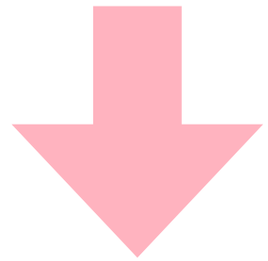
Webontwikkeling 4

JavaScript (JS)

Elke Steegmans

HTML



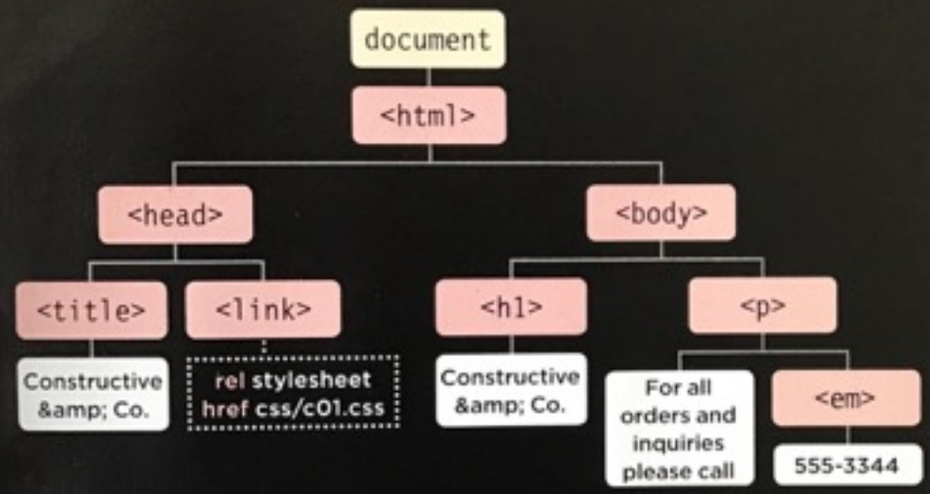
DOM

RENDER

```
<!DOCTYPE html>
<html>
  <head>
    <title>Constructive &amp;amp; Co.</title>
    <link rel="stylesheet" href="css/c01.css" />
  </head>
  <body>
    <h1>Constructive &amp;amp; Co.</h1>
    <p>For all orders and inquiries please call
      <em>555-3344</em></p>
  </body>
</html>
```

1

The browser receives an HTML page.



2

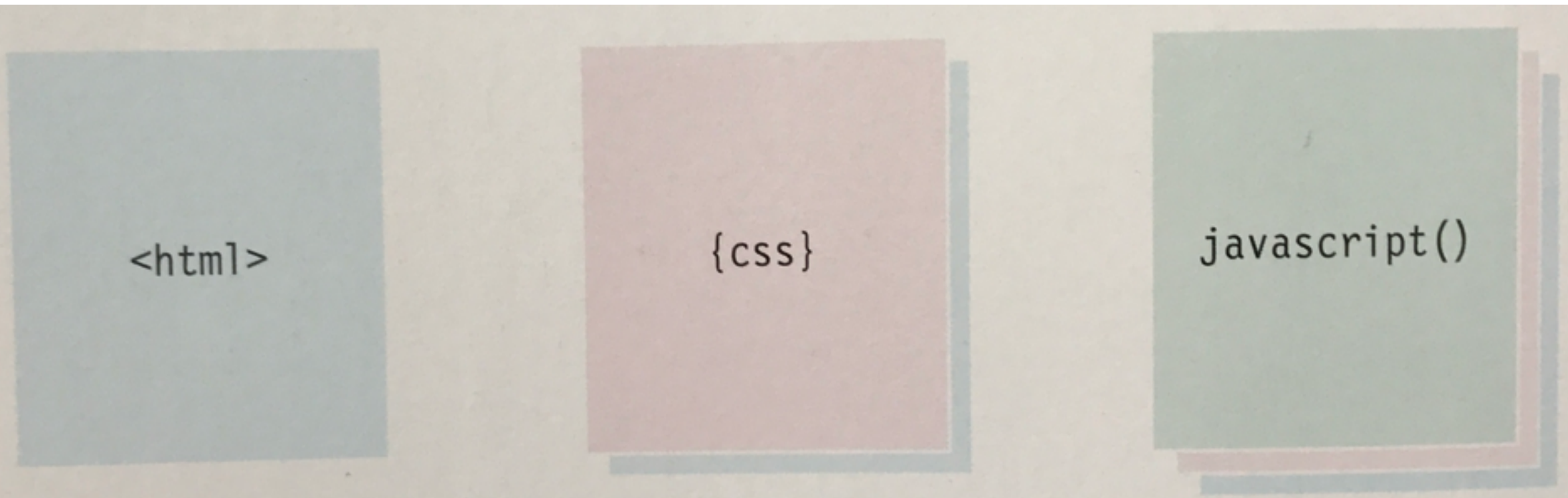
It creates a model of the page and stores it in memory.

- OBJECT
- ELEMENT
- TEXT
- ATTRIBUTES

3

It shows the page on screen using a rendering engine.

HTML, CSS en JavaScript



CONTENT LAYER

`.html` files

This is where the content of the page lives. The HTML gives the page structure and adds semantics.

PRESENTATION LAYER

`.css` files

The CSS enhances the HTML page with rules that state how the HTML content is presented (backgrounds, borders, box dimensions, colors, fonts, etc.).

BEHAVIOR LAYER

`.js` files

This is where we can change how the page behaves, adding interactivity. We will aim to keep as much of our JavaScript as possible in separate files.

HTML ONLY

Starting with the HTML layer allows you to focus on the most important thing about your site: its content.

Being plain HTML, this layer should work on all kinds of devices, be accessible to all users, and load quite quickly on slow connections.

HTML+CSS

Adding the CSS rules in a separate file keeps rules regarding how the page looks away from the content itself.

You can use the same style sheet with all of your site, making your sites faster to load and easier to maintain. Or you can use different style sheets with the same content to create different views of the same data.

HTML+CSS+JAVASCRIPT

The JavaScript is added last and enhances the usability of the page or the experience of interacting with the site.

Keeping it separate means that the page still works if the user cannot load or run the JavaScript. You can also reuse the code on several pages (making the site faster to load and easier to maintain).

Demo

UNTYPED LANGUAGE

```
var price;

var userName;

var inStock;

var colors;

price = 5;

userName = 'Elke';

inStock = true;

colors = ['white', 'pink', 'black'];
```

Function Declaration

```
function area (width, height ) {
   return width*height;
}


var size = area(3, 4);
```

Function Expression or Anonymous Function

```
var area = function(width, height ) {
   return width*height;
};


var size = area(3, 4);
```

Demo

OBJECTS

```
var hotel = {
  name : ‘Quay’,
  rooms : 40,
  booked : 25,
  checkAvailability : function () {
    return this.rooms - this.booked;
  }
};
var hotelName = hotel.name; // or var hotelName = hotel[‘name’];
```

OBJECTS

```
var hotel =  new Object();

hotel.name = 'Quay';

hotel.rooms = 40;

hotel.booked = 25;

hotel.checkAvailability = function () {

    return this.rooms - this.booked;

  }

};
```

OBJECTS

```
function hotel (name, rooms, booked) {

   this.name = name;

   this.rooms = rooms;

   this.booked = booked;

   this.checkAvailability = function () {

      return this.rooms - this.booked;

   };

};


var quayHotel = new Hotel('Quay', 40, 25);

var parkHotel = new Hotel('Park', 120, 77);
```

ARRAYS

- arrays are objects

```
var costs = {
    room1: 420,
    room2: 460
};
var priceRoom1 = costs['room1'];
```

- objects are arrays

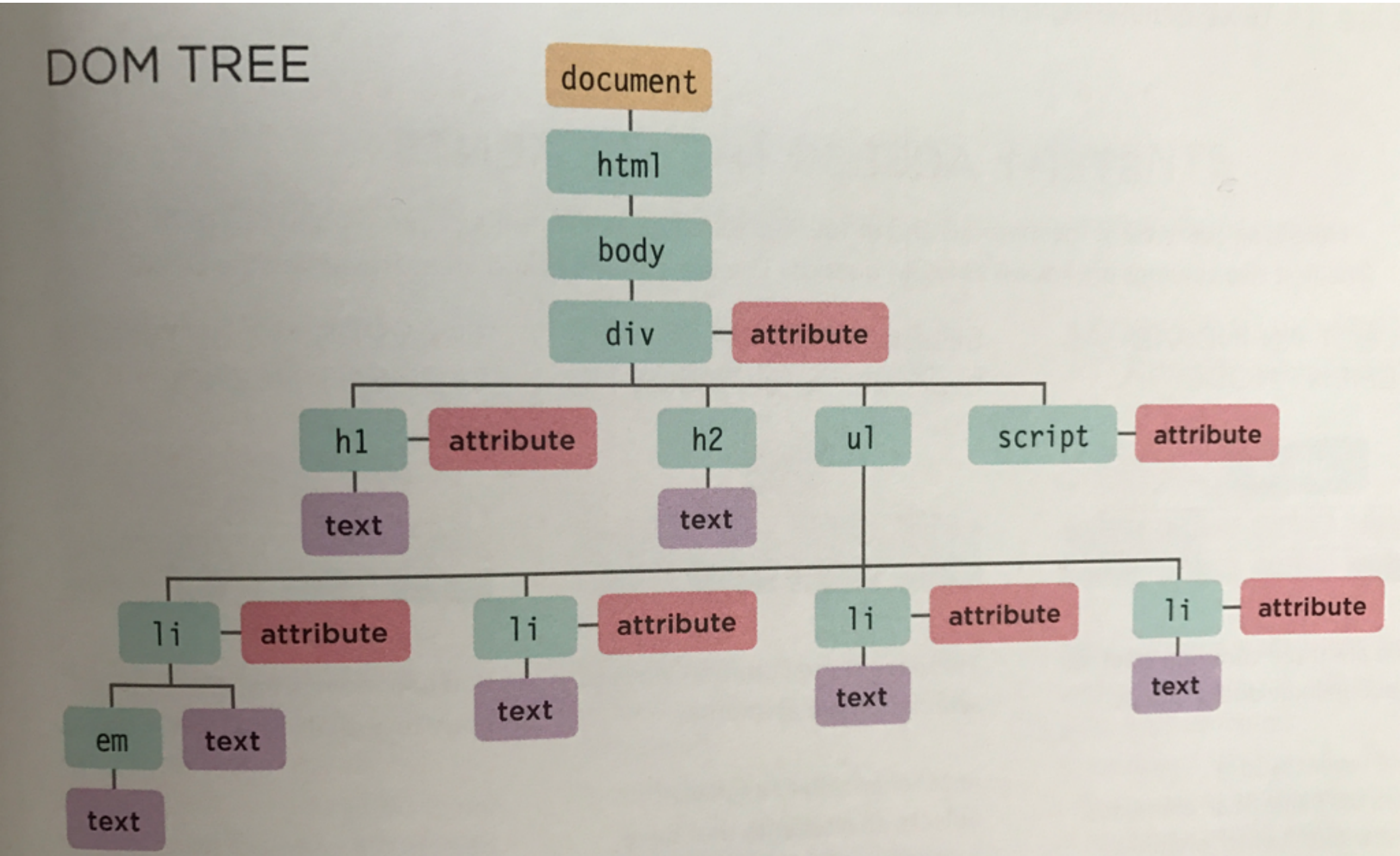
Demo

DOM

BODY OF HTML PAGE

```
<html>
  <body>
    <div id="page">
      <h1 id="header">List</h1>
      <h2>Buy groceries</h2>
      <ul>
        <li id="one" class="hot"><em>fresh</em> figs</li>
        <li id="two" class="hot">pine nuts</li>
        <li id="three" class="hot">honey</li>
        <li id="four">balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
```

DOM


DOM TREE
document
html
body
div
attribute
h1
attribute
text
h2
text
ul
script
attribute
li
attribute
em
text
text
li
attribute
text
li
attribute
text
li
attribute
text

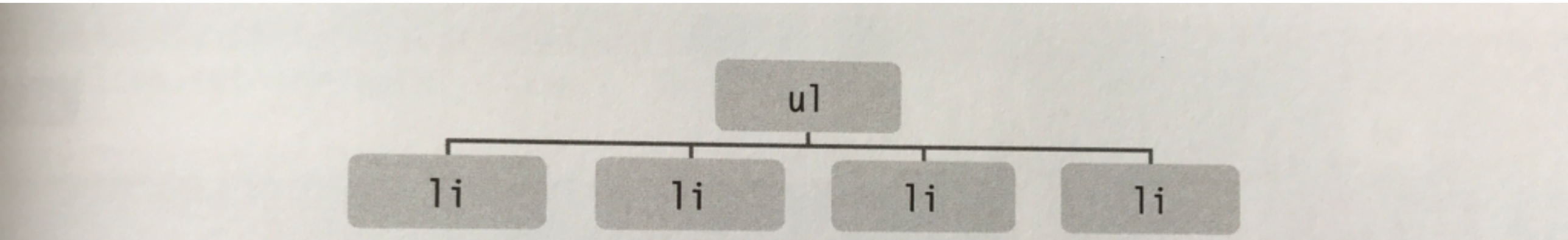
ACCESSING ELEMENTS

- getElementById('id')
  - getElementById('one')
- getElementsByTagName('tagname')
  - getElementsByTagName('li')
- getElementsByClassName('class')
  - getElementsByClassName('error')

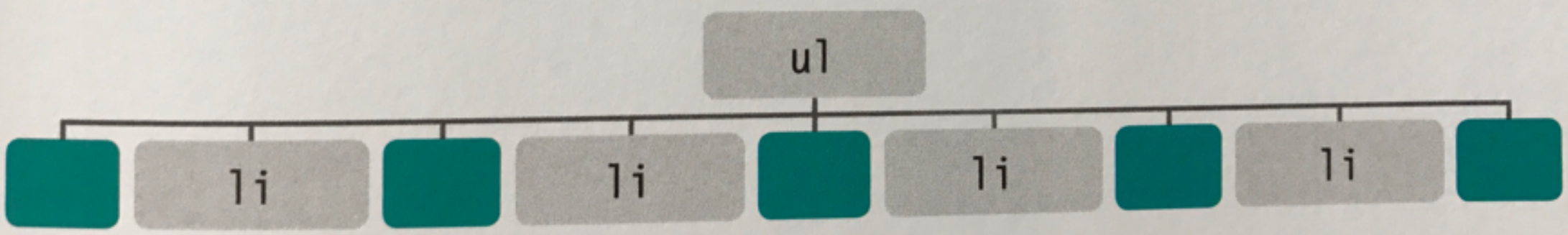
ACCESSING ELEMENTS

- querySelector('css selector')
  - querySelector('li.error')
- querySelectorAll('css selector')
  - querySelector('li.correct')

DOM



Internet Explorer (shown above) ignores whitespace and does not create extra text nodes.



Chrome, Firefox, Safari, and Opera create text nodes from whitespace (spaces and carriage returns).

Demo

EVENTS

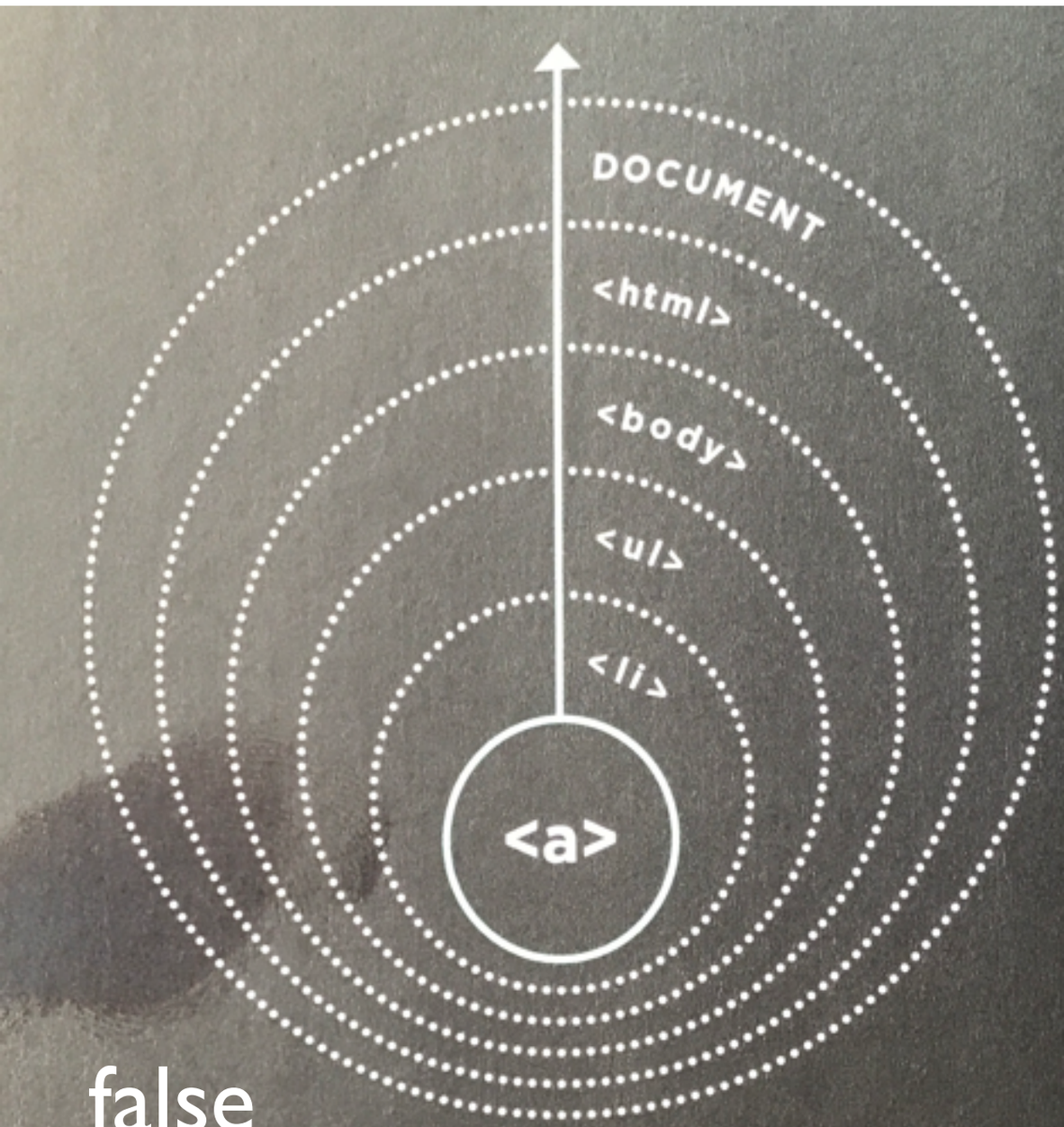
- click
- submit
- load
- mouseover
- keyup
- focus
- …

EVENTS

- HTML event handler attributes
  - do not use
- Using DOM event handlers
- Using event listeners

Demo

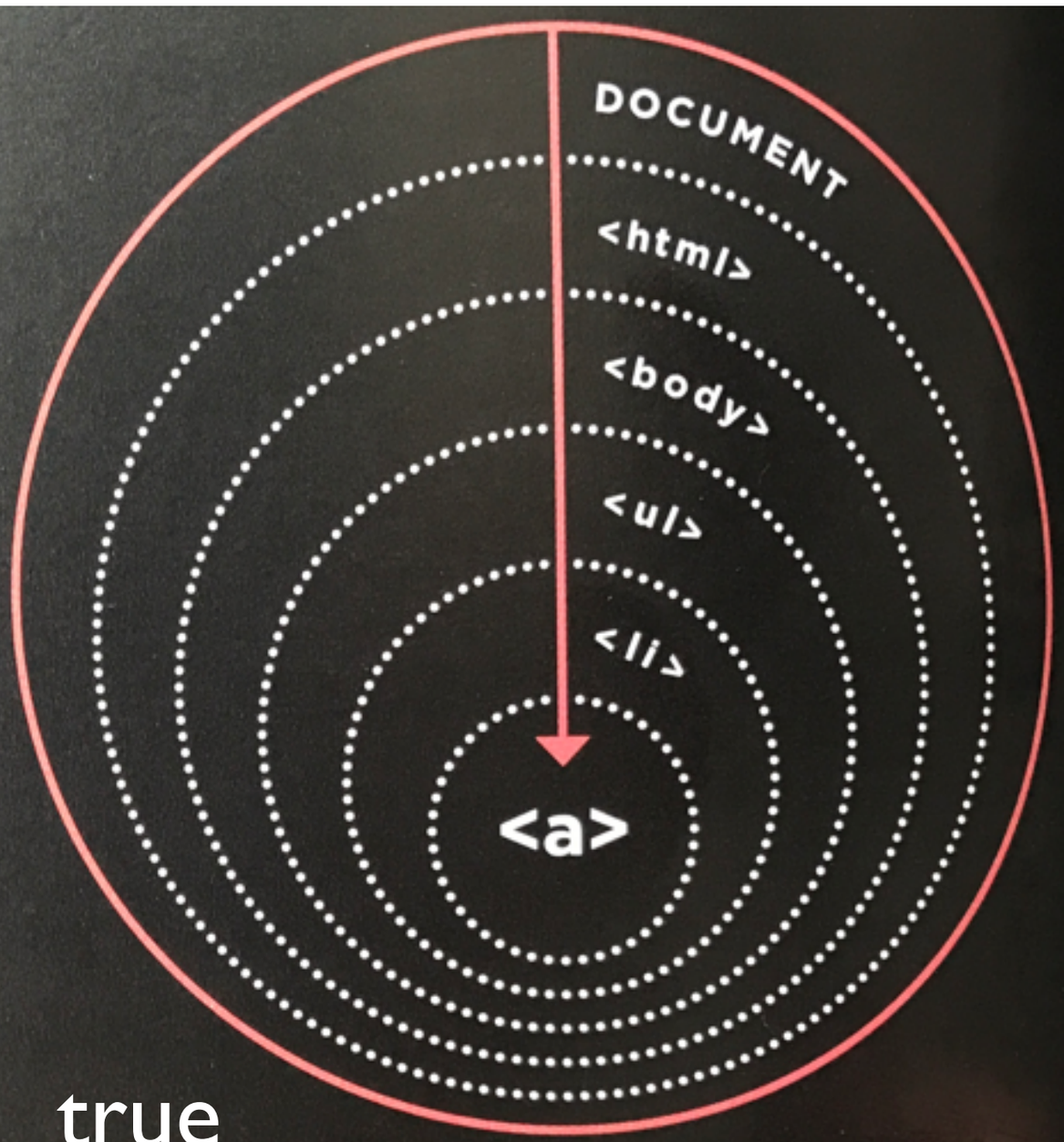
Event Flow



false

EVENT BUBBLING

The event starts at the *most* specific node and flows outwards to the *least* specific one. This is the default type of event flow with very wide browser support.



true

EVENT CAPTURING

The event starts at the *least* specific node and flows inwards to the *most* specific one. This is not supported in Internet Explorer 8 and earlier.

Demo

REFERENTIES

- JAVASCRIPT&JQUERY interactive front-end web development, Jon Duckett