



WebDesign

HERZLICH WILLKOMMEN !



■ JavaScript im Server-seitigem Stil



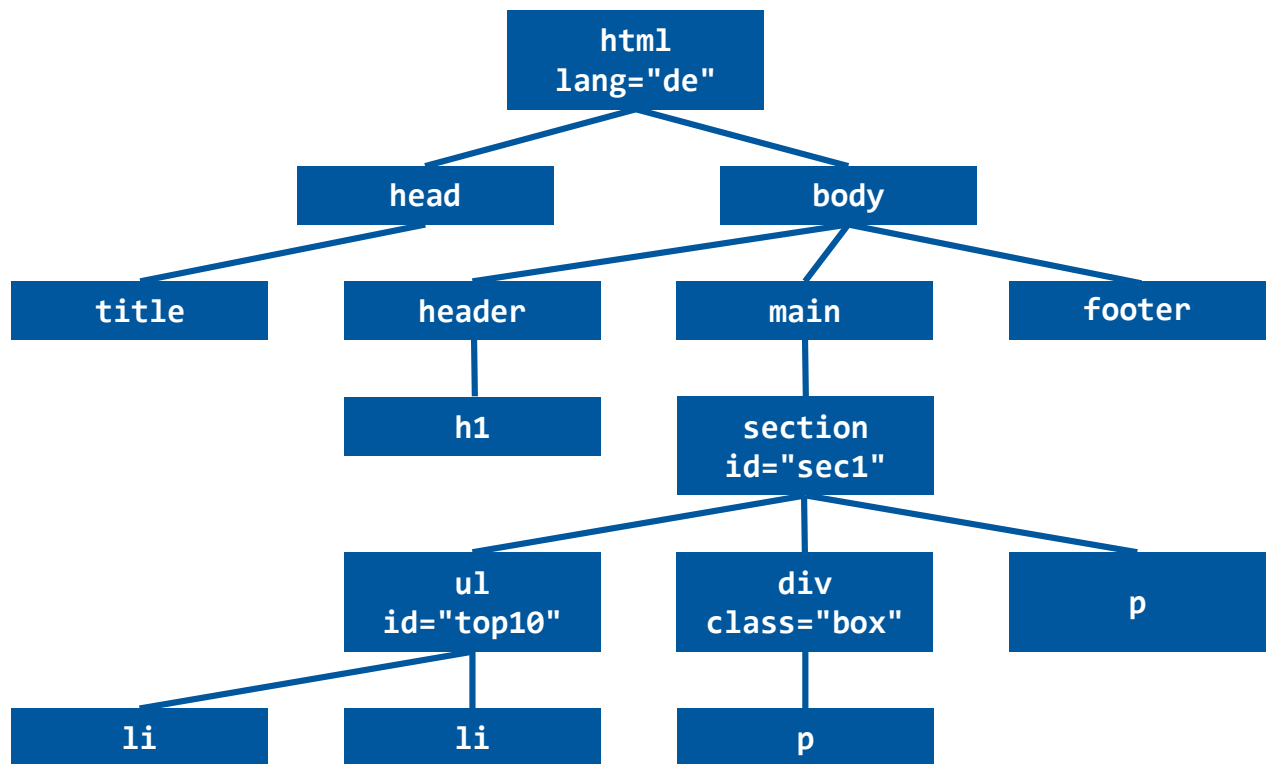
Server-seitiger Stil

- `<script> code </script>`
- `document.write()`

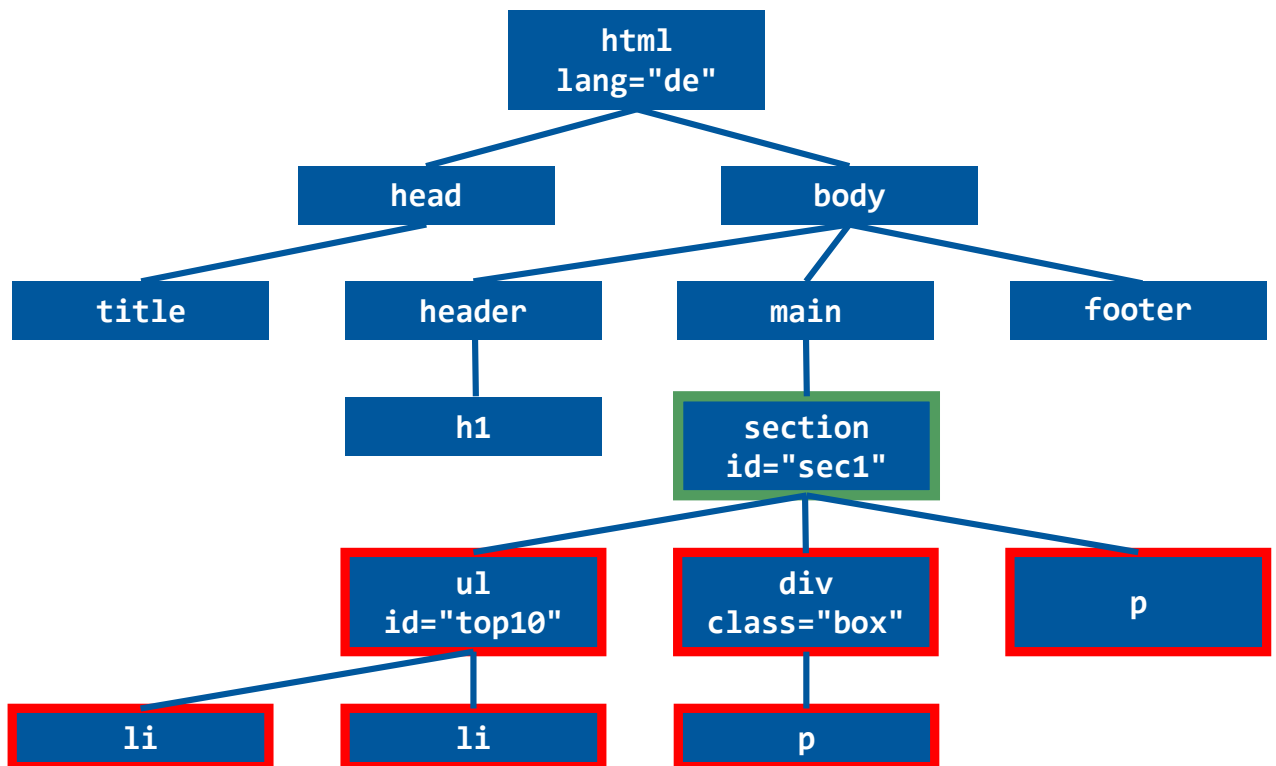
- `<script defer src="deferred.js"></script>`
- `<script async src="async.js"></script>`
- Das defer-Attribut weist den Browser an, die Ausführung des Skripts aufzuschieben, bis das Dokument vollständig geladen und geparkt wurde und manipuliert werden kann.
- Das async-Attribut veranlasst den Browser, das Skript so schnell wie möglich auszuführen, das Parsen des Dokuments aber fortzusetzen, während das Skript heruntergeladen wird.
- Kommen in einem `<script>`-Tag beide Attribute vor, hat das async-Attribut Vorrang.

■ JavaScript DOM

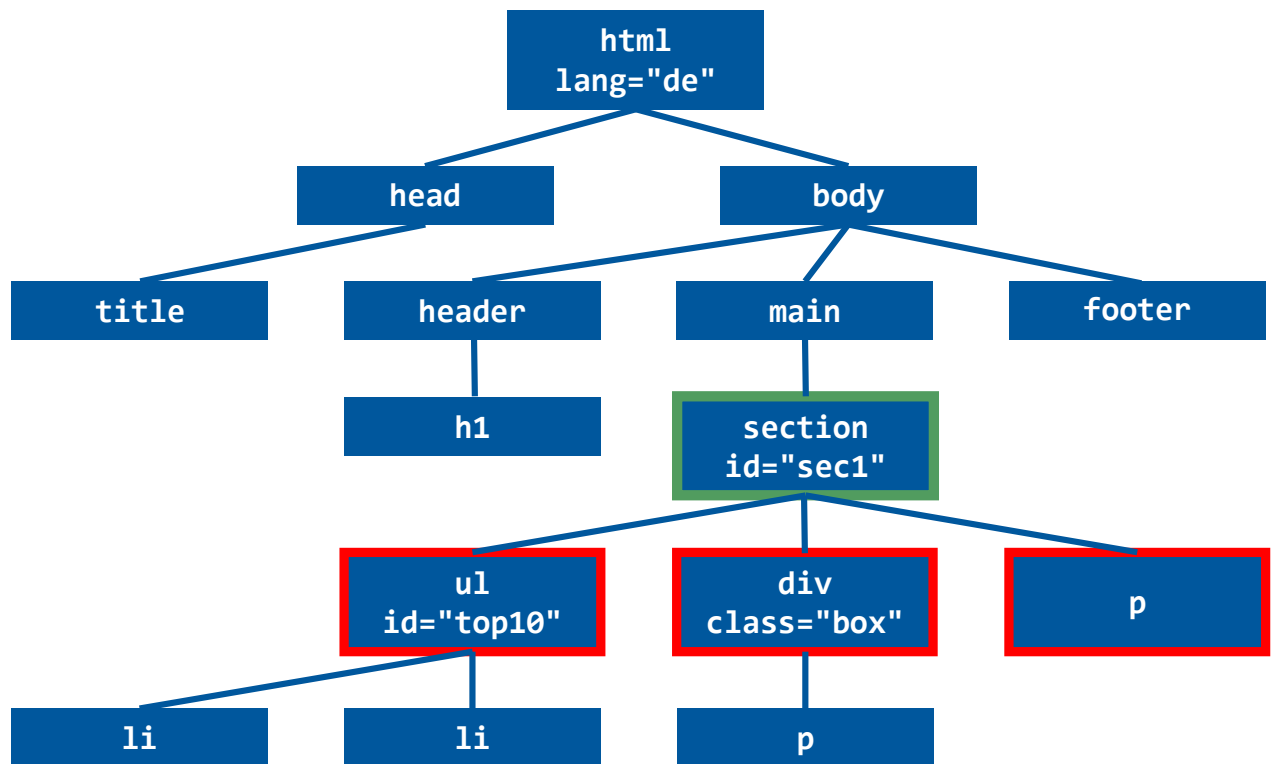




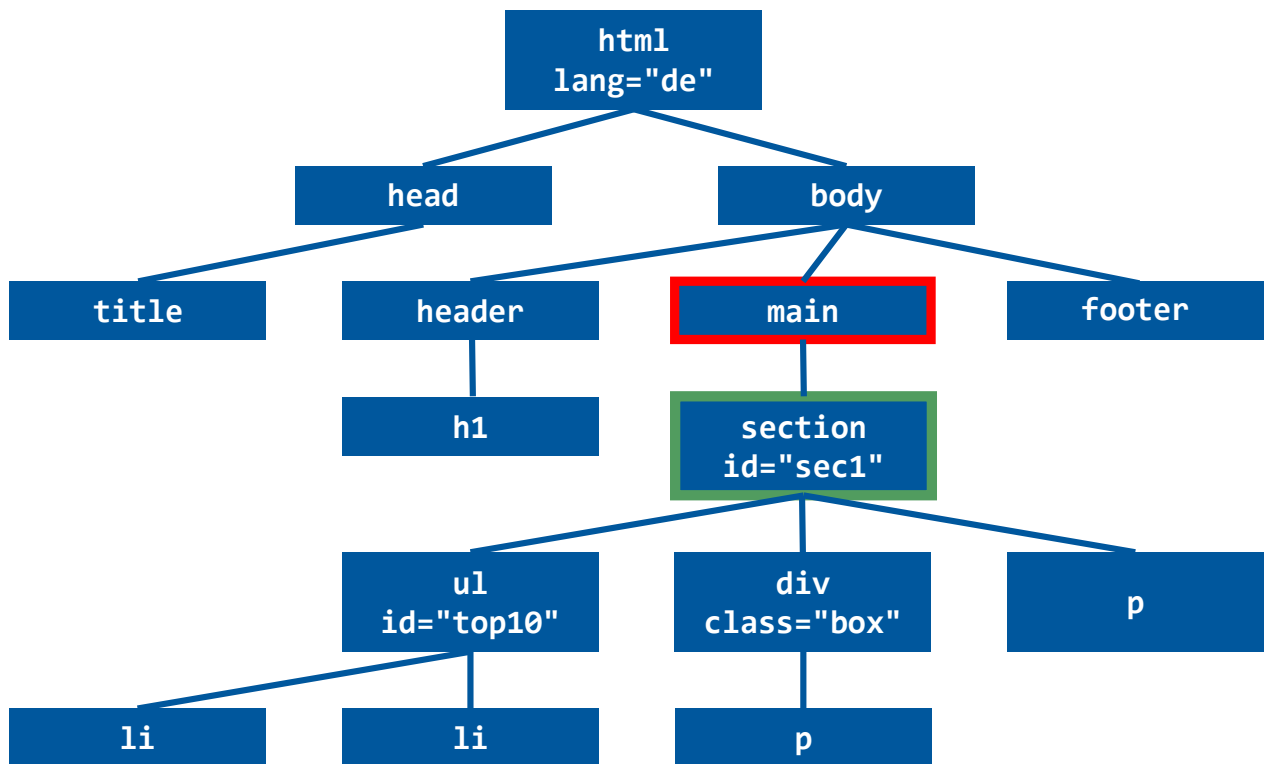
Nachfahren (*descendants*)

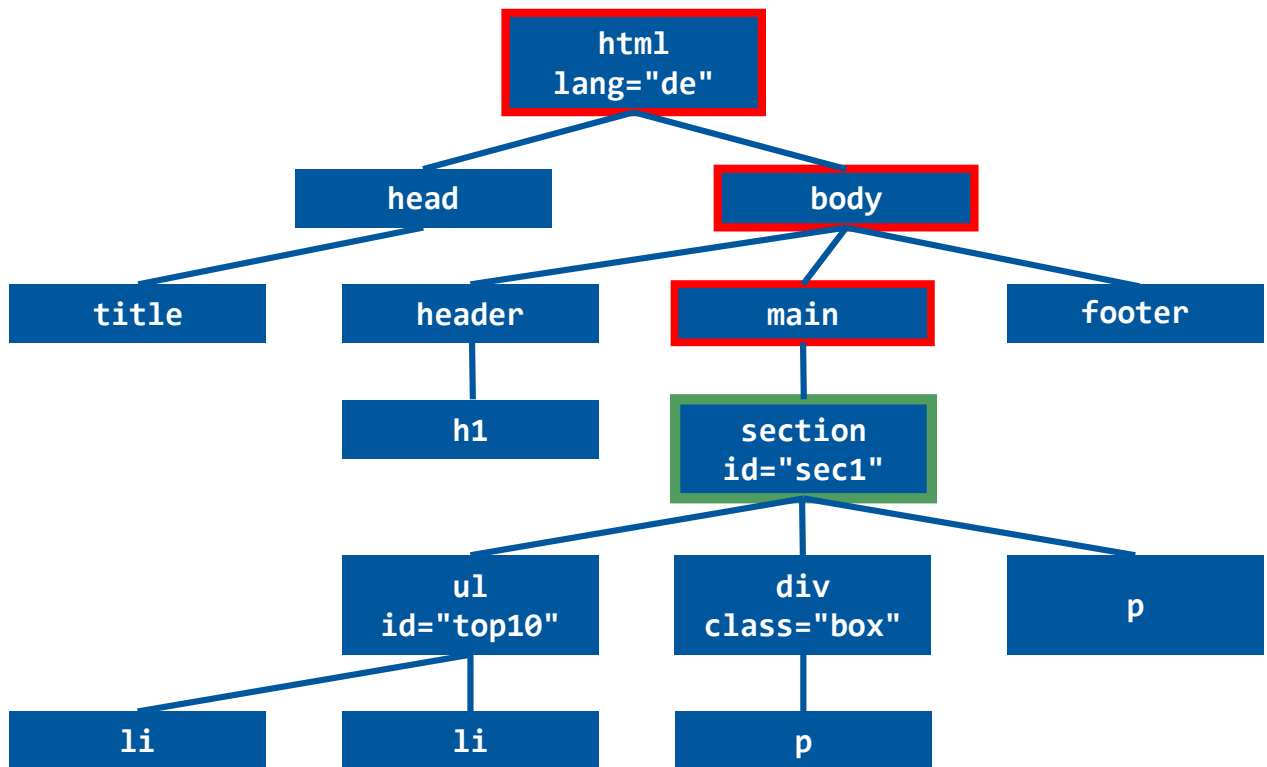


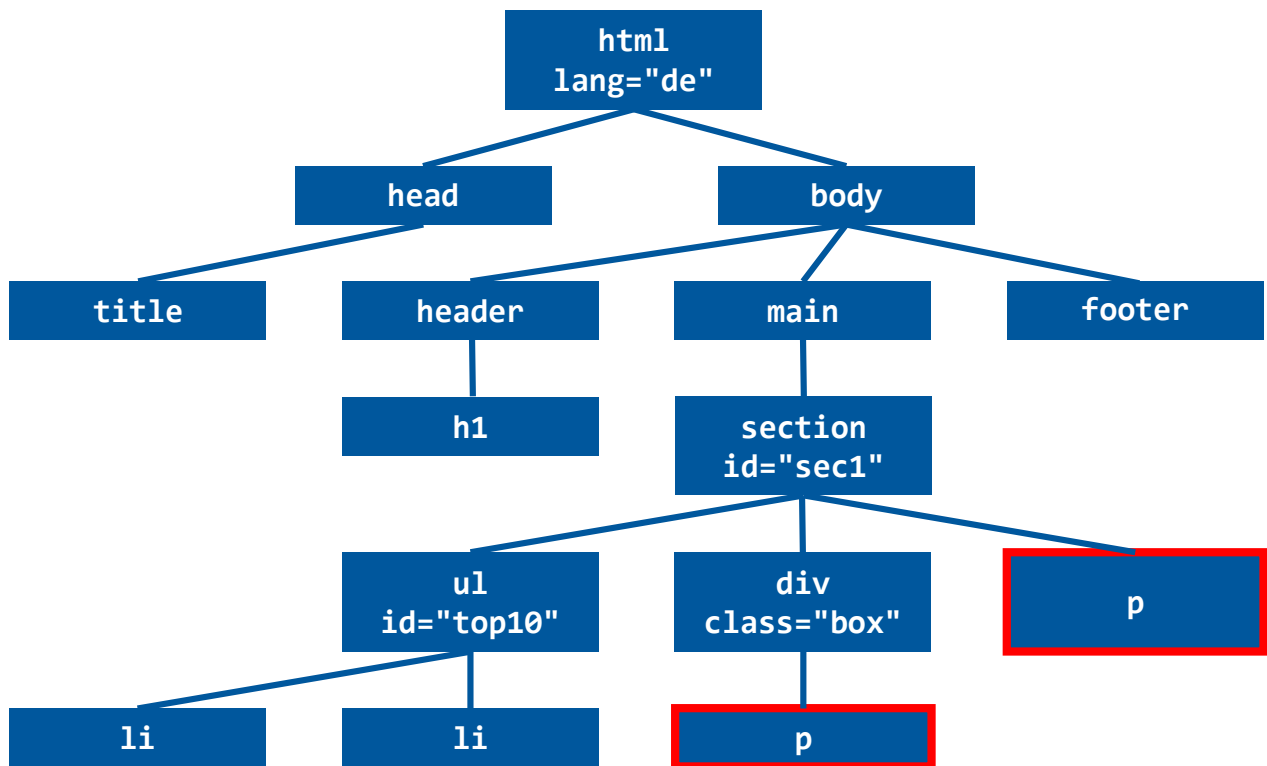
Kinderelemente (*children*)

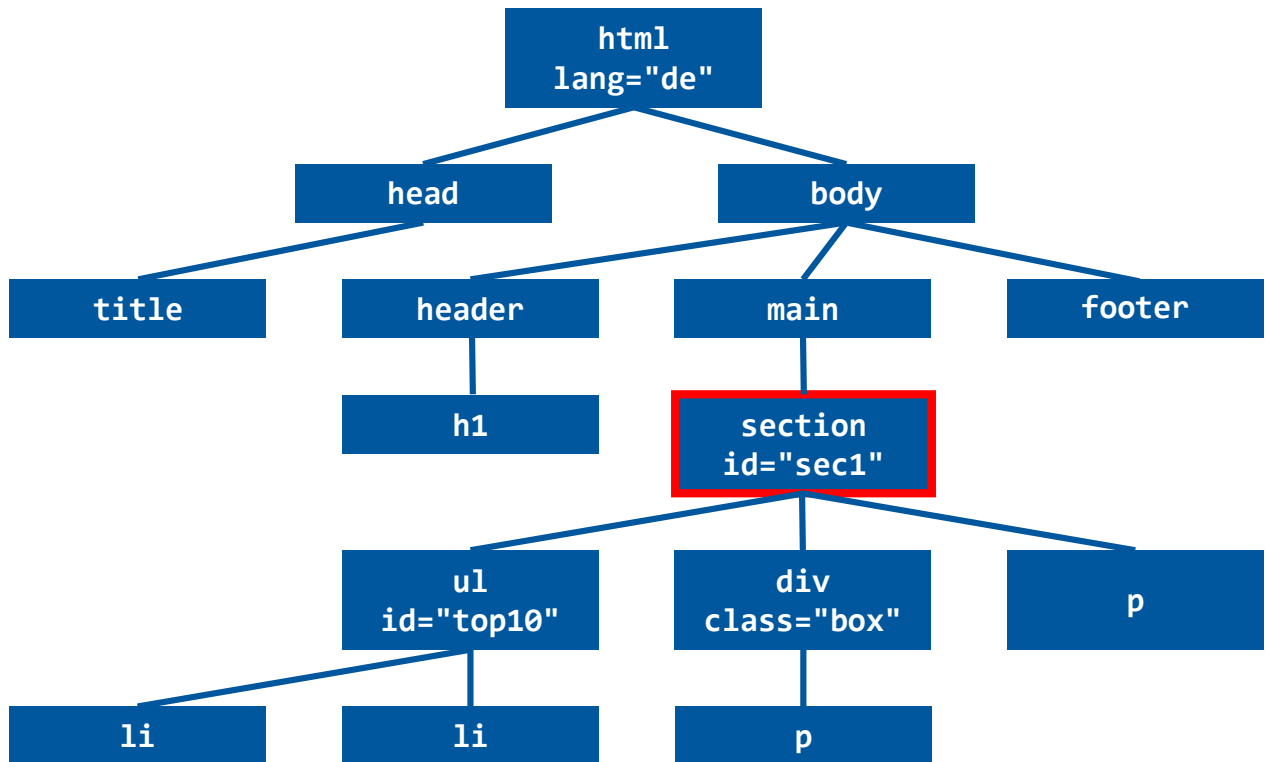


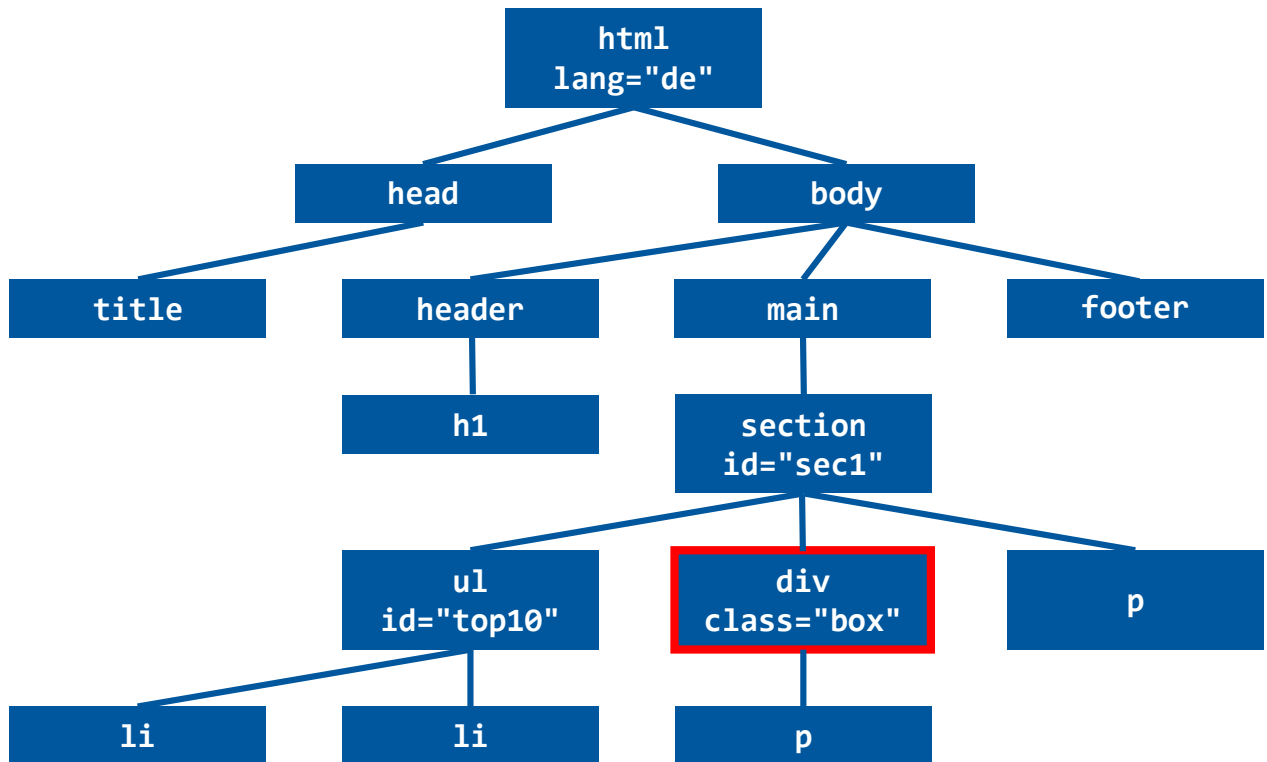
Elternelement (*parent*)











Selektiert nach Klasse.

```
let elem = document.querySelector("CSSSelector");
```

```
let sect1 = document.getElementById("sect1");
```

```
document.body
```

```
document.body.innerHTML = "<h1>Oops</h1>";
```

```
function toggleTheme() {  
    let lightTheme = document.querySelector("#light-theme");  
    let darkTheme = document.querySelector("#dark-theme");  
    if (darkTheme.disabled) {  
        // Derzeit hell, auf dunkel schalten.  
        lightTheme.disabled = true;  
        darkTheme.disabled = false;  
    } else { // Derzeit dunkel, auf hell schalten.  
        lightTheme.disabled = false;  
        darkTheme.disabled = true;  
    }  
}
```



```
function setTheme(name) {  
    let link = document.createElement("link");  
    link.id = "theme";  
    link.rel = "stylesheet";  
    link.href = `themes/${name}.css`;  
    let currentTheme = document.querySelector("#theme");  
    if (currentTheme) {  
        currentTheme.replaceWith(link);  
    } else {  
        document.head.append(link);  
    }  
}
```

■ Events



Geräte-abhängige Events

- Diese Events sind direkt an bestimmte Eingabegeräte gebunden, wie z.B. Mäuse oder Tastaturen. Dazu gehören Eventtypen wie mousedown, mousemove, mouseup, touchstart, touchmove, touchend, keydown und keyup.

Geräte-unabhängige Events

- pointerdown, pointermove und pointerup, click

logische Benutzer-Aktionen

- focus-Event (wenn ein Texteingabefeld den Tastaturfokus erhält)
- change-Event (wenn der Benutzer den von einem Formularelement angezeigten Wert ändert)
- submit-Event (wenn der Benutzer in einem Formular auf eine Senden-Schaltfläche klickt).

- load und DOMContentLoaded, die auf den Window- bzw. Document-Objekten am Ende des Ladevorgangs eines Dokuments ausgelöst werden
- online- und offline-Events, wenn sich die Konnektivität der Netzwerkverbindung ändert
- popstate

- Medien: waiting, playing, seeking, volumechange
- Asynchrone Web-APIs (die aus einer Zeit stammen, als es in JavaScript noch keine Promises gab)
- IndexedDB-API: success- und error-Events

■ Einbindung CSS




```
selektor {  
    eigenschaft: wert;  
    eigenschaft: wert wert;  
}  
  
/* kommentar */  
selector {  
    eigenschaft: wert;  
}
```

In geschweiften Klammern stehen sogenannte Deklarationen.

```
<html>
  <head>
    <link rel="stylesheet" href="css/meineStile.css">
  </head>
  . . .
```

```
<html>
  <head>
    <style>
      p {color: red;}
    </style>
  </head>
```

```
<html>
  <head>
  </head>
  <body>
    <p style="color: red;">Hallo!</p>
    <p style="color: red; background-color: gray">
      Hallo nochmals!</p>
  </body>
</html>
```

Übersicht Einbindungsoptionen

- Externes CSS:
 - Einheitlicher Stil für ein ganzes Unternehmen einfach zu erreichen
 - Evtl. Abhängigkeit zu externen Quellen
 - Anpassungen an einer Stelle
- Eingebettetes CSS:
 - Übersichtlich
 - Schwer zu pflegen, wenn viele HTML-Dateien
- Inline:
 - Nur vereinzelt vertretbar