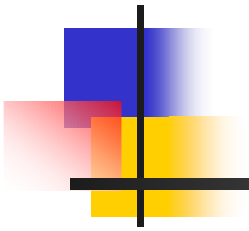
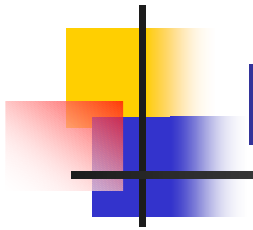


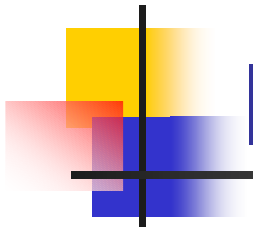
Javascript

Programmation objet



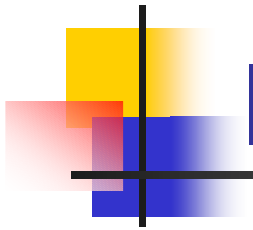
Introduction

- Programmation objet
  - Elle consiste à modéliser informatiquement un concept du monde réel en entités informatiques
  - Ces entités informatiques sont appelées **objets**
  - Exemple : compteBancaire
    - Propriétaire
    - Montant
    - Opérations possibles : débit, crédit…



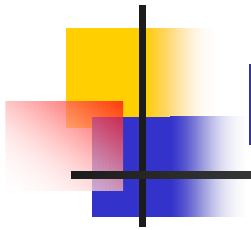
Introduction

- Un objet
  - est caractérisé par plusieurs notions :
  - **L'identité** :
    - Nom qui le distingue
  - **Les attributs** :
    - Données qui le caractérisent
    - Variables stockant des informations d'état de l'objet
  - **Les méthodes** (appelées parfois *fonctions membres*) :
    - caractérisent son comportement
    - c'est-à-dire l'ensemble des actions (appelées *opérations*) que l'objet est à même de réaliser



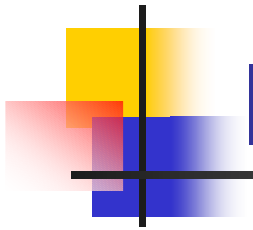
Introduction

- Notion de classe
  - On appelle **classe** la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui composeront un objet
  - Un objet est donc "issu" d'une classe, c'est le produit qui sort d'un moule
- Une classe est composée de deux parties :
  - **Les attributs** (parfois appelés *données membres*) :
    - il s'agit des données représentant l'état de l'objet
  - **Les méthodes** (parfois appelées *fonctions membres*) :
    - il s'agit des opérations applicables aux objets



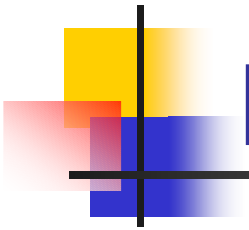
Introduction

- Exemple
  - Si on définit la classe *voiture*
    - les objets *Peugeot 406*, *Renault 18* seront des instanciations de cette classe
    - Il pourra éventuellement exister plusieurs objets *Peugeot 406*, différenciés par leur numéro de série
    - Mieux :
      - Deux instanciations de classes pourront avoir tous leurs attributs égaux sans pour autant être un seul et même objet
      - C'est le cas dans le monde réel
        - deux T-shirts peuvent être strictement identiques et pourtant ils sont distincts



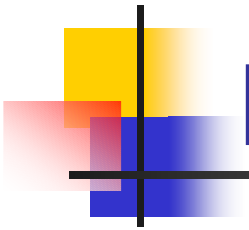
Introduction

- Notion d’objet en Javascript
  - Le Javascript traite les éléments qui s'affichent dans votre navigateur comme des objets, c'est-à-dire des éléments :
    - classés selon une hiérarchie pour pouvoir les désigner précisément
    - auxquels des propriétés et des actions (méthodes) sont associées



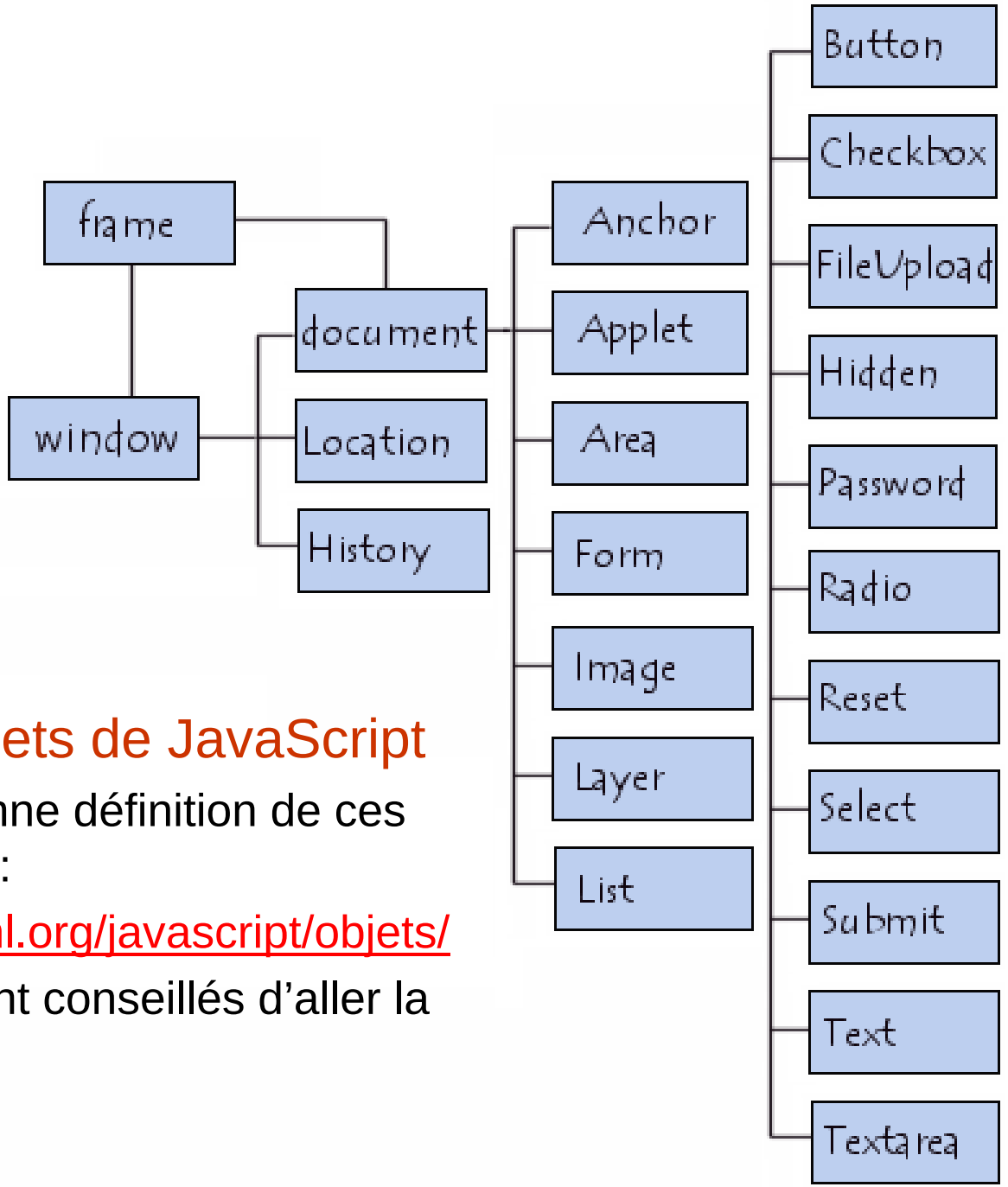
Les objets

- Comment JavaScript définit les objets ?
  - Javascript divise la page du navigateur en éléments (objets), afin de permettre d'accéder à n'importe lequel d'entre-eux et de pouvoir les manipuler par l'intermédiaire de leurs propriétés
  - On commence généralement par l'objet le plus grand (celui contenant tous les autres) puis on descend dans la hiérarchie jusqu'à arriver à l'objet voulu
    - L'objet le plus grand est l'objet fenêtre : ***window***
    - Dans la fenêtre s'affiche une page, c'est l'objet ***document***
    - Cette page peut contenir plusieurs objets, comme
      - des formulaires,
      - des images, etc.



Les objets

- Les objets de base de JavaScript
  - navigator : qui contient des informations sur le navigateur de celui qui visite la page
  - window : c'est l'objet où s'affiche la page, il contient donc des propriétés concernant la fenêtre elle-même mais aussi tous les objets-enfants contenus dans celle-ci
  - location : contient des informations relatives à l'adresse de la page à l'écran
  - history: c'est l'historique, c'est-à-dire la liste de liens qui ont été visités précédemment
  - document : il contient les propriétés sur le contenu du document (couleur d'arrière plan, titre, ...)



- Hiérarchie des objets de JavaScript
  - On trouve une bonne définition de ces objets à l'adresse :
    - http://fr.selfhtml.org/javascript/objets/
  - Vous êtes vivement conseillés d'aller la consulter

Les objets

- L'objet Navigator
  - a plusieurs propriétés concernant votre navigateur
    - *appName :*
      - *connaître le nom : Netscape, IE, Mozilla*
    - *appVersion :*
      - *connaître la version*
    - *language :*
      - *FR, AN*
    - *platform :*
      - *windows, Linux…*
  - *Pour le savoir : exécuter :*

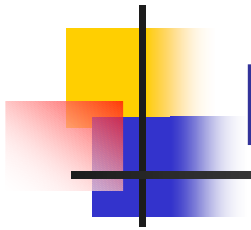
```
<script language="Javascript">
   document.write(navigator.propriété);
</script>
```

  - *Exemple : cours4-exemples/navigateur1.html*

Les objets

- *Exemple d'utilisation de Navigator : navigateur2.html*

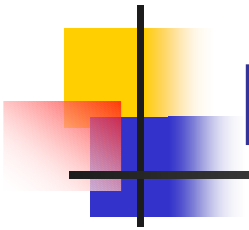
```
Nom = navigator.appName;
if (Nom == 'Netscape') {
    placer ici les instructions à exécuter s'il s'agit de
   Netscape Navigator 4 ou supérieur }
 if (Nom == 'Microsoft Internet Explorer') {
placer ici les instructions à exécuter s'il s'agit de
   Microsoft Internet Explorer 4 ou supérieur
}
```



Les objets

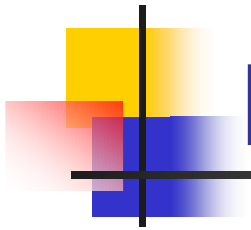
- *Autre exemple d'utilisation de Navigator :* navigateur3.html

```
<html>
<head><title>Test</title>
</head>
<body>
<script type="text/javascript">
   if(navigator.language.indexOf("en")>-1) document.write("dear
   visitor, welcome on our pages");
   if(navigator.language.indexOf("fr")>-1) document.write("Cher
   visiteur, soyez le bienvenu sur nos pages");
</script>
</body>
</html>
```



Les objets

- L’objet Window
  - est l'objet par excellence dans Javascript, car il est le parent de chaque objet qui compose la page web
  - il contient donc :
    - l'objet document :
      - la page en elle-même
    - l'objet location :
      - le lieu de stockage de la page
    - l'objet history :
      - les pages visitées précédemment
    - l'objet frames :
      - les cadres (division de la fenêtre en sous-fenêtres)



Les objets

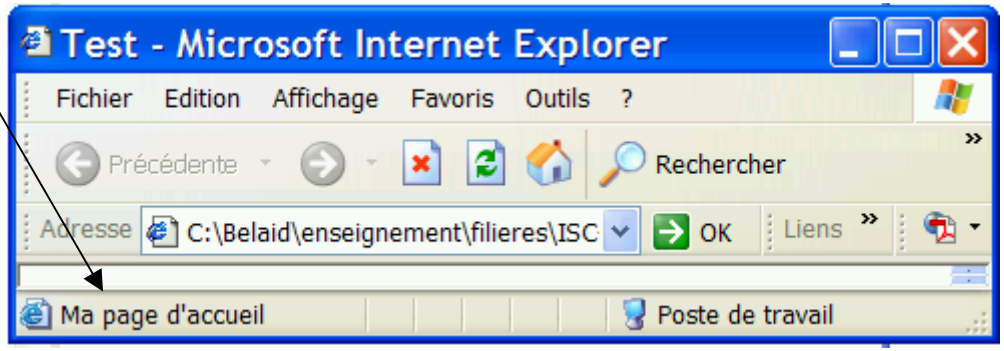
- L'objet Window
  - Propriétés :
    - Frames[] : tableau de cadres contenus
    - Length : nombre de cadres (nombre d'éléments du tableau *frames*
    - Name : nom de la fenêtre dans laquelle on se trouve
    - Parent : fenêtre qui englobe celle dans laquelle on se trouve
  - Méthodes :
    - alert(), confirm() et prompt() : font apparaître une boîte de dialogue
    - open(), et close() : permettent l'ouverture et la fermeture de fenêtres
  - Plusieurs exemples sous :

    http://fr.selfhtml.org/javascript/objets/window.htm

Les objets

- L'objet Window : Propriété defaultStatus
  - affiche dans la barre d'état de la fenêtre d'affichage la valeur "Ma page d'accueil"
  - Exemple : window0.html

```
<html>
<head>
<title>Test</title>
<script type="text/javascript">
   window.defaultStatus = "Ma page d'accueil";
</script>
</head>
<body>
</body>
</html>
```



http://fr.selfhtml.org/javascript/objets/window.htm#closed

Les objets

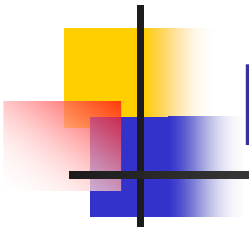
- L'objet Window : Propriétés innerHeight, innerwidth
  - permettent de fixer au cours de l'exécution la hauteur et la largeur de la fenêtre

```
<html>
<head><title>Test</title>
<script type="text/javascript">
   window.innerHeight = 300;
</script>
</head>
<body>
</body>
</html>
```

Les objets

- L'objet Window : la méthode open () :
  - Cette fonction ouvre une nouvelle fenêtre, voici sa syntaxe :
    - window.open("URL","nom_de_la_fenetre","options_de_la_fenetre")
      - Ouvre « secondefenetre » et y affiche le fichier test.html. Secondefenetre peut être utilisé comme target pour l'affichage de l'extérieur
  - Exemple : window01.html

```
<html>
<head>
<title>Test</title>
<script type="text/javascript">
   function nouvellefenetre() {
            mafenetre = window.open("window0.html",
            "secondefenetre", "width=300,height=200,scrollbars"); }
</script> </head>
<body>
<a href="javascript:nouvellefenetre()"> nouvelle fenêtre </a>
</body></html>
```



Les objets

- L’objet Window : la méthode close () :

```
<a href="javascript:mafenetre.close()">fermer la
  fenêtre</a>
```

  - En cliquant sur ce lien, cela ferme la fenêtre précédemment ouverte avec le nom « mafenetre »

Les objets

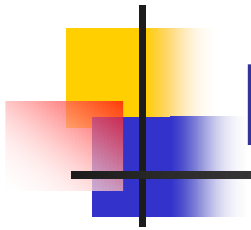
- L'objet Window
  - Vérification de l'état de la fenêtre
  - Exemple : window2.html

```
<html><head><title>Test</title>
<script type="text/javascript">
<!--
    var InfoWin = window.open("fichier1.htm", "secondefenetre");
    function CheckOpen() {
    if(InfoWin.closed == true) alert("La fenêtre a été fermée"); else
    alert("La fenêtre est encore ouverte"); }
//-->
</script>
</head>
    <body>
            <a href="javascript:CheckOpen()">La fenêtre est-elle
            fermée?</a>
    </body>
</html>
```

Les objets

- L’objet Window
  - Fermeture automatique d’une fenêtre, après 2’

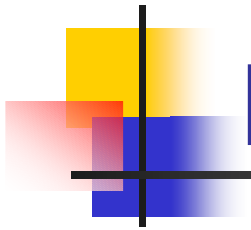
```
<html>
    <head>
        <title>Test</title>
        <script type="text/javascript">
            var InfoWin = window.open("exercice1.html",
            "secondefenetre");
          InfoWin.setTimeout("top.close()",2000);
        </script>
</head>
   <body>
   </body>
</html>
```



Les objets

- L’objet document
  - L'objet document est un élément majeur, il va vous permettre de récupérer des informations d'un formulaire, créer des calques et les déplacer, écrire du texte...
  - Propriétés :
    - document.fgColor, permet de récupérer et de changer la couleur du texte de votre page HTML

      document.fgColor = "#993333";
    - document.bgColor, permet de récupérer et de changer la couleur de fond de votre page HTML
    - document.lastModified, permet de savoir quand la page html a été modifiée
      - document.lastModified;
        - Internet explorer renvoie : 11/07/2000 19:41:00
          Netscape renvoie : Tuesday, November, 7 /2000 19:41:00

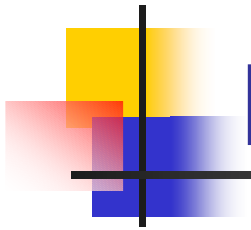


Les objets

- document.linkColor
  - permet de récupérer et de changer la couleur des liens de votre page HTML
- document.location
  - permet de récupérer et changer l'url de votre page HTML, ce qui revient à charger une autre page HTML

```
document.location = "URL/monDoc.HTML";
```

- document.write()
  - permet d'écrire dans votre page HTML
- document.images[ ]
  - permet de récupérer et changer les images de votre page HTML
- document.forms[ ]
  - permet de récupérer et changer les informations de votre formulaire



Les objets

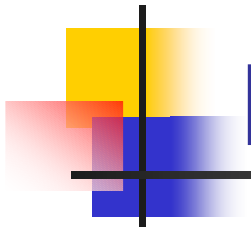
- L’objet document :
  - Plusieurs exemples

    http://fr.selfhtml.org/javascript/objets/document.htm

Les objets

- L'objet document : document1.html
  - L'exemple fait un fondu enchaîné depuis le noir en passant par les nuances de gris jusqu'au blanc
  - La fonction setColor() est appelée avec à chaque fois un délai de 20 millièmes de seconde grâce à setTimeout

```
<script type="text/javascript">
var X = new
    Array("0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F");
var x1 = 0, x2 = 0;
function setColor() {
    document.bgColor ="#" + X[x1] + X[x2] + X[x1] + X[x2] + X[x1] +
    X[x2]; x2 = x2 + 1;
    if(x2 % 16 == 0) { x2 = 0; x1 = x1 + 1; }
}
for(var i = 0; i < 255; ++i)
    { window.setTimeout("setColor()",20); }
</script>
```



Les objets

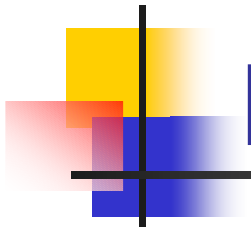
- L’objet document : document2.html
  - Donne la date de la dernière modification du document

```
<html>
<head><title>Test</title></head>
<body>
  <script type="text/javascript">
        document.write("dernière mise à jour: " +
        document.lastModified);
  </script>
</body>
</html>
```

Les objets

- L'objet document : document3.html
  - Permet de récupérer le contenu de la balise <title>

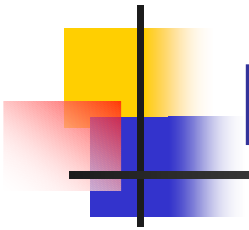
```
<html>
  <head>
        <title>Test</title>
  </head>
  <body>
        <h1>
        <script type="text/javascript">
        <!--
              document.write(document.title);
        //-->
        </script>
        </h1>
  </body>
</html>
```



Les objets

- L'objet document : document4.html
  - Permet de récupérer l'URL où se trouve le document

```
<html>
  <head>
        <title>Test</title>
  </head>
  <body>
        <script type="text/javascript">
        <!--
        document.write("Ce fichier: " + document.URL);
        //-->
        </script>
  </body>
</html>
```



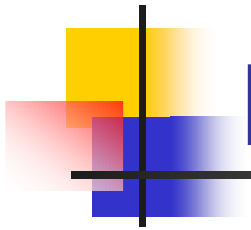
Le formulaire

- L’objet : forms
  - Avec l'objet forms, qui se trouve sous l'objet **document** dans la hiérarchie JavaScript, vous avez accès aux formulaires définis dans un fichier HTML
  - Syntaxe :
    - document.forms["nom_formulaire"].propriété
    - document.forms["nom_formulaire"].méthode
  - Plusieurs exemples sous :
    - http://fr.selfhtml.org/javascript/objets/forms.htm

Le formulaire

- Exemple : forms1.html
  - Il s'agit d'accéder à la case à cocher pour modifier le contenu de la zone du texte en inscrivant : case cochée ou case non cochée
  - La modification se fera par la fonction ModifChamp();
  - On déclare la case à cocher et la zone texte comme suit :

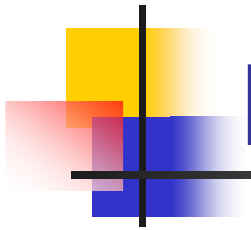
```
<form name="form1"> <br>
<input type="checkbox" name="checkbox"
   onClick="ModifChamp();"> <br>
<input type='TEXT' name='champ_text' value='Essai du
   javascript' size='24'>
</form>
```



Le formulaire

- Ensuite, on se réfère à la case à cocher et à la zone de texte à travers forms :

```
<script language="Javascript">
  function ModifChamp() {
  if (document.forms["form1"].checkbox.checked) {
    document.forms["form1"].champ_text.value='Bouton
  coché' }
  else {
  document.forms["form1"].champ_text.value='bouton
  non coché' } }
</script>
```



Le formulaire

- Le champ form a plusieurs propriétés :
  - **Action ()**
    - Définit l'URL où le formulaire sera envoyé
  - **Elements**
    - Tableau représentant les éléments du formulaire
  - **Length**
    - Nombre d'éléments à l'intérieur du formulaire
  - **Method**
    - Définit le type d'envoi du formulaire (get ou post)
  - **Name**
    - Définit le nom du formulaire
  - **Target**
    - Définit la page (fenêtre ou frame) de réponse
  - **Parent**
    - Indique une fenêtre d'un cadre (frame)

Le formulaire

- La propriété **action ()** : forms2.html
  - Pour réaliser l'action qui accompagne le formulaire

```
<html>
  <head><title>Test</title>
        <script type="text/javascript">
              function confirmation() {
              window.confirm("Ce formulaire est envoyé à
              " + document.formulaire_test.action); }
        </script>
  </head>
  <body>
        <form name="formulaire_test" action="mailto:toi-
        meme@cheztoi.com"  onSubmit="confirmation()">
        <input type="text" size="40" name="saisie">
        <input type="submit" value="Envoyer">
        </form>
  </body>
</html>
```

Le formulaire

- La proriété **length :** form3.html
  - donne le nombre de formulaires définis

```
<body>
<form name="formulaire_test" action="mailto:toi-meme@cheztoi.com">
    <input type="hidden" value="Daniel">
    <input type="submit" value="Daniel">
</form>
<form name="formulaire_test" action="mailto:toi-meme@cheztoi.com">
    <input type="hidden" value="Antoine">
    <input type="submit" value="Antoine">
</form>
<script type="text/javascript">
    document.write(document.forms.length + " formulaires");
</script>
</body>
```

Le formulaire

- La propriété **method :**
  - Sauvegarde la valeur qui figure lors de la définition du formulaire dans l'attribut method, en principe égale à "get" ou "post"
  - Si l'utilisateur envoie le formulaire en cliquant sur le bouton d'envoi la fonction Methode() est appelée

```
<body>
  function Methode() {
        if(document.formulaire_test.action.indexOf("@") > 0)
          document.formulaire_test.method = "post";
        else
          document.formulaire_test.method = "get";
          return true;
  }
  …
  <form name="formulaire_test" action="mailto:toi-
        meme@cheztoi.com" onSubmit="return Methode()">
        <input type="text" size="40" name="saisie">
        <input type="submit" value="Envoyer">
  </form>
```

Le formulaire

- La propriété **name :**
  - Sauvegarde le nom d'un formulaire

```
<html>
   <head><title>Test</title>
   </head>
<body>
   <form name="formulaire_test" action="mailto:toi-
   meme@cheztoi.com">
            <input type="text" size="40" name="saisie">
            <input type="submit" value="Envoyer">
   </form>
   <script type="text/javascript">
            <!--
                     document.formulaire_test.saisie.value =
                              document.formulaire_test.name;
            //-->
   </script>
</body>
</html>
```

Le formulaire

- L’envoi de mail :
  - On ne peut pas envoyer un formulaire tel qu’il est par mail (il faut utiliser php)
  - Cependant, on peut utiliser la formule suivante pour composer totalement un mail :
    - window.open("MAILTO:" + sDestinataire + " ?subject= " + sObjet + " &body=" + document.forms[0].elements["ta_commentaires"]) ;
      - mailto : pour l’adresse
      - ?subject : pour le sujet
      - &body : pour le texte du mail
      - ta_commentaires : est une chaîne de caractères qui rassemble l’information à mettre dans le corps du mèl
    - Exemple : forms5.html

Le formulaire

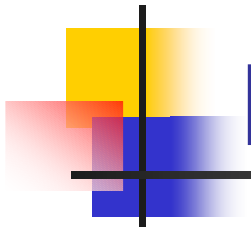
- La propriété **target :**
  - Précise la cible (cadre) dans laquelle l'affichage sera fait :

```
<html>
<head><title>Test</title>
        <script type="text/javascript">
            function cible() {
                    document.formulaire_test.target = "bas";
                    return true; }
        </script>
</head>
<body>
        <form name="formulaire_test" action="fichier.htm"
         onSubmit="return cible()">
        <input type="text" size="40" name="saisie">
        <input type="submit" value="Envoyer">
        </form>
</body>
</html>
```

Le formulaire

- L'action **submit() :**
  - Permet l'envoi du formulaire : JavaScript lance un compte à rebours avec la méthode **setTimeout()**. Après 60000 millièmes de secondes, donc après une minute, la fonction on_y_va() est appelée. Celle-ci envoie le formulaire avec submit()

```
<html>
<head><title>Test</title></head>
<body>
    <form name="formulaire_test" action="/cgi-bin/estime.pl"
        method="get">
            <input type="text" size="40" name="champ1"><br>
            <input type="text" size="40" name="champ2"><br>
        </form>
        <script type="text/javascript">
            function on_y_va() {
              document.formulaire_test.submit(); }
              window.setTimeout("on_y_va()",60000);
    </script>
</body></html>
```



Les elements

- Le champ **elements : sous-objet de forms**
  - Propriétés :
    - checked (coché)
    - defaultChecked (coché par défaut)
    - defaultValue (valeur entrée par défaut)
    - form (nom du formulaire)
    - name (nom de l'élément)
    - type (type de l'élément)
    - value (valeur/contenu de l'élément)
  - Méthodes :
    - blur() (quitter l'élément)
    - click() (cliquer sur l'élément)
    - focus() (positionner sur l'élément)
    - handleEvent() ((traiter l'événement)
    - select() (sélectionner du texte)

Le formulaire

- Checked : exemple: Sauvegarde si oui ou non une case à cocher ou une case d'option est activée. Les valeurs possibles sont true ou 1 ou false ou 0.

```
<script type="text/javascript">
    <!-- function Ensuite() {
       if(document.formulaire_test.mode[0].checked == true)
       window.location.href="fichierfrm.htm"; else
       if(document.formulaire_test.mode[1].checked == true)
       window.location.href="fichier.htm";
       else alert("Veuillez faire un choix"); }
    //-->
</script>
</head><body>
<form name="formulaire_test" action="">
    <input type="radio" name="mode" value="avec"> avec cadres
    <input type="radio" name="mode" value="sans"> sans cadres <br>
    <input type="button" value="Lancer" onClick="Ensuite()">
</form>
</body>
</html>
```

Le formulaire

- **defaultValue :** Sauvegarde le texte par défaut d'un champ de saisie

```
<html><head><title>Test</title> </head>
<body>
    <form name="formulaire_test" action=""> uri: <input size="40"  name="uri"
    value="http://www.xy.fr/">
      <input type="button" value="Vas-y"
      onClick="window.location.href=document.formulaire_test.uri.value">
    </form>
    <script type="text/javascript">
    <!--
      if(navigator.userAgent.indexOf("en") > 0) {
      document.formulaire_test.url.defaultValue = "http://www.xy.com/";
      document.formulaire_test.url.value =
      document.formulaire_test.url.defaultValue; }
      //-->
    </script>
    </body>
</html>
```