



DOM



- Document Object Model

API créé par Netscape en même temps que le JavaScript qui permet la manipulation du HTML en mémoire sous la forme d'un arbre

- W3C

Une norme existe depuis 1998, aujourd'hui dans sa 4e édition

DOM Level 4 : <http://www.w3.org/TR/dom/>

DOM Level 3 Events : <http://www.w3.org/TR/DOM-Level-3-Events/>

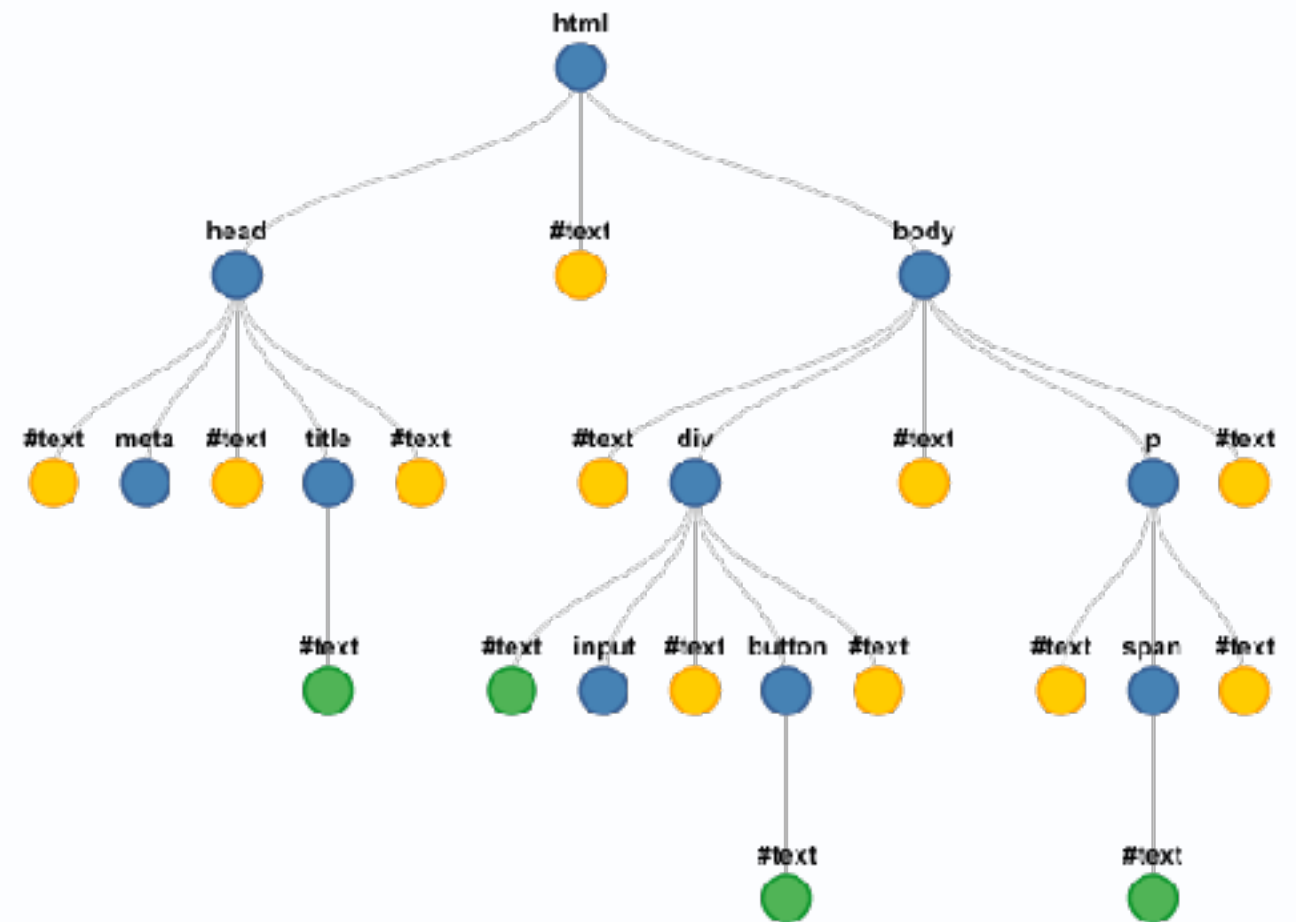
HTML5 : <http://www.w3.org/TR/html5/>

DOM - Arbre



- Représentation sous la forme d'un arbre
<https://github.com/bioub/d3.DOMVisualizer>

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Helloworld</title>
</head>
<body>
  <div>
    Prénom :
    <input name="prenom" id="prenom">
    <button id="clicme">Hello</button>
  </div>
  <p>
    Bonjour <span id="result"> </span>
  </p>
</body>
</html>
```



DOM - Level 1



- Helloworld en DOM 1 (années 1990)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Helloworld</title>
  <script>
    function helloworld() {
      var input = document.getElementsByName('prenom')[0];
      var span = document.getElementsByTagName('span')[0];

      span.firstChild.nodeValue = input.value;
    }
  </script>
</head>
<body>
<div>
  Prénom :
  <input type="text" name="prenom" id="prenom" onkeyup="helloworld()">
</div>
<p>
  Bonjour <span id="result"> </span>
</p>
</body>
</html>
```



- Helloworld en DOM 2 - 3 (années 2000)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Helloworld</title>
  <script>
    window.addEventListener('load', function() {
      var input = document.getElementById('prenom');
      var span = document.getElementById('result');

      input.addEventListener('input', function () {
        span.textContent = input.value;
      });
    });
  </script>
</head>
<body>
<div>
  Prénom :
  <input type="text" name="prenom" id="prenom">
</div>
<p>
  Bonjour <span id="result"> </span>
</p>
</body>
</html>
```



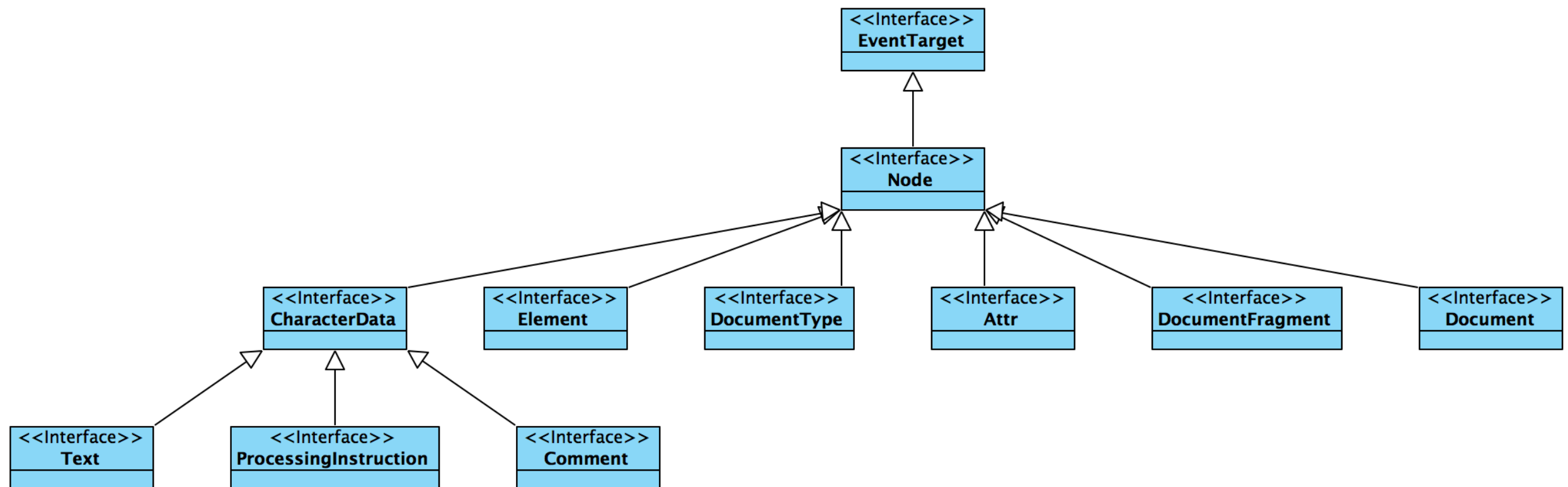
- Helloworld en DOM 4 (années 2010)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Helloworld</title>
</head>
<body>
<div>
  Prénom :
  <input type="text" name="prenom" id="prenom">
</div>
<p>
  Bonjour <span id="result"> </span>
</p>
<script>
  (function() {
    'use strict';

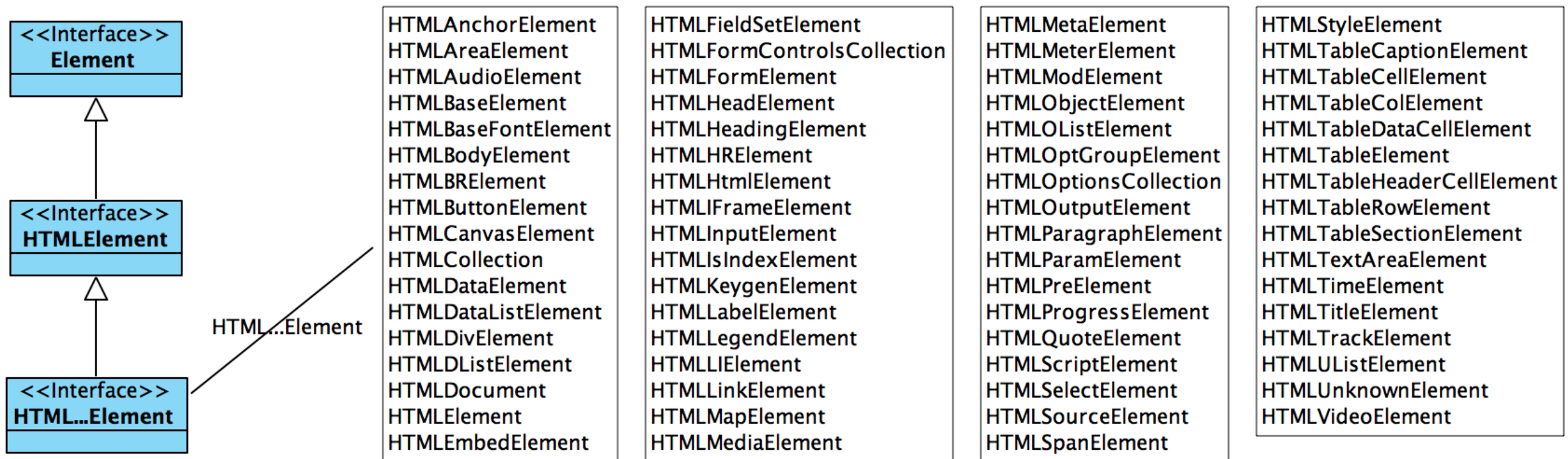
    var input = document.querySelector('input[type=text][name=prenom]');
    var span = document.querySelector('body > p span#result');

    input.addEventListener('input', function () {
      span.innerHTML = '<b>' + input.value + '</b>';
    });
  }());
</script>
</body>
</html>
```

DOM - Interfaces




DOM - HTML Interfaces





- Depuis n'importe quel noeud Element (DOM 4, IE9+) :
 - children : les noeuds Element enfants
 - firstElementChild : le premier noeud Element enfant
 - lastElementChild : le dernier noeud Element enfant
 - previousElementSibling : le frère Element précédent (même parent)
 - nextElementSibling : le frère Element suivant (même parent)
 - parentElement : le parent

Browser compatibility

Desktop	Mobile				
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support (on Element)	1.0	3.5 (1.9.1)	9.0	10.0	4.0
Support on Document and DocumentFragment 	29.0	25.0 (25.0)	Not supported	16.0	Not supported



- Rechercher le premier noeud qui matche un sélecteur CSS
 - `document.querySelector('selecteur')`
 - `element.querySelector('selecteur')`
- Rechercher tous les éléments matchent un sélecteur CSS (retourne un NodeList, itérable)
 - `document.querySelectorAll('selecteur')`
 - `element.querySelectorAll('selecteur')`



- Tous les éléments : `document.all`
- Body : `document.body`
- Head : `document.head`
- Forms : `document.forms[position]` ou `document.forms[id]`
- Images : `document.images[position]` ou `document.images[id]`
- Scripts : `document.scripts[position]` ou `document.scripts[id]`
- Title : `document.title`



- Entre la balise ouvrante / fermante
 - `Element.textContent`
 - `Element.innerText`
 - `Element.innerHTML`
- Valeur d'un champs (input, select, textarea)
 - `Element.value`



- Sauf exception : les propriétés d'un Element porte le nom de l'attribut
- Ex : Element.id, HTMLElement.lang, HTMLElement.title, HTMLFormElement.action, HTMLFormElement.name...
- Exceptions :
 - Element.className (class est un mot clé de JS),
 - HTMLMetaElement n'a pas de propriété charset
 - ...
- Pour accéder à n'importe quel attribut :
 - Element.getAttribute(name)
 - Element.setAttribute(name, value)

DOM - Attributs booléens



- Certains attributs ont un caractère booléen
- En HTML5 :
 - `<input type="text" required>`
- En XHTML
 - `<input type="text" required="required">`
- DOM :
 - `Element.required (true/false)`



- Pour stocker une valeur dans une balise dans un attribut "custom"
- DOM
 - `Element.dataset.monAttribut = "Valeur"`
- HTML
 - `<balise data-mon-attribut="valeur">`



- Documentation
- DevDocs
<http://devdocs.io/dom/>