DÉVELOPPEMENT INFORMATIQUE

LOGIQUE & PROGRAMMATION

LANGAGE JAVASCRIPT

Y. DELVIGNE CH. LAMBEAU

MANIPULER LE DOM (DOCUMENT OBJECT MODEL) AVEC JAVASCRIPT

PROGRAMMATION ÉVÉNEMENTIELLE : "IN-LINE MODEL"

CONTEXTE (RAPPEL)

➢ LES DEUX PHASES DE LA 'VIE' D'UNE PAGE :

- **PHASE PRÉLIMINAIRE DE RENDERING**

  construction et mise en forme par le navigateur (html, css + script éventuel)

- **PUIS, PHASE ÉVÉNEMENTIELLE**

  réactivité suite aux événements déclenchés

  - **par le navigateur, p.ex.**
    - phase de rendering terminée
    - fermeture de la page, ...
  - **par l'utilisateur, p.ex.**
    - mouvements ou clic de la souris sur les éléments
    - champs de formulaire, widgets, liens ...

DOM : 'ÉVÉNEMENTIEL'

QUELQUES ☺ ÉVÉNEMENTS ...

LA PAGE ELLE-MÊME

Handler	déclenché
onBlur	quand la page perd le focus (p.ex. si le curseur de souris la quitte)
onFocus	quand la page reçoit le focus (p.ex. par clic de souris)
onLoad	quand la page est chargée (fin du rendu)
onMove	quand la page est déplacée
onUnLoad	quand la page est fermée ou réinitialisée

LES 'ACTIONS' DE LA SOURIS

Handler	déclenché
onClick	quand on clique un élément
onDblClick	quand on double-clique un élément
onMouseDown	quand un bouton de souris est appuyé
onMouseEnter	quand la souris 'entre' sur un élément
onMouseMove	quand la souris se déplace à l'intérieur d'un élément
onMouseOut	quand la souris 'sort' d'un élément
onMouseOver	quand la souris 'entre' sur un élément ou un de ses fils
onMouseUp	quand un bouton de souris est relâché

DOM : 'ÉVÉNEMENTIEL'

QUELQUES ☺ ÉVÉNEMENTS ...

LES FORMULAIRES ET CHAMPS

Handler	déclenché
onBlur	quand un champ de formulaire perd le focus (sortie du curseur)
onChange	après modification d'un champ de formulaire (sortie du curseur)
onClick	quand un objet de formulaire est cliqué
onFocus	quand un champ de formulaire reçoit le focus (clic souris ou tabulation clavier)
onReset	quand on réinitialise un formulaire (p.ex. bouton reset)
onSelect	lors de la sélection de texte dans un champ texte (input, texarea)
onSubmit	quand on soumet le formulaire (p.ex. bouton submit)

LES LIENS (ANCRES)

Handler	déclenché
onClick	quand le lien est cliqué
onMouseOut	quand la souris quitte le lien
onMouseUp	quand le lien est décliqué
onMouseOver	quand la souris passe sur le lien

PROGRAMMATION ÉVÉNEMENTIELLE : PRINCIPE ("IN-LINE" MODEL)

on sait associer toutes sortes d'événements aux différents (à n'importe quel) éléments HTML (`body, form, p, a, img` ...) et y associer des scripts JavaScript (= exécuter du code lorsque les événements surviennent)

un événement a pour nom générique **on*Event***
p.ex. onLoad, onLink, onClick, onSubmit ...

l'association entre événement et code JS à exécuter se fait sous forme d'attribut de l'élément HTML : **«on*Event*» = '«codeJs»'**

il est de bonne pratique que le code JS à exécuter soit contenu dans une **fonction** et que l'ensemble des fonctions constitue le script placé dans le **head** (ou dans fichier externe)

ce script devient ainsi une librairie de fonctions

et de la sorte l'association devient **«on*Event*» = '«fonctionJs» '**

PROGRAMMATION ÉVÉNEMENTIELLE : PRINCIPE ("IN-LINE" MODEL)

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function dClick() {
      alert('div cliqué !');
    }
  </script>
</head>

<body>
  <div onClick = 'dClick();'>
     contenu du div
  </div>
  <p>
    contenu du paragraphe
  </p>
</body>
</html>
```

ON ASSOCIE À L'ÉLÉMENT
UN "EVENT HANDLER" OU "LISTENER"

sur cet élément …

… quand cet événement survient …

… j'invoque cette fonction …

… définie là

events01.html

PROGRAMMATION ÉVÉNEMENTIELLE : EXEMPLE

```
<head>
  <script>
    function pLoaded() { alert('page chargée !'); }
    function dClick() { alert('div cliqué !'); }
    function pEnter() { alert('para in !'); }
    function pOut()   { alert('para out !'); }
  </script>
</head>

<body onLoad = 'pLoaded();'>

  <div onClick = 'dClick();'>
     contenu du div
  </div>

  <p onMouseEnter = 'pEnter();' onMouseOut = 'pOut();'>
    contenu du paragraphe
  </p>

</body>
```

events02.html

PROGRAMMATION ÉVÉNEMENTIELLE : PARAMÈTRES

events03.html

```
<html>
<head>
  <script>
    function colorize(d, c) {
      document.getElementById(d).style.background = c;
    }
  </script>
</head>
<body>
  <div id=d1>(div id=d1)
    <p>para 1</p>
    <p>para 2</p>
    <div id=d2>(div id=d2)
      <p>para 3</p>
      <p>para 4</p>
    </div>
  </div>
  couleur de fond du div id=d2 :
  <input type=button value=vert  onClick = "colorize('d2', 'green');">
  <input type=button value=rouge onClick = "colorize('d2', 'red');">
</body>
</html>
```

```
<style>
  #d1, #d2 { margin: 10px;
             padding: 10px;
             border: 1px solid black;
             background: white; }
</style>
```

PROGRAMMATION ÉVÉNEMENTIELLE : THIS

events04.html

```
<head>
  <style> div, p { background : green; } </style>
  <script>
    function oClick(o) { alert(o.innerHTML); }
    function oEnter(o) { o.style.background = 'pink'; }
    function oOut(o)   { o.style.background = 'green'; }
  </script>
</head>


<body>
  <div onClick = 'oClick(this);'
       onMouseEnter = 'oEnter(this);'
       onMouseOut = 'oOut(this);'>
     contenu du div
  </div>
  <p onClick = 'oClick(this);'
     onMouseEnter = 'oEnter(this);'
     onMouseOut = 'oOut(this);'>
    contenu du paragraphe
  </p>
</body>
```

… symbolise l'élément en tant qu'objet … … passé comme paramètre à la fonction…

… qui peut ainsi être réutilisée …

PROGRAMMATION ÉVÉNEMENTIELLE : PROPAGATION & BUBBLING

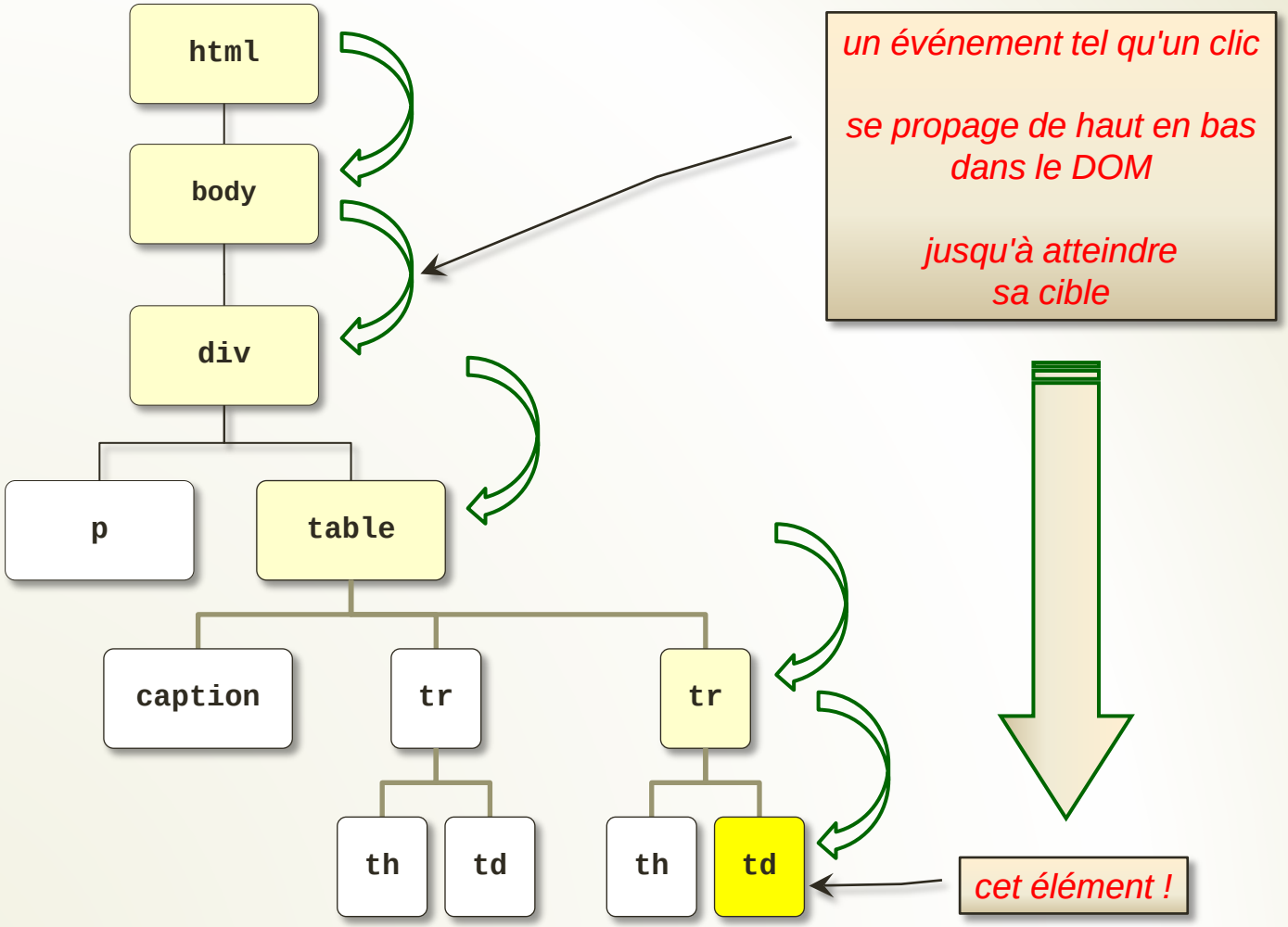
```
<head>
  <script>
      function oClick(o) {
        o.style.background = 'yellow';
        alert(o.nodeName);
      }
  </script>
</head>

<body onClick = 'oClick(this);'>
  <div onClick = 'oClick(this);'>
     <p>contenu du div</p>
     <table onClick = 'oClick(this);'>
       <caption onClick = 'oClick(this);'> titre table </caption>
       <tr onClick = 'oClick(this);'>
         <th onClick = 'oClick(this);'> titre 1 </th>
         <td onClick = 'oClick(this);'> cellule 1 </td>
       </tr>
       <tr onClick = 'oClick(this);'>
         <th onClick = 'oClick(this);'> titre 2 </th>
         <td onClick = 'oClick(this);'> cellule 2 </td>
       </tr>
     </table>
  </div>
</body>
```

events05.html

DOM : 'ÉVÉNEMENTIEL'

'EVENT MODEL' : PROPAGATION (IN-LINE EVENT)



DOM : 'ÉVÉNEMENTIEL'

'EVENT MODEL' : BUBBLING (IN-LINE EVENT)