DOM
Présentation

Document Object Model

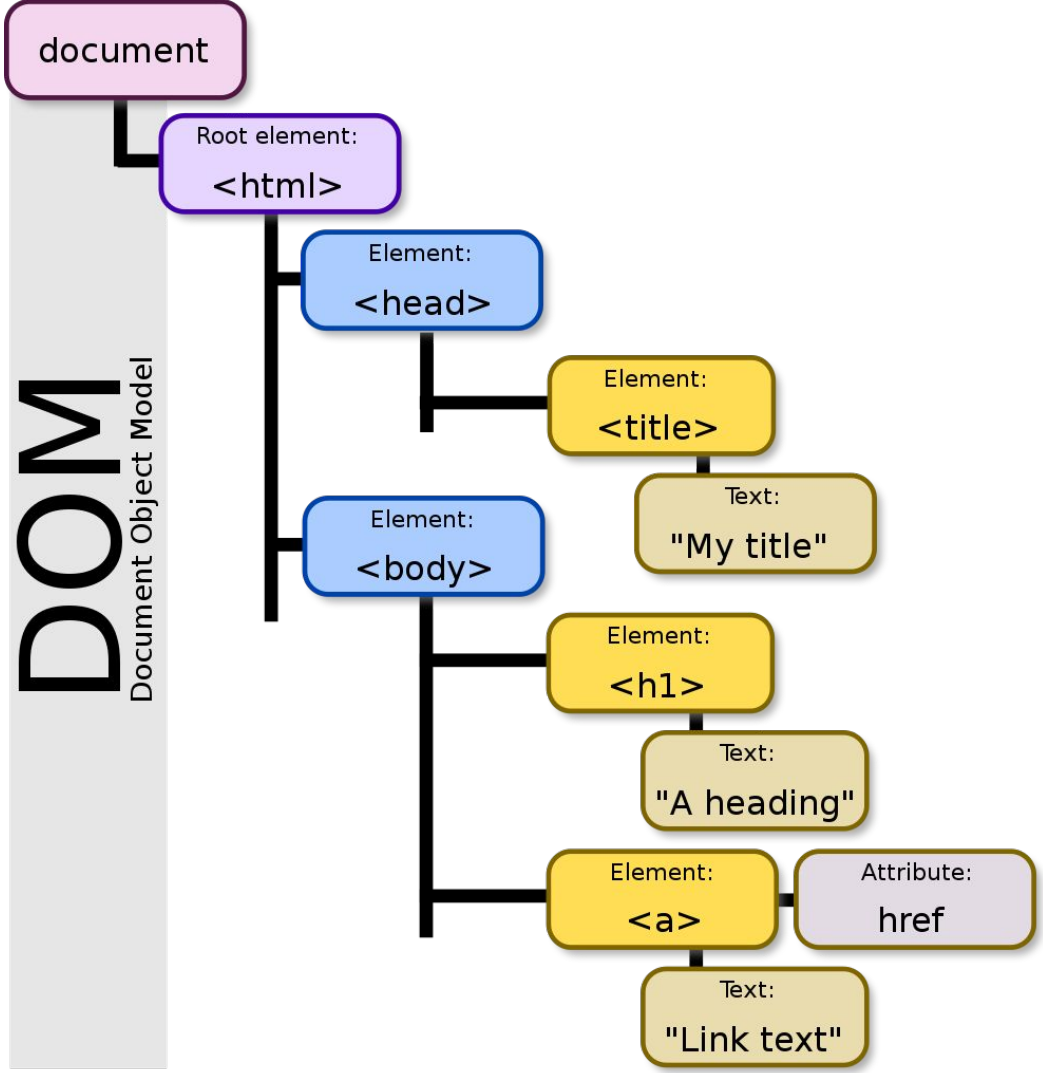
Permet d’interagir avec la structure et les éléments HTML

Exemples

→ Modification du style ou du contenu d’un élément

→ Insertion ou suppression d’un nouvel élément dans la page web

→ Interactions avec l’utilisateurs (clics, saisie d’un champ…)


DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text:
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribute:
href
Text:
"Link text"

Comment agir avec le DOM ?

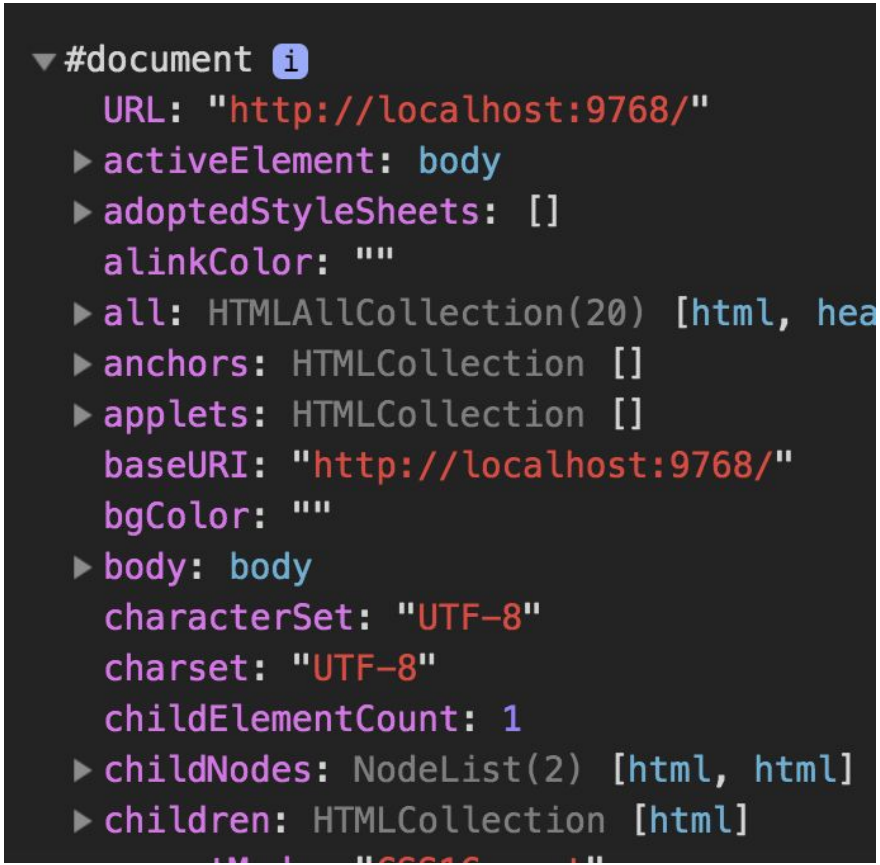
Pour comprendre comment agir avec le DOM,
il faut comprendre l'objet en JS

const obj = {} → Objet littéral en JS

```
const contact = {
  firstname: 'Coco',
  lastname: 'DURAND',
  age: 46,
  email: 'coco@mail.fr',
  man: true,
  address: {
    street: '13 rue de la Liberté',
    zip: '75019',
    city: 'Paris',
  }
}
```

Objet depuis la console du navigateur

Objet document depuis la console du navigateur



Objet document depuis la console du navigateur

```
document.body.firstElementChild
▶ <header>…</header>
document.body.firstChild
▶ #text
document.body.firstChild.textContent
"

  "
```

DOM
Propriétés à connaître

Recherche depuis la racine de la page

⇛ *document.<method|property>*

getElementById → recherche d'un élément HTML par son identifiant

getElementsByTagName → recherche d'éléments HTML par le nom de balise

getElementsByClassName → recherche d'éléments HTML par une classe

querySelector → recherche d'un élément HTML par sélecteur

querySelectorAll → recherche d'éléments HTML par sélecteur

Recherche depuis un élément HTML

⇛ *element.<method|property>*

children → les enfants de l'élément

parentElement → le parent direct de l'élément

nextElementSibling → le frère direct qui suit l'élément

previousElementSibling → le frère direct qui précède l'élément

Modifications du DOM

Modification de contenu ou de classe(s)

innerHTML, textContent

className, classList (.add, .remove, .contains, .toggle, .replace)

Modification de style ou d’attribut(s)

style, set|get|removeAttribute

Création / suppression / remplacement d’éléments

createElement PUIS append|remove|replaceChild

Programmation événementielle

Définition

Le programme réagit à des événements (survol de souris, appui sur une touche...)

Exemple du clic de souris

onclick="..." → mélange HTML/JS (non recommandé)

element.onclick → écrasement (overwriting) des définitions antérieures

element.addEventListener → respect des définitions antérieures

Saisies de l’utilisateur

Récupération des saisies depuis un élément de formulaire

onchange → détection du changement de valeur

onfocus / onblur → détection du focus / de la perte de focus

checked → état d’une case à cocher / d’un bouton radio

document.myForm.specificField PUIS **event.target.value** OU **element.value**

FIN