

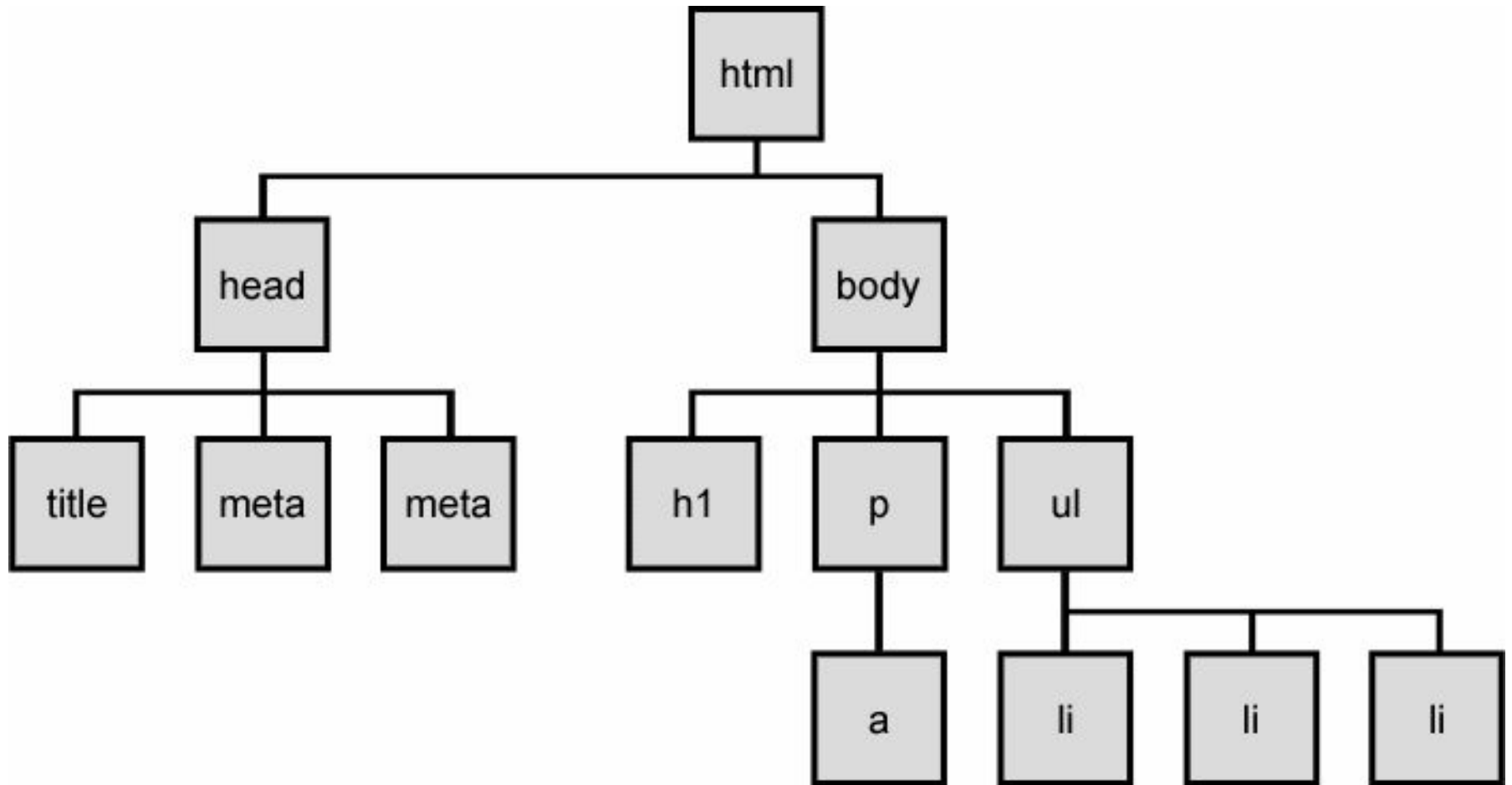
Lección 25

Manipulación del DOM II

Navegación del DOM



Navegación del DOM: Parentezco de elementos



Navegación en el DOM: Parentezco de elementos

.parentNode: Selecciona al elemento padre de otro elemento.

```
function seleccionar_padre() {  
    var hijo = document.getElementById('parrafo');  
    alert(hijo.parentNode);  
}
```

.firstChild y .firstElementChild

Retorna el primer nodo hijo del nodo especificado.

La diferencia entre ellas es que **.firstChild** retorna el primer nodo hijo, ya sea elemento o texto (dependiendo cual está primero), mientras que **.firstElementChild** retorna el primer nodo de tipo elemento.

```
document.getElementById('otro-parrafo').firstChild;
```

.lastChild y .lastElementChild

La propiedad **.lastChild** funciona exactamente como `firstChild`, pero se refiere el último de los hijos de un elemento. Se aplican, por tanto, las mismas indicaciones anteriores.

```
document.getElementById('mi-caja').lastChild;
```

.nextSibling/nextElementSibling

Gracias a **.nextSibling**, lo que podemos seleccionar es el siguiente hermano de un elemento.

Se aplican las mismas limitaciones que para las dos propiedades anteriores.

```
document.getElementById('box1').nextSibling;
```

.previousSibling/previousElementChildren

.previousSibling funciona igual que nextSibling, pero selecciona el hermano anterior de un elemento.

Igual para sus limitaciones.

```
document.getElementById('box2').nextSibling;
```


Modificación de atributos



Modificación de atributos

.getAttribute: Devuelve el valor del atributo especificado en el elemento. Si el atributo especificado no existe, el valor retornado puede ser tanto null como " " (una cadena vacía).

```
var atributo = element.getAttribute(nombreAtributo);
```

.setAttribute: Agrega un nuevo atributo o cambia el valor de un atributo en un elemento especificado.

```
element.setAttribute(name, value);
```

Modificación de atributos

.hasAttribute: Devuelve verdadero si existe el atributo especificado, de lo contrario, devuelve false.

```
element.hasAttribute( attributename );
```

.removeAttribute: Elimina un atributo del elemento especificado.

```
element.removeAttribute( attributename );
```

Modificación de atributos

.classList: Esta propiedad retorna el nombre de la clase de un elemento. Esta propiedad es usada para agregar, remover y switchear clases de elementos CSS.

```
element.classList
```

Ejemplos



Al elemento con el id "myDiv" le queremos agregar la clase "myClass".

```
document.getElementById("myDiv").classList.add("myClass");
```

Ahora queremos remover esa clase:

```
document.getElementById("myDiv").classList.remove("myClass");
```

Después usaremos **toggle** para switchear la clase "newClass", esto nos permitirá agregar y quitar nuestra nueva clase.

```
document.getElementById("myDiv").classList.toggle("newClass");
```