



UNIVERSITÉ
DE REIMS
CHAMPAGNE-ARDENNE

JavaScript

Nils Schaefer

[nils.schaefer@snicw.fr](mailto:nil.schaefer@snicw.fr)

Langage Javascript

- Vrai langage de programmation
 - Exécution côté client
- Apporter du dynamisme aux pages HTML
 - Dynamisme d'interface
- Syntaxe générale similaire au C/C++, Java
- Dans le fond, rien à voir avec Java
- Prise en compte de la casse
- Utilisé sur la grande majorité des sites Web
- Très bonne compatibilité entre les navigateurs
 - Quelques petites adaptations restent parfois nécessaires

Variables

- Typage dynamique des variables
 - Une variable est du type de la donnée qu'elle contient !
- Déclaration (généralement facultative) : `var i`
- Types de base
 - Nombres entiers
 - Nombres à virgule
 - Booléens (*true / false*)
 - Chaînes de caractères ("`...`" ou '`...`')
 - Tableaux
 - Exemple : `var tab=new Array()`
 - Case 2 du tableau : `tab[1]`
 - Longueur d'un tableau : `tab.length`

Expressions

- « Calcul » qui possède une valeur
 - Utilisation possible des parenthèses ()
- Opérateurs numériques : + - * / %
 - Exemple : $(5 + 8) * 9 - i$
- Opérateurs booléens : == != && || < <= >= >
 - Exemple : $5 * i > 10 \ \&\& \ 10 - i == 2$
- Opérateur de concaténation : +
 - Exemple : "voici" + nb + "voitures"
 - Attention : risque de confusion entre l'addition et la concaténation
 - "1" + "1" → "11"
 - $1 + 1 \rightarrow 2$
 - `Number("1")` → 1
 - Dans « a + b », le + correspond à l'opérateur d'addition ou de concaténation ?

Instructions

- Réalise une action
 - Convention de placer un ; en fin d'instruction
- Affectation : attribuer une valeur à une variable
 - Affectation standard : =
 - Exemple : $nb = 5 * i$
- Affectations raccourcies : += -= *= /=
 - Exemple : $q += 5$ (équivalent à $q = q + 5$)
- Incrémentation/décrémentation : ++ --
 - Exemple : $i++$ (équivalent à $i = i + 1$)
- Bloc d'instructions : { ... }
 - Regroupe plusieurs instructions

Structure conditionnelle

- Structure conditionnelle

- `if(test) { ... }`
- `if(test) { ... } else { ... }`
- `if(test1) { ... } else if(test2) { ... } else { ... }`
- ...

- Exemple

```
if(nb>5 && prix==10)
{
    ...
}
else
{
    ...
}
```

Instructions de boucle

- Boucle « Pour »
 - Nombre d'itérations connu
 - Exemple : `for (i=0; i<10; i++) { ... }`
- Boucle « Tant que »
 - Nombre d'itérations inconnu avec de 0 à n itérations
 - Exemple : `while (i>10) { ... }`
- Boucle « Faire tant que »
 - Nombre d'itérations inconnu avec de 1 à n itérations
 - Exemple : `do { ... } while (i>10);`
- `continue` : passe à l'itération suivante
- `break` : sort de la boucle

Fonctions

- Sous programme avec le mot-clé fonction
 - Liste de paramètres entre parenthèses (séparés par des virgules)
 - Une fonction retourne une valeur

- Exemple

```
function somme(n)
{
    s=0;
    for(i=0;i<n;i++)
    {
        s=s+i;
    }
    return s;
}
...
x=somme(15);
```


- Sous programme avec le mot-clé fonction
 - Liste de paramètres entre parenthèses (séparés par des virgules)
 - Une procédure ne retourne pas de valeur
 - Pas de return contrairement à une fonction
- Exemple

```
function DitBonjour(nom)
{
    alert("Bonjour "+nom);
}
...
DitBonjour("Pierre");
```

Variables globales/locales

- Variables dont la durée de vie est limitée à une fonction/procédure
 - Un paramètre est toujours une variable locale
 - Déclaration d'une variable locale : var i
- Une variable locale peut avoir le même nom qu'une variable globale
 - Variable globale masquée

- Exemple

```
function incrementer(i)
{
    return i+1;
}

...
i=3;
i=incrementer(i);
```

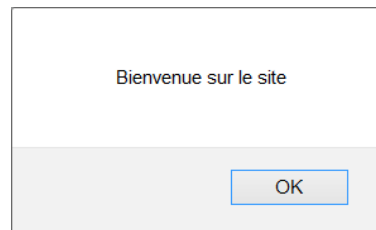
Interactions entre HTML et Javascript

- Balise `<SCRIPT> ... </SCRIPT>`
 - `type` : identification du langage de script utilisé (plus utile)
 - `<script type="text/javascript"> ... </script>`
 - `language` : similaire à `type` (déprécié)
 - `<script language="Javascript"> ... </script>`
 - `src` : désignation d'un fichier de code externe (extension .js)
 - `<script type="text/javascript" src="script.js"></script>`
- Le code Javascript est placé à l'intérieur des balises et s'exécute lors de la lecture de la page
- Localisation
 - En-tête : `<HEAD> ... </HEAD>`
 - Corps : `<BODY> ... </BODY>`

Interactions avec l'utilisateur (1)

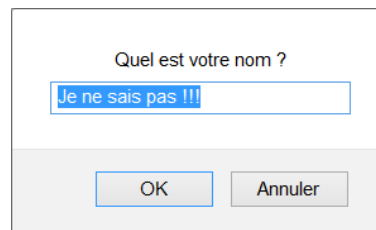
- Fonction alert : message dans une fenêtre

- `alert('Bienvenue sur le site');`



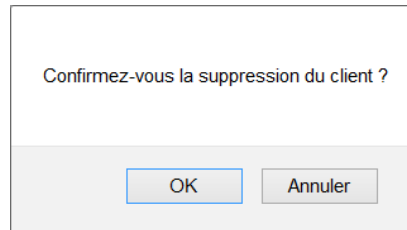
- Fonction prompt : récupérer une donnée en provenance de l'utilisateur

- `nom=prompt('Quel est votre nom ?','Je ne sais pas !!!');`
- Valeur saisie stockée dans la variable nom



Interactions avec l'utilisateur (2)

- **Fonction confirm : demander à l'utilisateur une confirmation**
 - `res=confirm('Confirmez-vous la suppression du client ?');`
 - `res` contient `true` ou `false`



- **Interactions bloquantes**
 - L'utilisateur ne peut plus rien faire d'autre sur la page
 - A utiliser de manière très limitée
- **Protection contre les affichages répétés**
 - Case à cocher pour brider les affichages

Code HTML dynamique

- Javascript produit « à notre place » du code HTML
- Lors du chargement de la page uniquement
- Fonction document.write
- Exemple

```
<script>
for (i=0;i<10;i++)
{
    document.write ('<h1>Titre ' + (i+1) + '</h1>');
}
</script>
```

Programmation événementielle

- Réaction de la page HTML à des événements de l'utilisateur ou du système dans son ensemble
 - Survol avec la souris
 - Clic à un endroit
 - Frappe d'une touche
 - Redimensionnement de la fenêtre
 - ...
- Code Javascript situé directement dans du HTML
- Exécution de code Javascript
 - Généralement une fonction ou une procédure

- Certains événements sont spécifiques à une balise, d'autres sont applicables à la majorité des balises.
- Nombreux événements
 - onclick : clic
 - ondblclick : double-clic
 - onmouseover : début du survol d'un élément
 - onmouseout : fin du survol d'un élément
 - onload : fin du chargement de la page Web
 - onchange : modification d'un champ de formulaire
 - onkeyup : frappe d'une touche
 - ...

Réaction à un événement

■ Exemples

- Exécuter du code suite à un clic sur un div

- `<div onclick="...">Test</div>`

- Exécuter du code lors du chargement de la page

- `<body onload="...">Test</body>`

- Exécuter du code lors du survol d'un lien

- `Google`

- Exécuter du code lors d'un changement de contenu

- `<input type=text onchange="...">`

Réaction à un événement

- Entre les " " doit être placé le code Javascript
 - Même s'il est possible de placer tout le code Javascript souhaité pour réaliser l'action on passe plutôt par l'utilisation d'une procédure.
- Exemple

```
<script>
function ActionA()
{
    // Code à exécuter lors du clic
}
</script>
<div onclick="ActionA();">Mon beau div</div>
```

Manipulation d'un élément quelconque

- Chaque élément de la page HTML est accessible
 - Plusieurs méthodes
 - En fonction de l'objectif
- Utilisation de la structure du document
 - Parcours du document comme un arbre
- Utilisation de l'identifiant d'une balise
 - Plus direct et plus simple
- Fonction `getElementById` de l'objet document
 - Ex (HTML) : `<p id="MonP"> ... </p>`
 - Ex (JS) : `document.getElementById('MonP')`

Manipulation d'un champ de formulaire

- Propriété **value** pour accéder au contenu du champ
- Exemple (Partie HTML)

```
<FORM action=script.php>
```

```
Nom : <INPUT Type=text Name=nom id=c1><BR>
```

```
...
```

```
</FORM>
```

- Exemple (Partie Javascript)

```
// Récupère le contenu du champ dans mavar
```

```
mavar=document.getElementById('c1').value;
```

```
// Affiche dans le champ la valeur 10
```

```
document.getElementById('c1').value=10;
```

Contrôle sur les formulaires (1)

- Propriétés HTML 5 de la balise `<input>`
 - Placeholder : texte indicatif sur la donnée à saisir
 - `<input type=text placeholder="Votre login">`
 - Required : indique un champ obligatoire
 - `<input type=email required>`
 - Pattern : expression rationnelle
 - `<input type=codepostal pattern="[0-9]{5}">`
- Très appréciable mais limité
 - Pas possible de créer des règles spécifiques
- Validation à réaliser en Javascript

Contrôle sur les formulaires (2)

- Événement onsubmit sur la balise <form>
 - Événement déclenché lors du clic sur le bouton de validation
 - Possibilité de retourner
- True : les données sont envoyées au programme
- False : l'envoi des données est bloqué
- Exemple

```
<form action=script.php onsubmit="return test();" >
    <input type=text>
    ...
</form>
<script>
function test()
{
    // Faire un return false ou true en fonction des tests
}
</script>
```

Contrôle du temps

- Fonction `setTimeout`
 - Prévoir d'exécuter une fonction plus tard
 - Exemple : `a1=setTimeout("test()", 5000);`
- Fonction `clearTimeout`
 - Annuler une exécution future
 - Exemple : `clearTimeout(a1);`
- Fonctions similaires avec répétition
 - `setInterval`
 - `clearInterval`

Manipulation des données d'une balise

- Les données (tout le contenu) d'une balise sont accessibles en Javascript

- Propriété **innerHTML**

- Exemple (Partie HTML)

```
<h1 id=t1>Mon titre de <u>niveau 1</u></h1>
```

- Exemple (Partie Javascript)

```
// Récupérer les données de la balise dans a  
a=document.getElementById('t1').innerHTML;  
// Remplacer les données de la balise par a  
document.getElementById('t1').innerHTML=a;
```


Manipulation des styles d'une balise

- Les styles peuvent être manipulés en Javascript
 - Propriété **style**
- Attention : le - n'est pas utilisable car il représente la soustraction donc on le retire et on met en majuscule le caractère suivant
 - Ex : background-color devient backgroundColor

■ Exemple (Partie HTML)

```
<h1 id=t1>Mon titre</h1>
```

■ Exemple (Partie Javascript)

```
document.getElementById('t1').style.color='red';
```

Objets du DOM

- *DOM = Document Model Object*
- Plusieurs objets accessibles
 - navigator : le navigateur
 - navigator.appName, navigator.language ...
 - window : la fenêtre qui contient la page HTML
 - window.innerWidth, window.location, window.history ...
 - document : la page HTML
 - document.cookie, document.forms, document.images, document.links ...
 - ...