

# Arbres Mathématiques Informatique et Navigateurs Internet

Jacques Duma

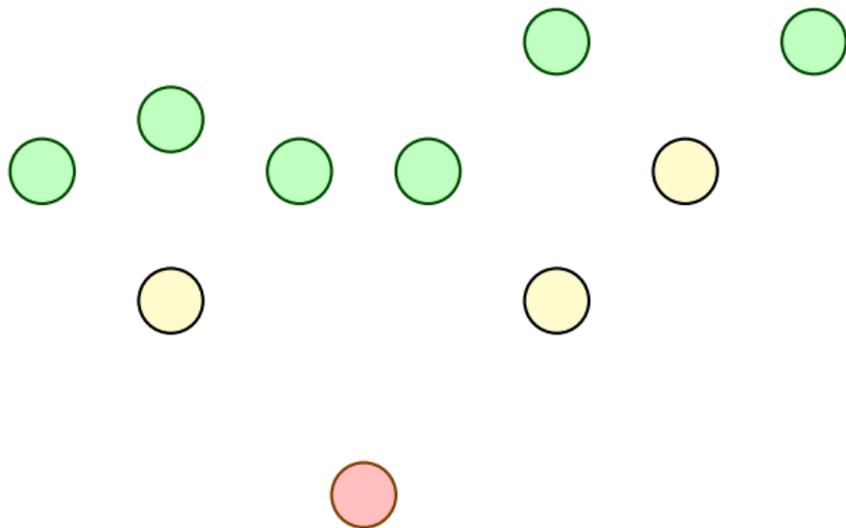


<http://math.et.info.free.fr/>  
<http://ateliermathematique.free.fr/>

# Arbre

Un arbre est un graphe connexe, sans cycle

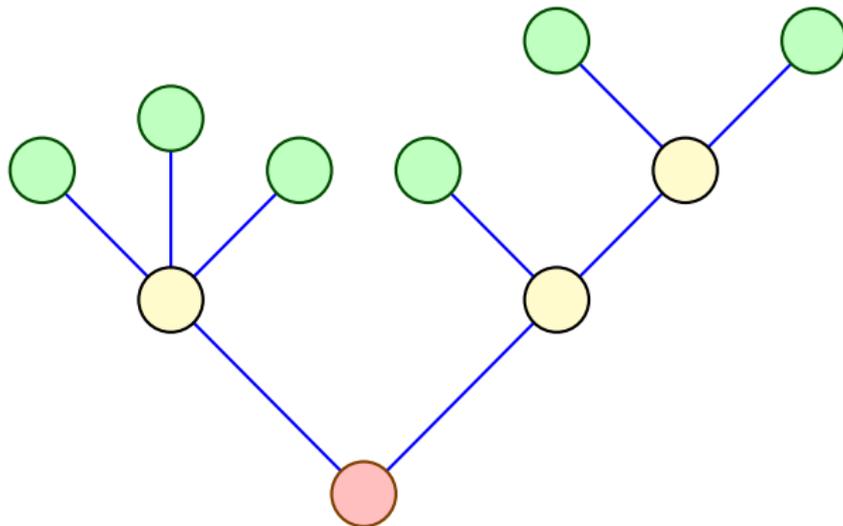
C'est un ensemble de sommets ou nœuds (*des points*)



# Arbre

Un arbre est un graphe connexe, sans cycle

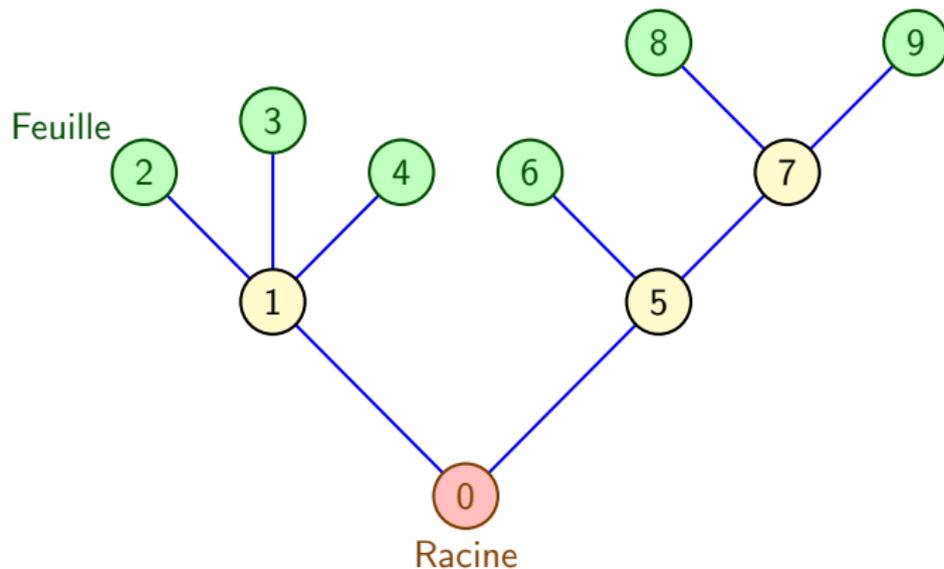
C'est un ensemble de sommets ou nœuds (*des points*)  
reliés par des arêtes ou branches (*des segments*)



# Arbre

Un arbre est un graphe connexe, sans cycle

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



# Différentes formes d'écriture d'une expression

## Notation usuelle

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

## Notation fonctionnelle

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

# Structure arborescente de l'expression

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

# Structure arborescente de l'expression

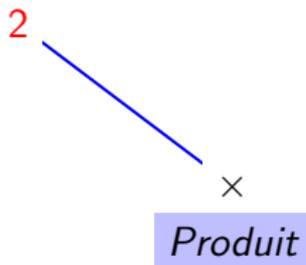
$$2 \times (a + b) = \textit{Produit}(2, \textit{Somme}(a, b))$$

×

*Produit*

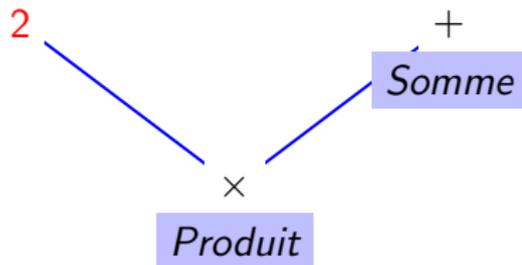
# Structure arborescente de l'expression

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



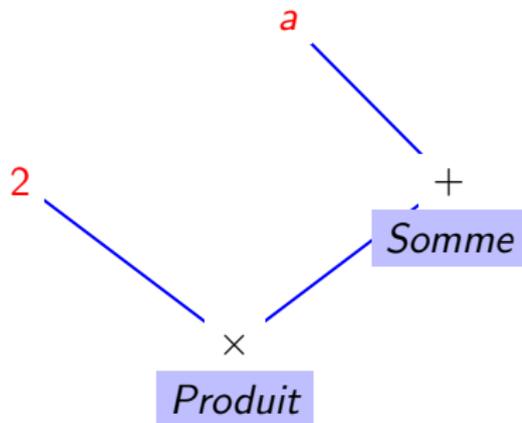
# Structure arborescente de l'expression

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



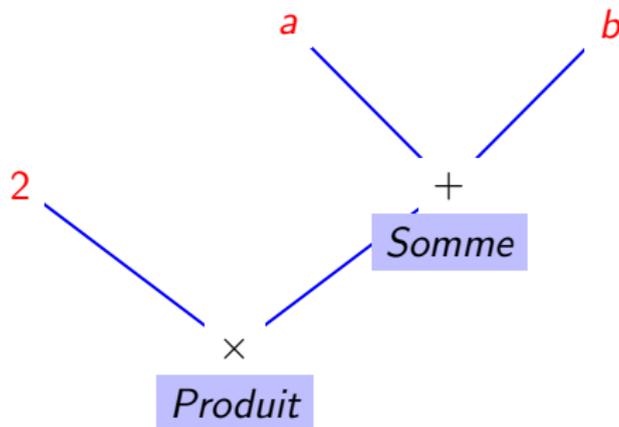
# Structure arborescente de l'expression

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



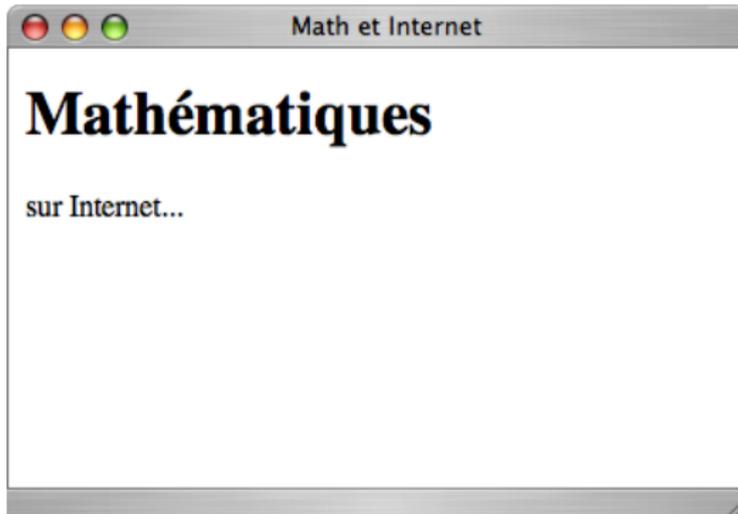
# Structure arborescente de l'expression

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



# Aspect visuel

Une page HTML affichée dans la fenêtre de Navigateur Internet



# Code source HTML : HyperText Markup Language

```
<html>
```

```
</html>
```

# Code source HTML : HyperText Markup Language

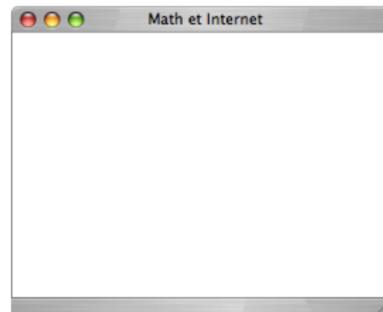
```
<html>  
  <head>  
  
  </head>  
  <body>  
  
  </body>  
</html>
```



# Code source HTML : HyperText Markup Language

```
<html>
  <head>
    <title>Math et Internet</title>
  </head>
  <body>

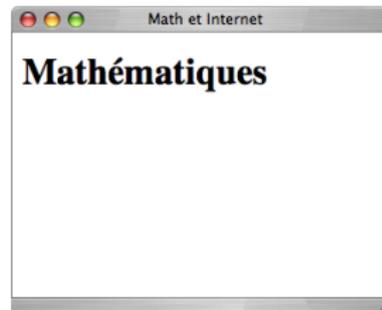
  </body>
</html>
```



# Code source HTML : HyperText Markup Language

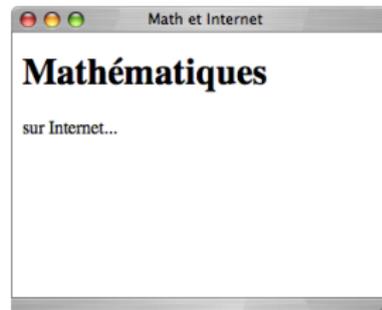
```
<html>
  <head>
    <title>Math et Internet</title>
  </head>
  <body>
    <h1>Mathématiques</h1>

  </body>
</html>
```



# Code source HTML : HyperText Markup Language

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

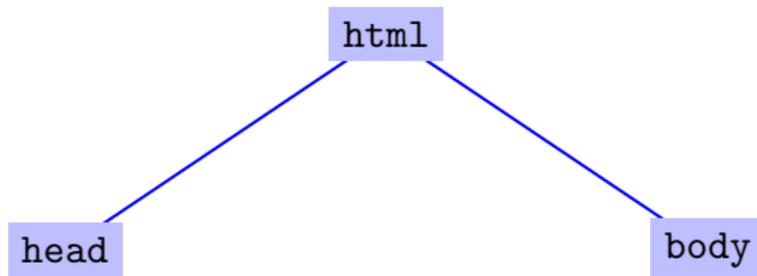


# Arbre de la structure DOM : Document Object Model

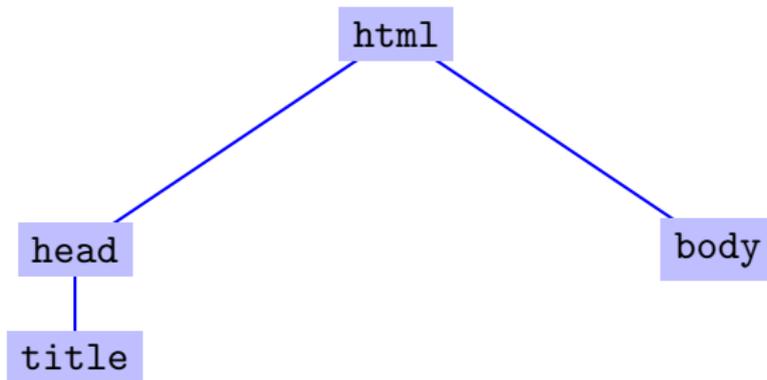
# Arbre de la structure DOM : Document Object Model

```
html
```

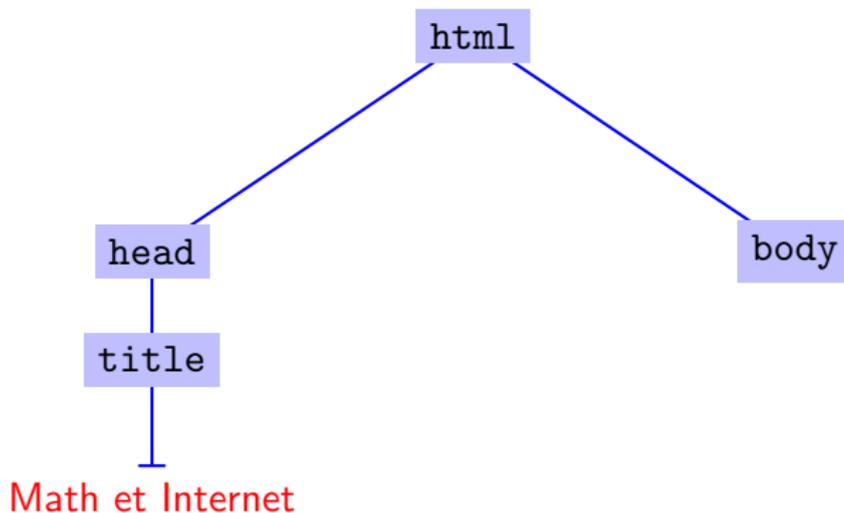
# Arbre de la structure DOM : Document Object Model



# Arbre de la structure DOM : Document Object Model

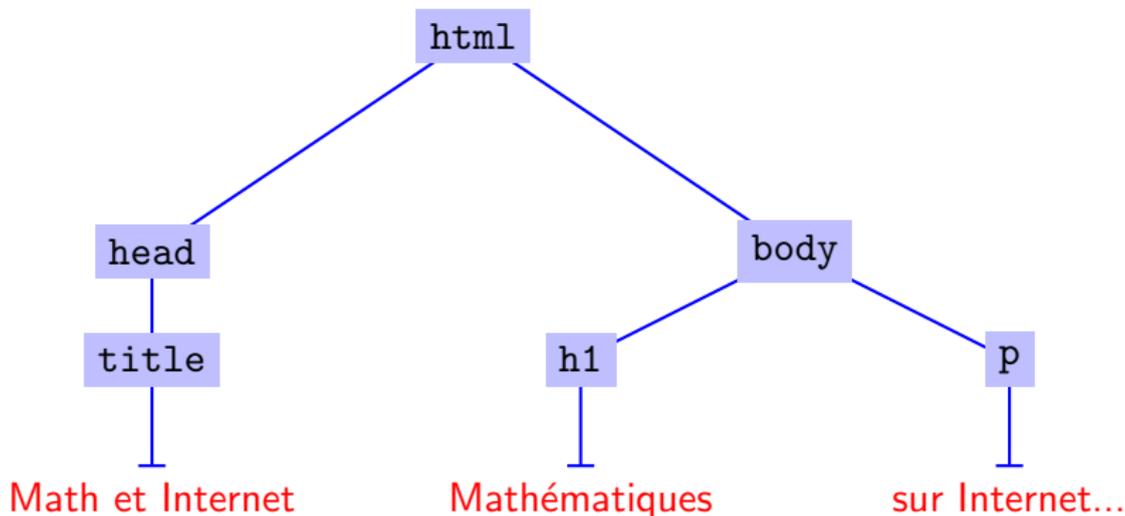


# Arbre de la structure DOM : Document Object Model





# Arbre de la structure DOM : Document Object Model



# Les Styles avec CSS : Cascading Style Sheets

```
<html>
  <head>
    <title> Math et Internet </title>
    <style type="text/css">

      </style>
  </head>
  <body>
    <h1> Mathématiques </h1>
    <p> sur Internet... </p>
  </body>
</html>
```

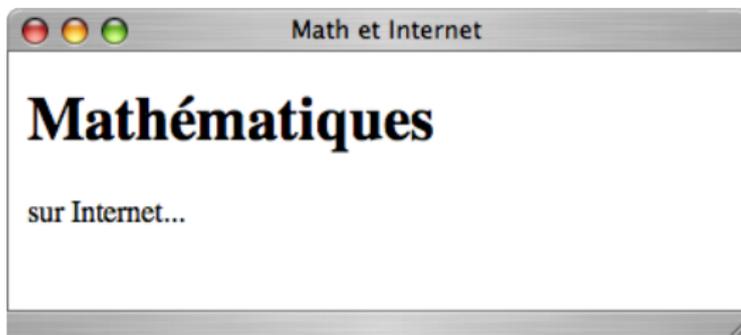
# Les Styles avec CSS : Cascading Style Sheets

```
<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>
```

# Modification de l'aspect visuel avec CSS

```
<style type="text/css">
```

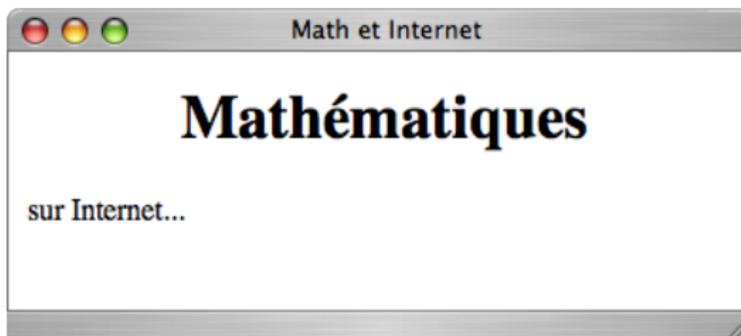
```
</style>
```



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

# Modification de l'aspect visuel avec CSS

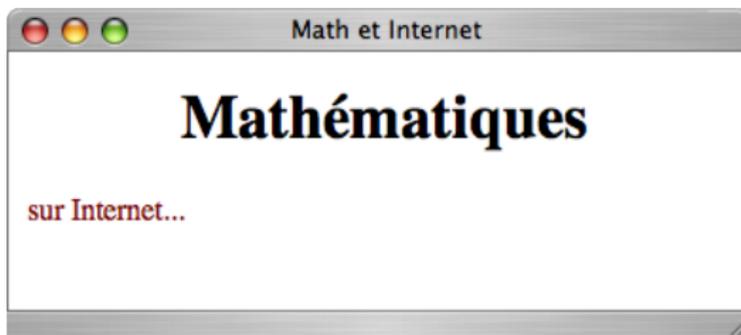
```
<style type="text/css">  
  h1 { text-align : center; }  
  
</style>
```



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

# Modification de l'aspect visuel avec CSS

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



Aspect final : élément `<p>` en couleur

# 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

# 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

# Codage de l'expression mathématique en HTML

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

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

`<div>`

`</div>`

# Codage de l'expression mathématique en HTML

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

On décide de représenter toute expression par une `<div>` HTML  
- qui contient le nom de la fonction, suivi des arguments

```
<div>  
  Produit  
  <div></div>  
  <div></div>  
</div>
```

# Codage de l'expression mathématique en HTML

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

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></div> <div></div> </div>
```

```
</div>
```

# Codage de l'expression mathématique en HTML

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

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
```

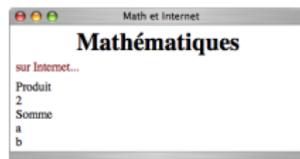
```
  <div> 2 </div>
```

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

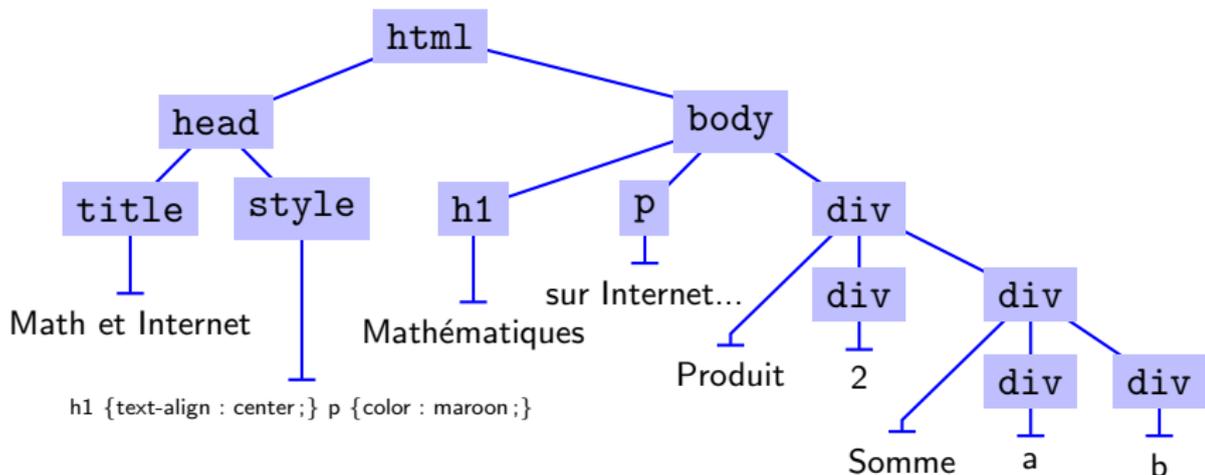
```
</div>
```

# Nouvelle structure du DOM

On ajoute ce texte dans `<body>` :

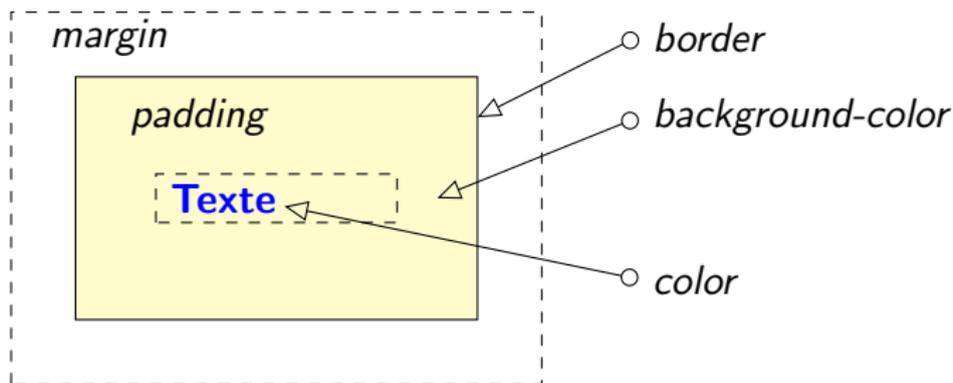


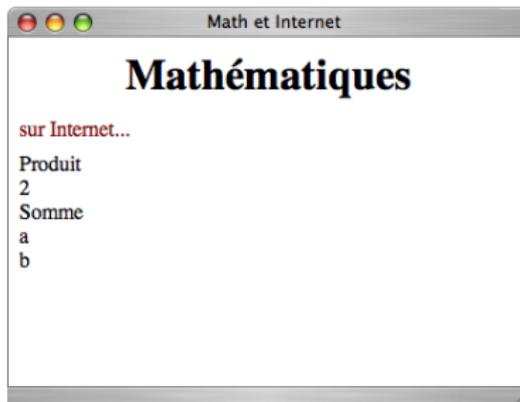
... et on obtient cette nouvelle structure du DOM



# Définition des styles de présentation en CSS

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



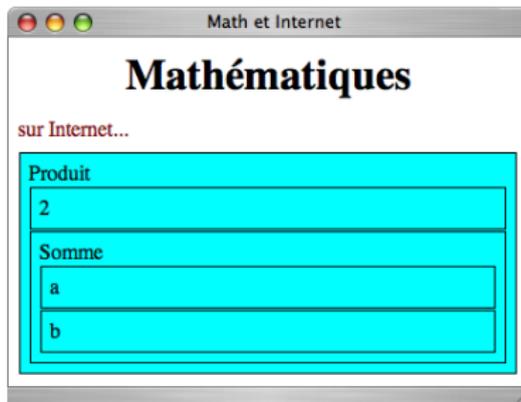


```
<style type="text/css">
```

```
  h1 {text-align : center;}
```

```
  p {color : maroon;}
```

```
</style>
```

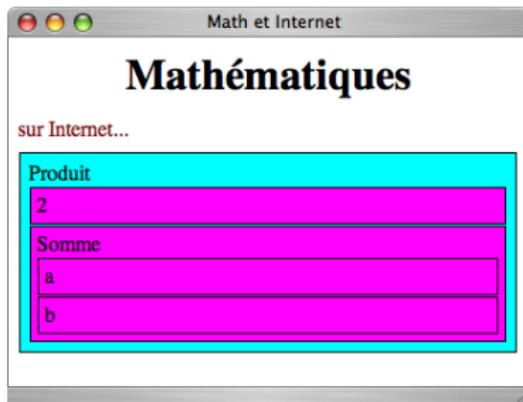


```

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

</style>

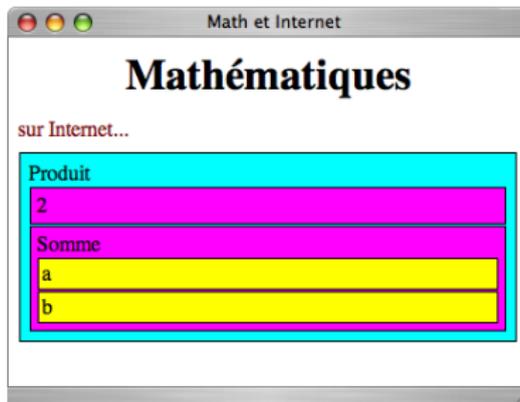
```



```

<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;}
</style>

```



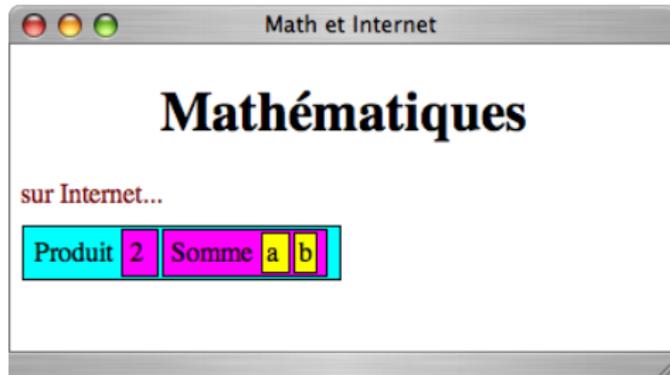
```

<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>

```

# 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; }
```

# Programmation d'un Navigateur Internet

Les pages HTML peuvent être engendrées par un programme.

# 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

# 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

# La page HTML : «index.html»

```
<html>
  <head>
    <title>Math et Internet</title>

  </head>
  <body>
</body>
</html>
```

---

Le corps du document est vide

# La page HTML : «index.html»

```
<html>
  <head>
    <title>Math et Internet</title>
    <link rel="stylesheet" href="style.css" type="text/css">

  </head>
  <body>
</body>
</html>
```

---

Le corps du document est vide

l'entête contient un lien vers la feuille de style : «style.css»

# 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>
  </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»

# 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">
  </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>`

# 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

# 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`

# Le programme JavaScript : «code.js»

```
function init() {  
  
  
  
  
  
  
  
  
  
}
```

---

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

# Le programme JavaScript : «code.js»

```
function init() {  
    var dom_body = document.getElementById("page");  
  
}
```

---

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"`

# 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>`.

# 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.

# Comment engendrer une expression mathématique ?

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

---

# 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>";  
}
```

---

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

# Comment engendrer une expression mathématique ?

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

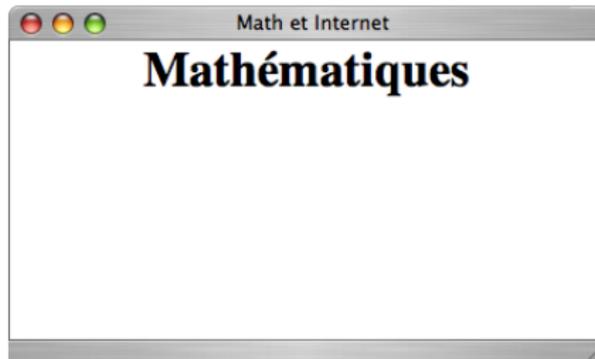
```
function init() {  
    var dom_body = document.getElementById("page");  
    dom_body.innerHTML = "<h1>Mathématiques</h1>";  
    dom_body.innerHTML += expr("Somme" + expr("a") + expr("b"));  
}
```

---

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

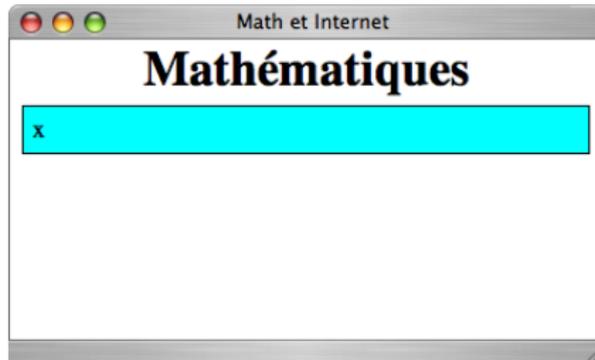
... voici l'appel qui permettra par exemple de construire l'expression « $a + b$ ».

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



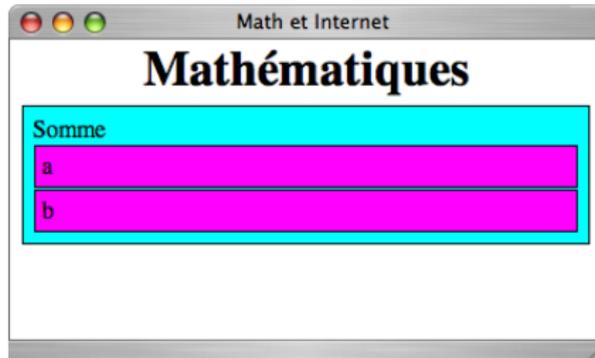
```
function init() {  
    var dom_body = document.getElementById("page");  
    dom_body.innerHTML = "<h1>Mathématiques</h1>";  
}
```

... ou simplement l'expression X



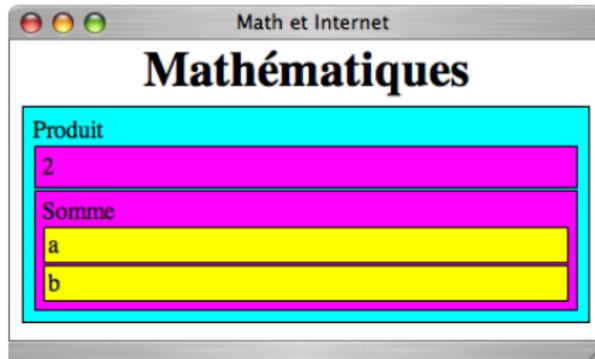
```
function init() {
  var dom_body = document.getElementById("page");
  dom_body.innerHTML = "<h1>Mathématiques</h1>";
  dom_body.innerHTML += expr("x");
}
```

... 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"))));
}
```

# C'est simple : vous pouvez le faire!

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

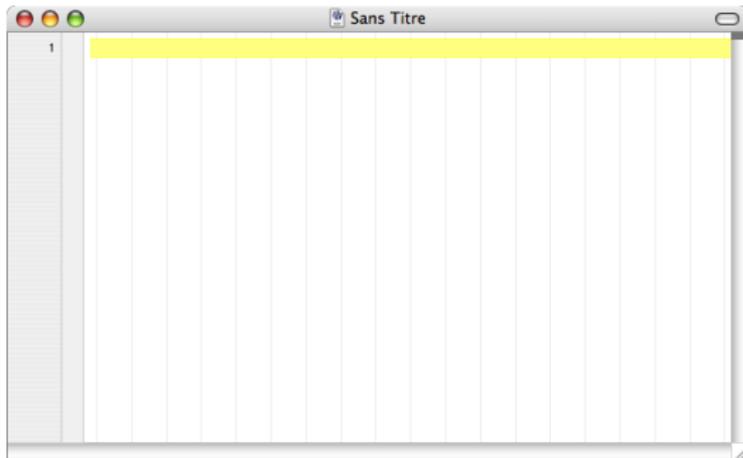
BEdit (Mac OS)



Araneae (Windows)



EditPad (Windows)



# C'est simple : vous pouvez le faire !

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

BEdit (Mac OS)



Araneae (Windows)



EditPad (Windows)

```

1  <html>
2  <head>
3    <meta http-equiv="content-type" content="text/html; charset=utf-8">
4    <title>Math et Internet</title>
5    <link rel="stylesheet" href="style.css" type="text/css">
6    <script type="text/javascript" charset="utf-8" src="code.js"></script>
7  </head>
8  <body id="page" onload="init()">
9  </body>
10 </html>
11

```

- ▶ «index.html» la page HTML

# C'est simple : vous pouvez le faire !

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

BEdit (Mac OS)



Araneae (Windows)



EditPad (Windows)

```

1 |h1 { text-align: center; }
2 |p { color: maroon; }
3 |
4 |div {
5 |    margin: 1px;
6 |    padding: 6px;
7 |    border: 1px black solid;
8 |    background-color: aqua;
9 |}
10|
11|div div {
12|    padding: 4px;
13|    background-color: fuchsia;
14|}
15|
16|div div div {
17|    padding: 2px;
18|    background-color: yellow;
19|}
20|

```

- ▶ «index.html» la page HTML
- ▶ «style.css» la feuille de style CSS

# C'est simple : vous pouvez le faire!

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

BEdit (Mac OS)



Araneae (Windows)



EditPad (Windows)

```

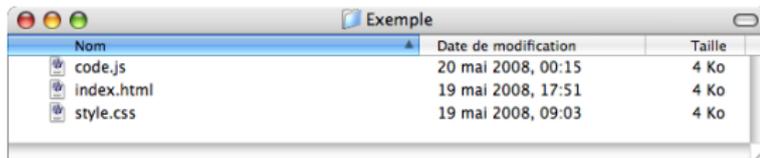
1
2 // engendre le code HTML d'une expression
3 function expr(argument) {
4     return "<div>" + argument + "</div>";
5 }
6
7 // engendre le code HTML de la page
8 function init() {
9     var dom_body = document.getElementById('page');
10    dom_body.innerHTML = "<h1>Mathématiques</h1>";
11    dom_body.innerHTML += "<p>sur Internet...</p>";
12    dom_body.innerHTML += expr("Produit" +
13        expr("2", "") +
14        expr("Somme" + expr("a", "") + expr("b", "")));
15 }
16

```

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

# C'est fait : vous pouvez le tester !

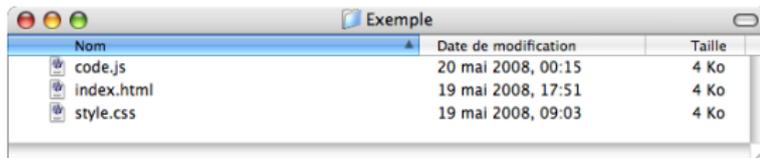
- ▶ Placez ces 3 fichiers dans un même dossier : «Exemple»



Nom	Date de modification	Taille
code.js	20 mai 2008, 00:15	4 Ko
index.html	19 mai 2008, 17:51	4 Ko
style.css	19 mai 2008, 09:03	4 Ko

# 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)



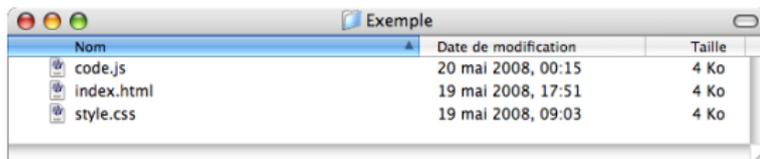
Firefox



Internet Explorer (Windows)

# 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...

## 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/>

