

Cours

Programmation avancée pour le web

Credit : 4

Coefficient : 2

Mode d'évaluation :

- Examen (60%) - contrôle continu (40%)

Author:

Aimad Eddine Debbi

Chap V JavaScript

JavaScript :

Langage de script léger, orienté objet, utilisé communément pour le développement web mais utilisé aussi en dehors des browsers. Le code JavaScript est interprété ou compilé à la volée (JIT) Just in Time.

Si l'HTML définit la structure et le contenu des pages web, et les CSS ajoutent des caractérisations et des allures, le javascript rend dynamique les pages et définit du comportement.

on se permet :

- la validation des formulaires, les calculs, les messages,...
- la modification de la page et style,...
- la communication avec le serveur (Ajax).

Chap V JavaScript

JavaScript :

On se permet de façon dynamique de

- Changer le contenu HTML.

```
document.getElementById('h_id').innerHTML = 'other title';
```

- Changer les valeurs des attributs.

```
document.getElementById('img_id').src = 'altrnt.gif';
```

Chap V JavaScript

JavaScript :

On se permet de façon dynamique

- La mise à jour du style.

```
document.getElementById('div_id').style.fontSize = '18px';
```

- Masquer / faire apparaitre un élément.

```
document.getElementById('p_id').style.display = 'none';  
document.getElementById('p_id').style.display = 'block';
```

Chap V JavaScript

JavaScript :

Emplacement:

- Au niveau du <head>.

```
<head>
<script>
function function__xmp1e() {
    document.getElementById("para__id").innerHTML = "Paragra
ph changed.";
}
</script>
</head>
```

Chap V JavaScript

JavaScript :

Emplacement:

- Au niveau du <head>. (suite)

```
<!-- suite -->
<body>
<h2>Demo JavaScript in Head</h2>
<p id=" para__id ">A Paragraph</p>
<button type="button" onclick="function__xemple()">Change
it</button>
</body>
```

Chap V JavaScript

JavaScript :

Emplacement:

- Au niveau du <body>.

```
<body>
<h2>Demo JavaScript in Head</h2>
<p id=" para__id ">A Paragraph</p>
<button type="button" onclick=" myFunction() ">Change
it</button>
<script>
function myFunction() {
    document.getElementById("para__id").innerHTML = "Paragrap
h changed.";
}
</script>
</body>
```

Chap V JavaScript

JavaScript :

Emplacement:

- Sur un fichier externe.

```
<body>
<h2>Demo JavaScript in Head</h2>
<p id=" para__id ">A Paragraph</p>
<button type="button" onclick=" myFunction() ">Change
it</button>

  <script src="myScript.js"></script>

</body>
```


Chap V JavaScript

JavaScript :

Emplacement:

- Sur un fichier externe.

myScript.js

```
function myFunction() {  
    document.getElementById("demo").innerHTML = "Paragraph  
changed."  
}
```

Chap V JavaScript

JavaScript :

Les sorties:

Différentes façon de faire des écritures:

- Au niveau d'un élément avec `.innerHTML`
- Sur la sortie avec l'appel `document.write()`
- Sur boîte "alerte" par `window.alert()`
- Sur la console par `console.log()`
- Boîte saisie de texte par `prompt()`
- Boîte confirmation par `confirm()`

Chap V JavaScript

JavaScript / Les sorties

Différentes façon de faire des écritures:

(innerHTML, document.write(), innerHTML,
document.write())

Exemple

```
<!DOCTYPE html>
<html>
<body>
<h1 id="h1__id"> page title</h1>
<p> paragraph text in page</p>
<input type="button" onclick='actn()' value='click it'>
<br><br>
```

Chap V JavaScript

JavaScript / Les sorties

Différentes façon de faire des écritures:

(innerHTML,document.write(),innerHTML, ocument.write())

Exemple

Rendu

```
<script>
s=prompt('enter string');
agree = confirm("do you agree");
document.write(" befor end-load ");
function actn(){ document.write(" after end-load by
function ");
document.getElementById("h1__id").innerHTML = " updated
title";
window.alert(' alerte text .....');
}
</script>
</body>
</html>
```

Chap V JavaScript

JavaScript / Les variables

Façons de déclaration :

(var, let et const)

Exemple

```
var person = "John Doe ",  
    carName = " BMW",  
    price = 50;  
  
let author = " John Doe ";  
let score = 10;
```

Chap V JavaScript

JavaScript / Les variables

La portée (scope) (**var**, **let**)

var a une portée global .

let pour portée locale.

Exemple

```
var x = 2;    // Allowed
let x = 3;    // Not allowed
{
let x = 2;    // Allowed
let x = 3     // Not allowed
}
{
let x = 2;    // Allowed
var x = 3     // Not allowed
}
```

Chap V JavaScript

JavaScript / Les objets

En javascript tout est objet presque.

Exemple (Définitions des propriétés et des méthodes
Pour l'objet person)

```
const person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
```

Chap V JavaScript

JavaScript / Les objets

La création faite par :

- Définition latérale de l'objet.
- Utilisant le mot réservé **new**.
- Utilisant le constructeur.
- Utilisant le mot réservé **Object.create()**

Chap V JavaScript

JavaScript / Les objets

La création faite par :

- Définition latérale de l'objet.

```
const person = {  
  firstName:"John",  
  lastName:"Doe",  
  age:50,  
  eyeColor:"blue"  
};
```

Chap V JavaScript

JavaScript / Les objets

La création faite par :

- Définition latérale de l'objet.

Et aussi

```
const person = {};  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

Chap V JavaScript

JavaScript / Les objets

La création faite par :

- Utilisant le mot réservé **new**.

```
const person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

Chap V JavaScript

JavaScript / Les objets

La création faite par :

- Utilisant le constructeur.

```
function Person(first, last, age, eye) {  
  this.firstName = first;  
  this.lastName = last;  
  this.age = age;  
  this.eyeColor = eye;  
}  
  
const Worker = new Person("John", "Doe", 50, "blue");
```

Chap V JavaScript

JavaScript / Les objets

La création faite par :

- Utilisant le mot réservé `Object.create()`

```
function Person(first, last) {
  this.firstName = first;
  this.lastName = last;
}
const Worker = new Person("John", "Doe ");
const farmer = Object.create(Person);
farmer.age = 40;
farmer.eye = "brown";
```

Chap V JavaScript (Rappel)

JavaScript / Events

- Événements causé par l'utilisateur: click, doubleclick, modification d'un champ de saisie,.....
- Événements dus au browser: fin de chargement de la page.

Evenements couramment utilisés:

Evenement	Description
onchange	Elément html a changé
onclick	L'utilisateur click sur élément
onmouseover	Sourie au dessus de l'élément
onmouseout	Sourie éloignée de l'élément
onkeydown	Touche clavier pressée
onload	Browser finit le chargement de la page

Chap V JavaScript (Rappel)

JavaScript / Events

Une sélection d'événements liés à l'objet Window:

Evenement	Description
onload	Browser finit le chargement de la page
ononline	Browser s'établie online
onoffline	Browser est hors ligne
onresize	La taille de la fenêtre winfow a été modifiée
onpageshow	L'utilisateur accède à la page (historique)
onpagehide	L'utilisateur quitte la page (historique)

Chap V JavaScript (Rappel)

JavaScript / Events

Une sélection d'événements liés au formulaire:

Evenement	Description
onchange	Valeur d'un élément html a changé
onfocus	L'élément obtient le focus
oninput	Une entrée utilisateur sur un champ de saisie
oninvalid	Élément non valide
onsearch	Saisie sur un champ input = 'search'
onsubmit	Lorsque le fomulaire est envoyé.
onreset	Le bouton reset du formulaire est cliqué
onselect	Un texte a été sélectionné

Chap V JavaScript (Rappel)

JavaScript / Events

Une sélection d'événements liés à la souris:

Evenement	Description
onclick	Click sur un élément
ondblclick	Double Click sur un élément
onmousedown	Boutton souris maintenu pressé sur un élément
onmousemove	Souris en mouvement au dessus d'un élément
onmouseout	Souris en mouvement quittant l'élément
onmouseover	Souris au dessus d'un élément
onmouseup	Souris relâchée au dessus d'un élément

Chap V JavaScript (Rappel)

JavaScript / Events

Une sélection d'événements liés au drag/drop:

[Voir chapitre précédent](#)

Chap V JavaScript (Rappel)

JavaScript / Events

Une sélection d'événements liés au clavier:

Evenement	Description
onkeypress	Touche clavier enfoncée
onkeyup	Touche clavier relachée

Chap V JavaScript (Rappel)

JavaScript / Actions asynchrones

Les actions/taches asynchrones sont implémentés avec les primitives javascript `setTimeout()` et `setInterval()`. On se permet de lancer des actions à des moments variables dans le temps. L'instant d'exécution de l'action est un délai en millisecondes défini par l'utilisateur commençant au moment de l'appel de `setTimeout()`. Ce délai est spécifié comme deuxième argument dans l'appel `setTimeout()`.

Fonction	Description
<code>setTimeout(action, delay)</code>	Lancer action après une durée égale delay millisecondes.
<code>setInterval(action, delay)</code>	Lancer action périodiquement après une durée égale delay millisecondes.

Chap V JavaScript (Rappel)

JavaScript / Actions asynchrones

`setTimeout()` et `setInterval()` renvoient un id qui pourra être utilisé pour stopper l'action via la primitive `clearInterval()`.

Fonction	Description
<code>Let id__task = setTimeout(action, delay)</code>	Lancer action après une durée égale delay millisecondes.
<code>clearInterval(id__task);</code>	Faire stopper l'action .

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- String Comme objet :

```
Let s = new String("John doe");
```

Method (extraction text)	Description
slice(start,end)	Extrait une partie et la renvoie comme string. start et end peuvent être négatifs.
substring(start,end)	Similaire à slice avec non négatifs indexes
substr(start,length)	Similaire à slice à part 2 ^{ème} argm

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- String Comme objet :

Method (remplacement)	Description
replace("txt", "new_txt")	Remplace l'occurrence txt par new_txt.
concat("txt1", "txt2")	Concatination
charAt(position)	Renvoie le cara figurant à position
toUpperCase()	Majiscule
toLowerCase()	Miniscue

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- Number Comme objet :

```
Let y = new Number(456);
```

Method (remplacement)	Description
toString(nbr)	Renvoie le nombre autant que string
valueOf(start,end)	Renvoie le nombre autant que nombre
parseInt("string_nbr")	Convert string to number
parseFloat("string_nbr")	Convert string to number
Number(expression)	Conversion vers nombre

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- Array Création :

```
const city = ["London", "Rome", "Istambul"];
```

Et aussi

```
Let city = [];  
city[0] = "London";  
city[1] = "Rome";  
city[2] = "Istambul";
```

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- Array as object :

```
const city = new Array("London", "Rome", "Istambul");
```

Methode (remplacement)	Description
toString()	Extrait les éléments autant que string
pop()	Extrait et supprime le dernier élément
push(element)	Empile un élément
shift()	Extrait et supprime le premier élément
Concat(an__array)	Concatenation

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- Date as object :

```
const d = new Date();
```

```
const d = new Date(year, month, day, hours, minutes,  
seconds, milliseconds);  
const today = new Date(2020, 5, 22, 10, 10, 30, 0);
```

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- Date as object :

```
const d = new Date();
```

Method (remplacement)	Description
getTime()	
getFullYear()	
getMonth()	
getHours()	
getDay()	

Chap V JavaScript (Rappel)

JavaScript / Les objets String, Number, Array, Date.

- Date as object :

```
const d = new Date();
```

Method (remplacement)	Description
getTime()	Full text string format
getFullYear()	
getMonth()	
getHours()	
getDay()	