
Cours de Javascript VII

The letters 'JS' in a bold, black, sans-serif font are centered within a bright yellow square. This square is positioned on the right side of the slide, overlapping a larger, lighter orange square that serves as a background element. The overall design is clean and modern, with a focus on the text and the 'JS' logo.

JS

Partie VII

1. Utilisation des différents sélecteurs d'éléments

- Sélecteur Javascript par valeur d'attribut id
- Sélecteur Javascript de type de balise HTML
- Sélecteur Javascript par valeur d'attribut name
- Sélecteur Javascript par valeur de l'attribut class
- Sélecteur Javascript unique avec sélecteurs CSS
- Sélecteur Javascript multiple avec sélecteurs CSS

1.1 Utilisation des différents sélecteurs d'éléments

- Sélecteur Javascript par valeur d'attribut id

La méthode javascript `getElementById()` de l'objet Document permet de sélectionner une balise ayant un attribut `id` (la valeur de l'id doit être unique dans votre page HTML).

```
document.getElementById('#monId')
```

La méthode javascript `document.getElementById(paramètre)` accepte comme paramètre:

- chaîne de caractère, qui représente la valeur de l'attribut `id`, sensible à la casse.

La méthode javascript `document.getElementById()` retourne :

- `null`, si pas de résultat.
- `Element`, objet de type `Element` qui est l'objet récupéré. Si vous avez plusieurs `id` de même valeur, c'est le dernier trouvé qui est sélectionné, mais cela peut varier selon les navigateurs.

1.1 Utilisation des différents sélecteurs d'éléments

- Sélecteur Javascript de type de balise HTML

La méthode javascript `getElementsByTagName()` de l'objet `Document` ou `Element` permet de sélectionner une balise par son nom (`tagName`).

```
document.getElementsByTagName('SPAN');  
element.getElementsByTagName('LI');
```

Si on utilise la méthode javascript `getElementsByTagName()` de l'objet `Document`, alors la recherche se fera sur l'ensemble du document.

Si on utilise la méthode javascript `getElementsByTagName()` de l'objet `Element`, alors la recherche se fera sur l'ensemble des enfants de cet élément.

La méthode Javascript `*.getElementsByTagName(paramètre)` accepte comme paramètre:*, sélecteur universel, représente tous les noms de balise.chaine de caractère, qui représente le nom de la balise, insensible à la casse.

1.1 Utilisation des différents sélecteurs d'éléments

- *, sélecteur universel, représente tous les noms de balise.
- chaîne de caractère, qui représente le nom de la balise, insensible à la casse.

La méthode Javascript `*.getElementsByTagName()` retourne:

- `NodeList`, liste des objets récupérés de type `Element`. `NodeList.length` renvoie le nombre d'éléments présents.

- Sélecteur Javascript par valeur d'attribut name

La méthode Javascript `getElementByName()` de l'objet `Document` permet de sélectionner une balise par sa valeur d'attribut name.

```
document.getElementById('attribut-name');
```

1.1 Utilisation des différents sélecteurs d'éléments

Si on utilise la méthode javascript `getElementsByName()` de l'objet `Document`, alors la recherche se fera sur l'ensemble du document.

La méthode Javascript `*.getElementsByName(paramètre)` accepte comme paramètre:

- chaîne de caractère, qui représente la valeur de l'attribut name, sensible à la casse.

La méthode Javascript `*.getElementsByName()` retourne:

- `NodeList`, liste des objets récupérés de type `Element`. `NodeList.length` renvoie le nombre d'éléments présents.

- Sélecteur Javascript par valeur de l'attribut class

La méthode Javascript `getElementsByClassName()` de l'objet `Document` ou `Element` permet de sélectionner une balise par un identifiant de `class` CSS contenu dans l'attribut HTML (`class`).

1.1 Utilisation des différents sélecteurs d'éléments

```
document.getElementsByClassName('MaClass');  
element.getElementsByClassName('item');
```

Si on utilise la méthode javascript `getElementsByClassName()` de l'objet `Document`, alors la recherche se fera sur l'ensemble du document.

Si on utilise la méthode javascript `getElementsByClassName()` de l'objet `Element`, alors la recherche se fera sur l'ensemble des enfants de cet élément.

La méthode Javascript `*.getElementsByClassName(paramètre)` accepte comme paramètre:

- **identifiant de class CSS**, qui représente l'identifiant recherché dans l'attribut `class`, sensible à la casse. Vous pouvez mettre plusieurs identifiants de `class` séparés par un espace, dans ce cas l'élément recherché devra avoir tous ces identifiants dans son attribut `class`

1.1 Utilisation des différents sélecteurs d'éléments

La méthode Javascript `*.getElementsByClassName()` retourne:

- `NodeList`, liste des objets récupérés de type `Element`. `NodeList.length` renvoie le nombre d'éléments présents.

- Sélecteur Javascript unique avec sélecteurs CSS

La méthode Javascript `querySelector()` de l'objet `Document` ou `Element` permet de sélectionner une balise grâce à un sélecteur CSS.

```
document.querySelector('BODY > TABLE .MaClass');  
elementTable.querySelector('.MaClass');
```

La méthode javascript `*.querySelector(paramètre)` accepte comme paramètre:

- sélecteur CSS, qui représente le chemin vers élément cible.

1.1 Utilisation des différents sélecteurs d'éléments

La méthode Javascript `*.querySelector()` retourne:

- `null`, si pas de résultat.
- `Element`, qui correspond au premier élément trouvé.

- Sélecteur Javascript multiple avec sélecteurs CSS

La méthode Javascript `querySelectorAll()` de l'objet `Document` ou `Element` permet de sélectionner une balise grâce à un sélecteur CSS.

```
document.querySelectorAll('BODY > TABLE .MaClass');  
elementTable.querySelectorAll('.MaClass');
```

La méthode javascript `*.querySelector(paramètre)` accepte comme paramètre:

- `sélecteur CSS`, qui représente le chemin vers élément cible.

1.1 Utilisation des différents sélecteurs d'éléments

La méthode javascript `*.querySelectorAll()` retourne:

- `HTMLCollection`, liste des objets récupérés de type `Element`
`static.HTMLCollection.length` renvoie le nombre d'éléments présents.



Remarque:

- `document.querySelector("#title");` pour un `id`
- `document.querySelector(".title");` pour une `class`

Liens externes pour aller plus loin

- <https://www.zonecss.fr/cours-css-javascript/les-selecteurs-javascript.html>
- <https://www.alsacreations.com/article/lire/1445-dom-queryselector-queryselectorall-selectors-api.html>