fait : vous pouvez le tester !

- Placez ces 3 fichiers dans un même dossier : «Exemple»



- Ouvrez alors «index.html» dans un Navigateur Internet.



Safari (Mac)



Firefox



Internet Explorer (Windows)

### Ensuite... donnez libre cours à votre imagination...

Dès qu'on a réussi à faire fonctionner ce petit exemple, on peut chercher sur Internet des informations pour continuer. De très nombreuses documentations sur HTML, CSS et JavaScript sont disponibles en ligne.

**Pour plus d'informations :**

Vous trouverez les exemples présentés dans cet exposé sur le site :

<http://math.et.info.free.fr/>

... ainsi que d'autres exemples et des informations complémentaires

Vous trouverez aussi de nombreux problèmes de mathématiques et des exemples informatiques sur le site de l'atelier :

<http://ateliermathematique.free.fr/>



On trouvera de nombreuses informations sur les sites indiqués ci-dessus.

Il est aussi possible de me contacter, pour demander éventuellement des informations complémentaires, en remplissant les formulaires disponibles sur chacun de ces sites.