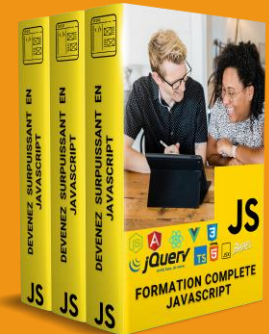


JavaScript : Manipuler le DOM



Formateur : Mr. AKPOLI Espero - Chef de projet digital - +33 7 77 67 41 57 - contact@apprenez-a-coder.fr



Progression

Partie 1

- Les bases

Partie 2

- Agir sur les éléments

Partie 3

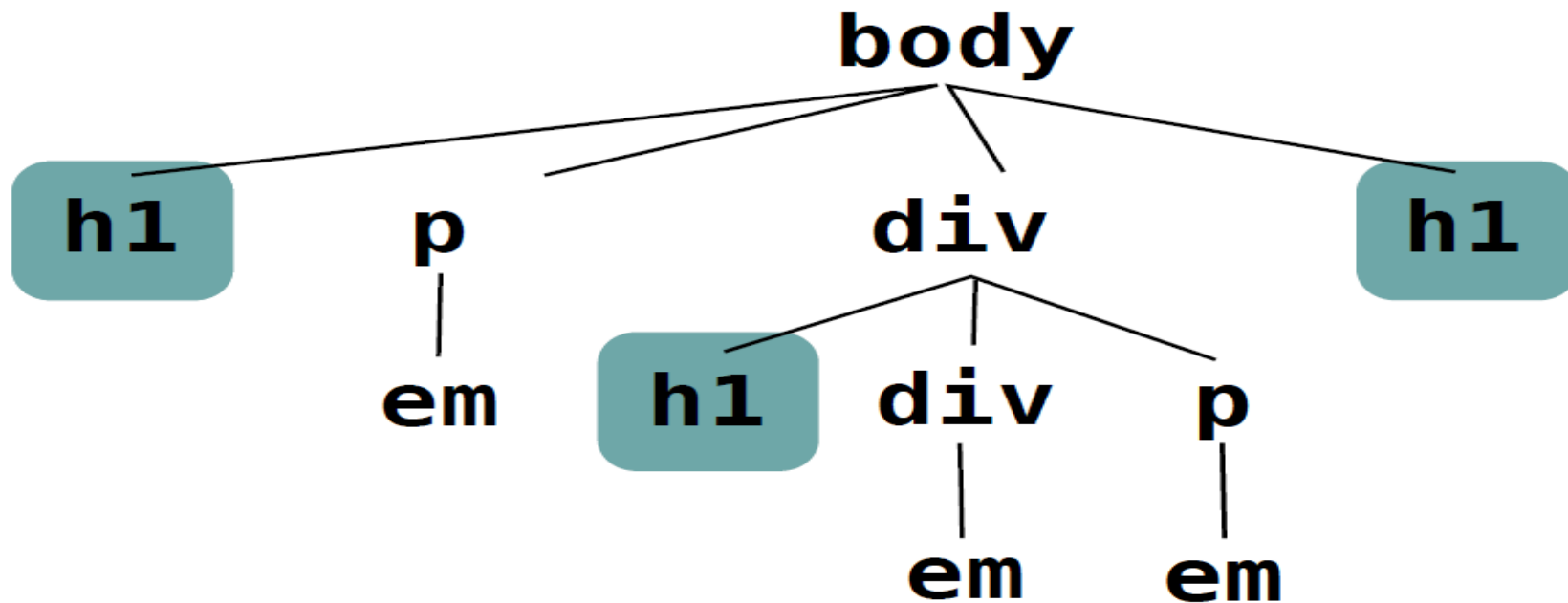
- Les évènements

Partie 4

- Manipuler le DOM

Node

Javascript propose des fonctionnalités pour manipuler la structure de l'arbre DOM :



Node

Javascript propose des fonctionnalités pour manipuler la structure de l'arbre DOM :

- Accès aux informations sur les nœuds de l'arbre

Node

Javascript propose des fonctionnalités pour manipuler la structure de l'arbre DOM :

- Accès aux informations sur les nœuds de l'arbre
- Création de nouveaux nœuds

Node

Javascript propose des fonctionnalités pour manipuler la structure de l'arbre DOM :

- Accès aux informations sur les nœuds de l'arbre
- Création de nouveaux nœuds
- Insertion, suppression, déplacement de nœuds dans l'arbre

Le type Node

Javascript propose des fonctionnalités pour manipuler la structure de l'arbre DOM :

- Accès aux informations sur les nœuds de l'arbre
- Création de nouveaux nœuds
- Insertion, suppression, déplacement de nœuds dans l'arbre

Toute manipulation de la structure de l'arbre a une répercussion immédiate sur le document affiché

Le type Node

Javascript propose des fonctionnalités pour manipuler la structure de l'arbre DOM :

- Accès aux informations sur les nœuds de l'arbre
- Création de nouveaux nœuds
- Insertion, suppression, déplacement de nœuds dans l'arbre

Toute manipulation de la structure de l'arbre a une répercussion immédiate sur le document affiché

type Node

Les nœuds de l'arbre DOM sont des objets de type Node.

Création de nœuds

`document.createElement(balise)`

Crée un nouveau nœud avec la balise donnée

`document.createTextNode(text)`

Crée un nouveau **nœud** texte avec pour contenu text (non interprété)

`node.cloneNode(bool)`

Crée un nouveau nœud copie de node
même attributs mais pas listeners, doit être ajouté au document

- `bool == true` clone également tous les descendants
- `bool == false` seul le nœud est cloné, pas ses descendants

Insertion de nœuds

```
noeudParent.insertBefore(nœud, noeudRéférence)
```

Insert nœud avant *noeudRéférence* comme fils de *noeudParent*

Nb: *noeudRéférence.parentNode* == *noeudParent*

```
parent.appendChild(nœud)
```

Nœud est ajouté à la fin des fils de parent

Nb : si le **nœud inséré** ou **ajouté** existe déjà dans le document, il est déplacé, donc **supprimé** de la position existante et inséré/ajouté à la position demandée

Suppression et remplacement de nœuds

```
parent.removeChild(nœud)
```

nœud est supprimé des fils de parent

```
parent.replaceChild(remplaçant, remplacé)
```

remplaçant prend la place de *remplacé* comme fils de parent

Nb : ces deux fonctions ont pour résultat le nœud supprimé/remplacé

L'objet Node

Propriété d'un objet Node

- ❑ `nodeName` : le nom du nœud
- ❑ `nodeType` : le type du nœud défini par des constantes nommées prédéfinies,
`node.ELEMENT_NODE` (= 1), `node.TEXT_NODE` (= 3)
- ❑ `nodeValue` : `null` si ce nœud est un nœud élément, le contenu pour un nœud texte
- ❑ `parentNode` : son nœud parent
- ❑ `childNodes` : la liste de ses nœuds enfants
- ❑ `firstChild`, `lastChild` : premier, dernier de ses nœuds enfants
- ❑ `previousSibling`, `nextSibling` : le nœud grère précédent ou suivant
- ❑ etc