

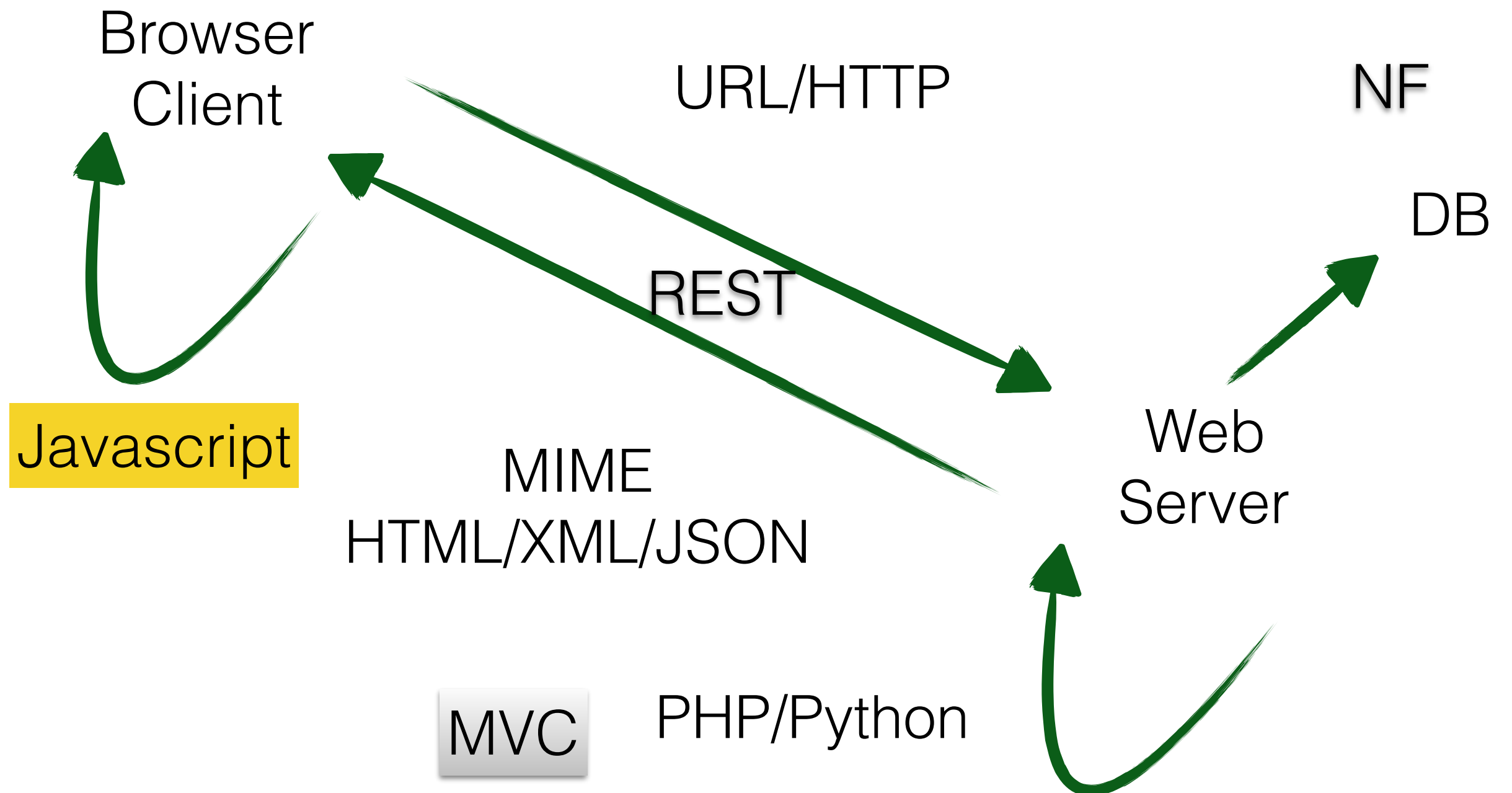
# Web dynamic

## Dynamic Client:

### DOM, Javascript, CSS

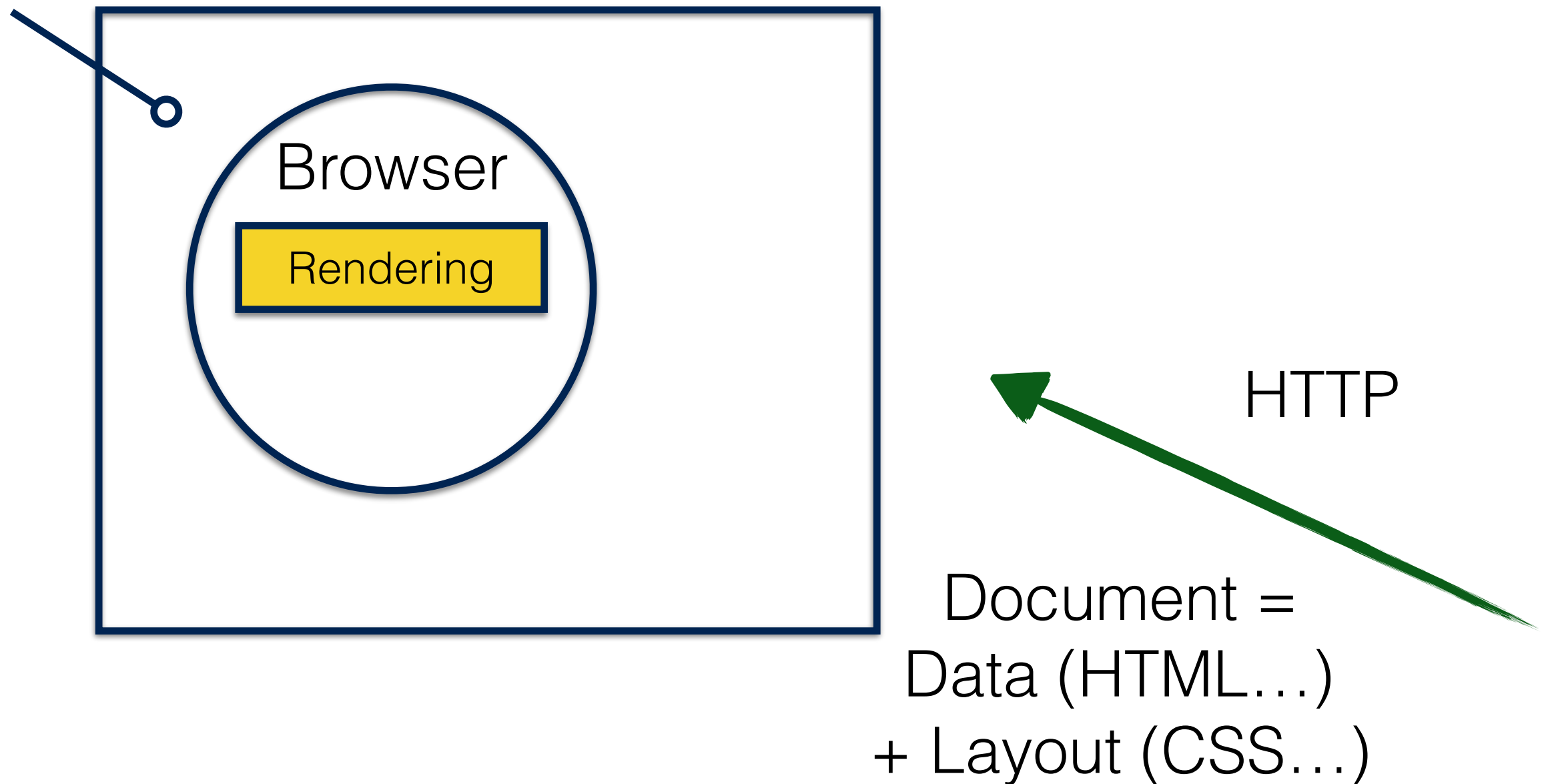
Frédéric Le Mouël

# The Big Picture



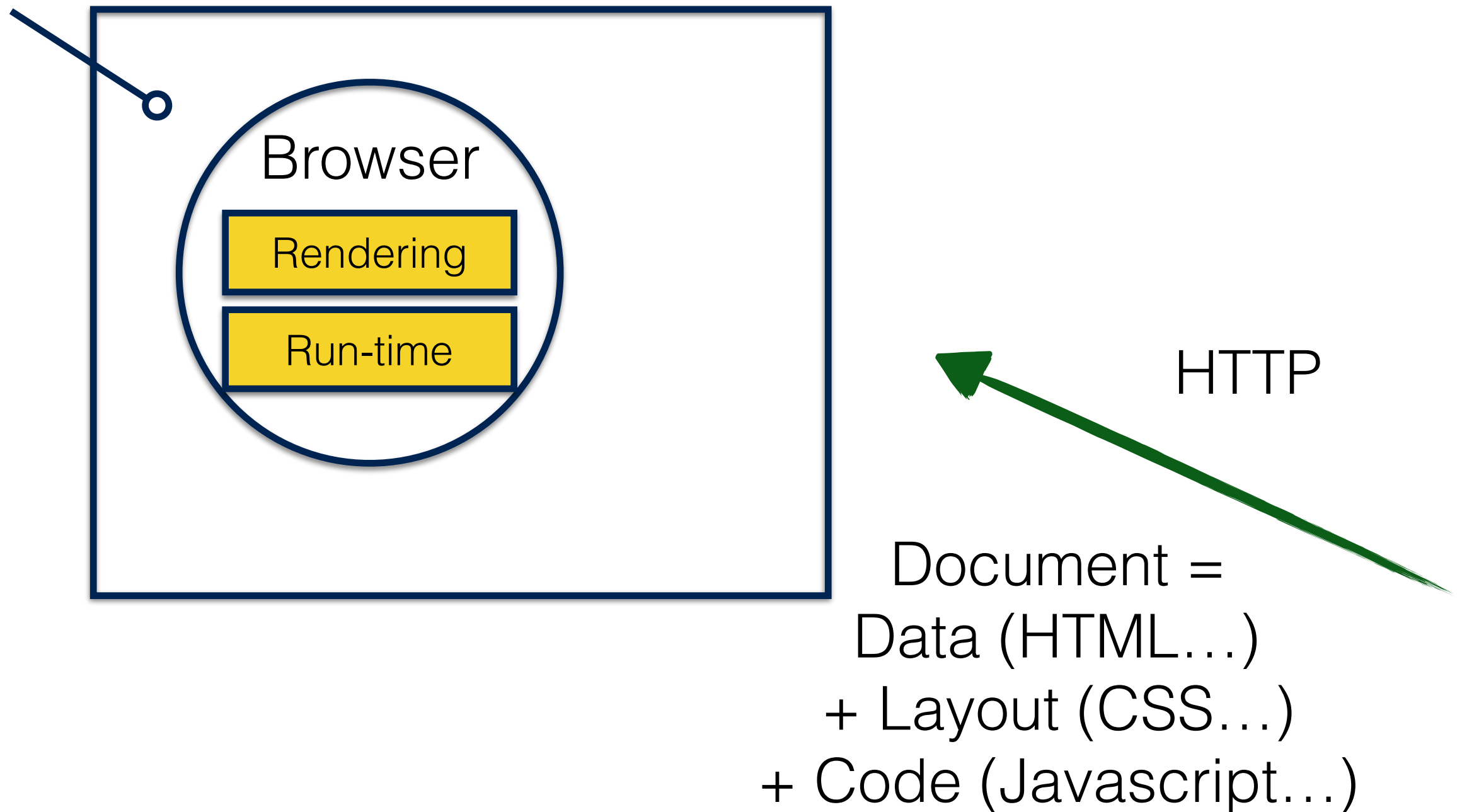
# Zoom Client

Hardware  
Machine



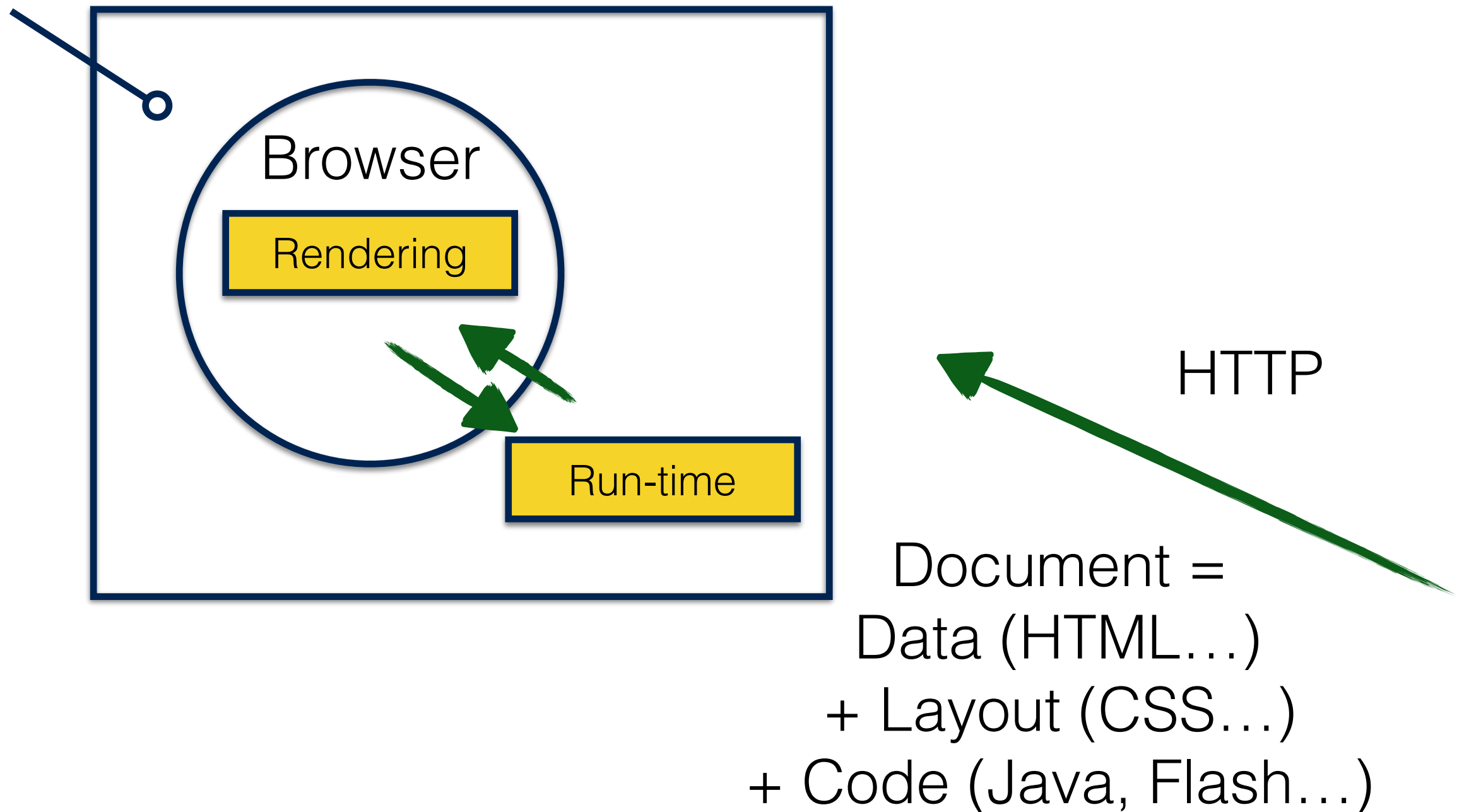
# Zoom Client

Hardware  
Machine

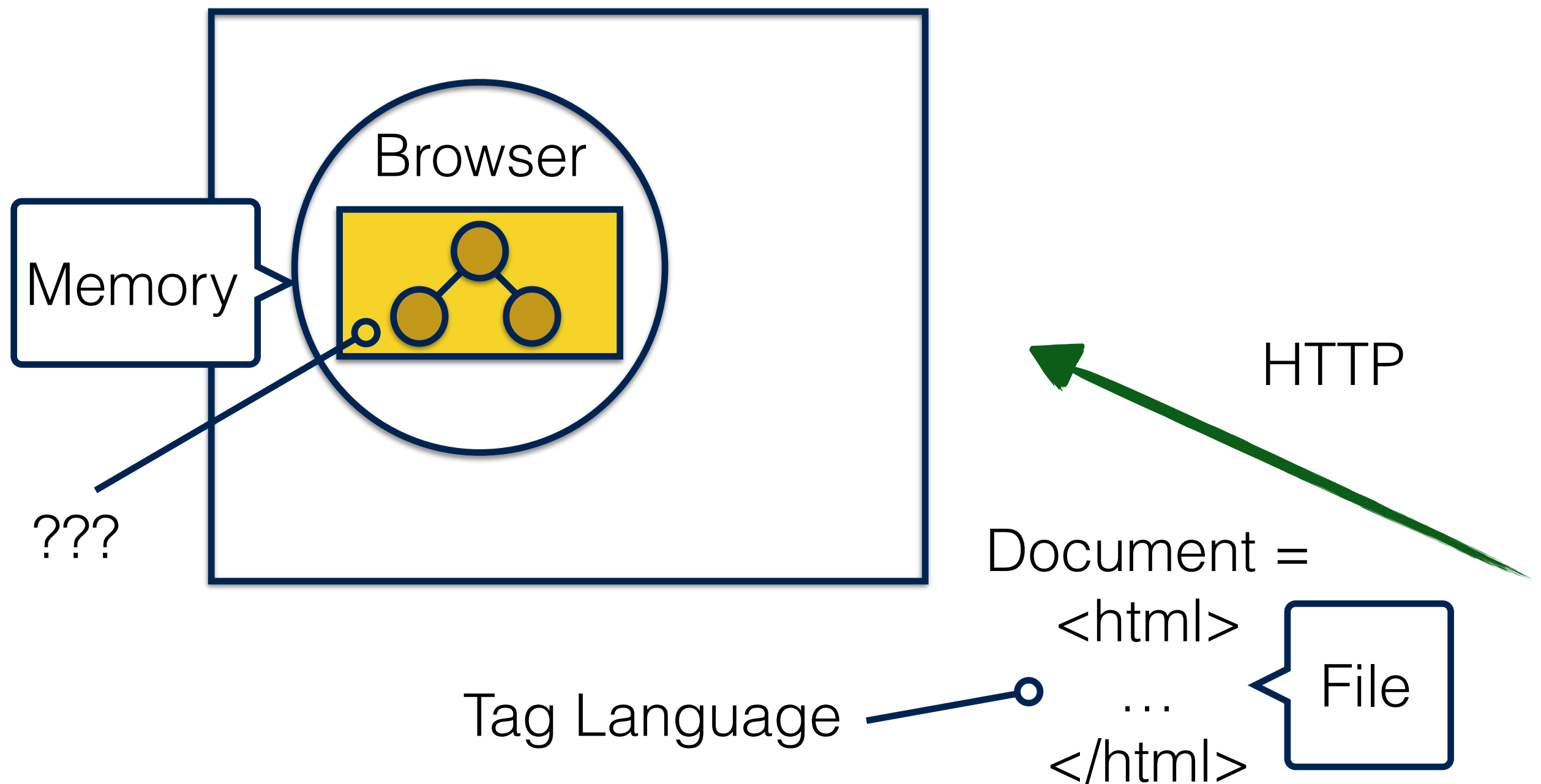


# Zoom Client

Hardware  
Machine

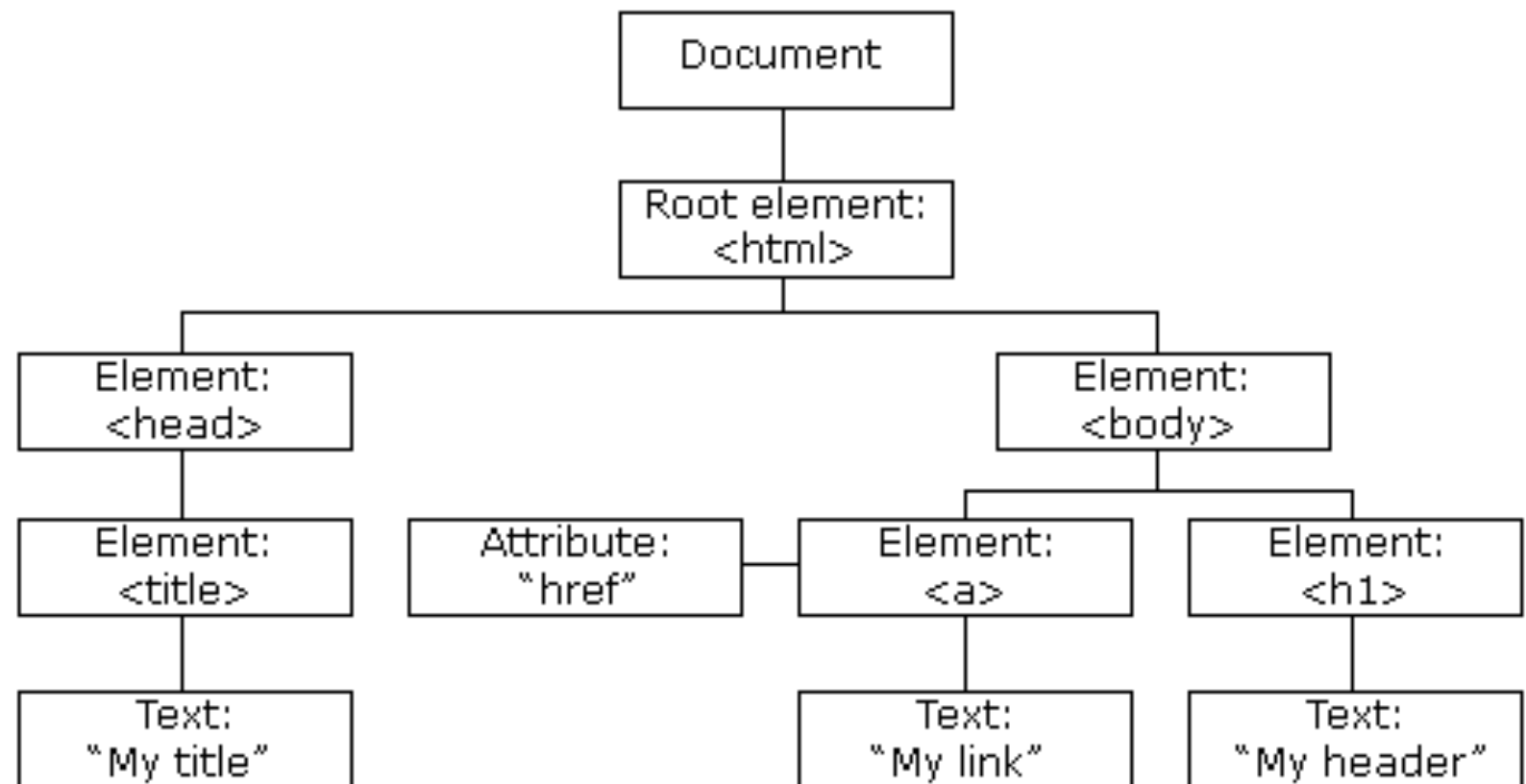


# Rendering/Manipulating



# Document Object Model (DOM)

- W3C Standard: DOM 4 working draft 2014
- Document as an Object Tree
- Manipulation by the Javascript Programming Language



# Document Object Model (DOM)

- DOM standard is composed of:
  - Core DOM: model for all document type
  - XML DOM: model for XML documents
  - HTML DOM: model for HTML documents



# HTML DOM

- HTML tags mapped to **Objects**
- Javascript API to manipulate:
  - **properties** of Objects, called ***values***
  - **methods** to access the Object tree / all Objects, called ***actions***
  - **events** to monitor Objects change

# HTML DOM

object -  
nodeId definition

<html>  
<body>

<p id="demo"></p>

<script>

document.getElementById("demo").innerHTML = "Hello World!";  
</script>

</body>  
</html>

object property -  
defining the content of an object

access method -  
search through the object tree by nodeId

# HTML DOM - API

## To find objects

```
document.getElementById(id)  
document.getElementsByTagName(name)  
document.getElementsByClassName(name)  
document.querySelectorAll(name);
```

## To add/remove objects

```
document.createElement(element)  
document.removeChild(element)  
document.appendChild(element)  
document.replaceChild(element)  
document.write(text)
```

# HTML DOM - API

To navigate through the object tree

```
element.parentNode  
element.childNodes[nodenum]  
element.firstChild  
element.lastChild  
element.nextSibling  
element.previousSibling
```

# HTML DOM - API

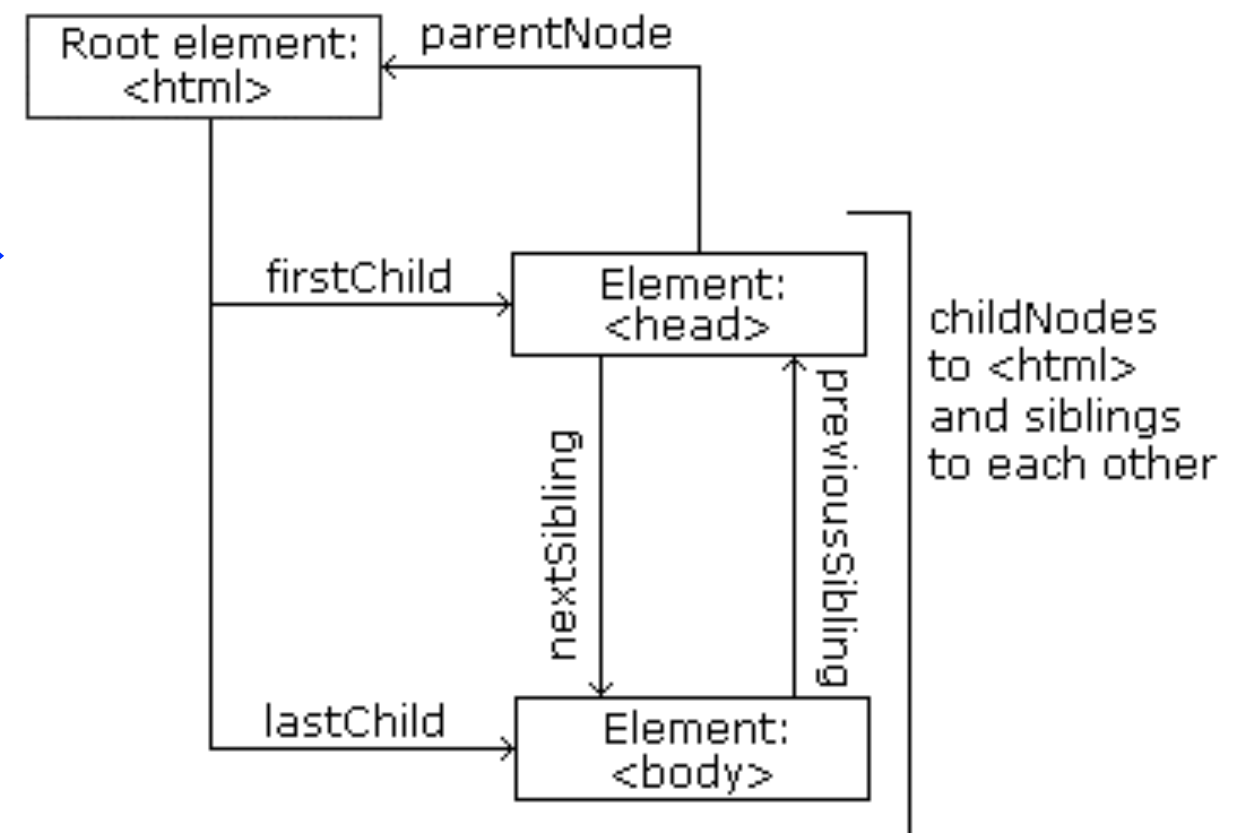
To manipulate the object tree

```
<html>

  <head>
    <title>DOM Tutorial</title>
  </head>

  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>

</html>
```



# HTML DOM - API

To change objects

```
element.innerHTML = new html content  
element.attribute = new value  
element.setAttribute(attribute, value)  
element.style.property = new style
```

# HTML DOM - API

To monitor objects

```
document.getElementById(id).onclick = function(){code}
```

event

handler

```
element.addEventListener(event, function, useCapture);
```

event bubbling/capturing

# HTML DOM - API

To monitor objects

```
document.getElementById(id).onclick = function(){code}
```

event

**Mouse**

onclick  
oncontextmenu  
ondblclick  
onmousedown  
...

**Keyboard**

...

**Forms**

...

```
element.addEventListener(event, function, useCapture);
```



# HTML DOM - API

To monitor objects

```
element.addEventListener("click",  
                           function(){alert("Hello World!");});
```

or


```
element.addEventListener("click", myFunction);
```

```
function myFunction() {  
    alert ("Hello World!");  
}
```

# HTML DOM - API

To monitor objects


several handlers  
for the same element



```
element.addEventListener("click", myFunction);  
element.addEventListener("click", mySecondFunction);
```

and

several events  
for the same element

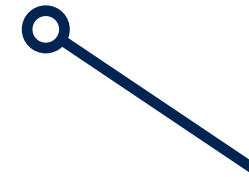


```
element.addEventListener("mouseover", myFunction);  
element.addEventListener("click", mySecondFunction);  
element.addEventListener("mouseout", myThirdFunction);
```

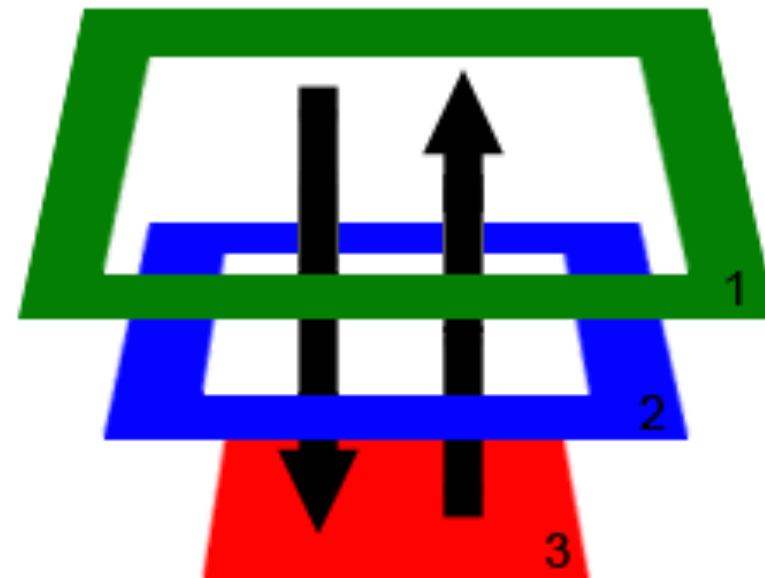
# HTML DOM - API

To monitor objects

```
element.addEventListener(event, function, useCapture);
```



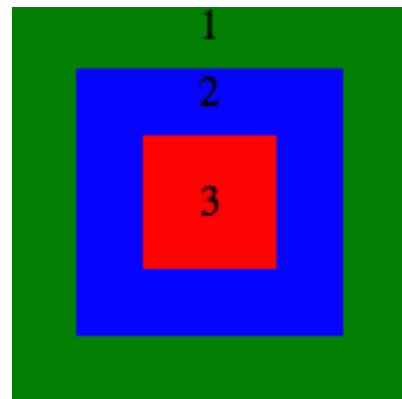
event bubbling/capturing?



# HTML DOM - API

To monitor objects

```
<div id="demo1">  
  <p id="demo2"></p>  
</div>
```



```
<script>  
  document.getElementById("demo1")  
    .addEventListener("mouseover", myFunction1);  
  document.getElementById("demo2")  
    .addEventListener("mouseover", myFunction2);  
</script>
```

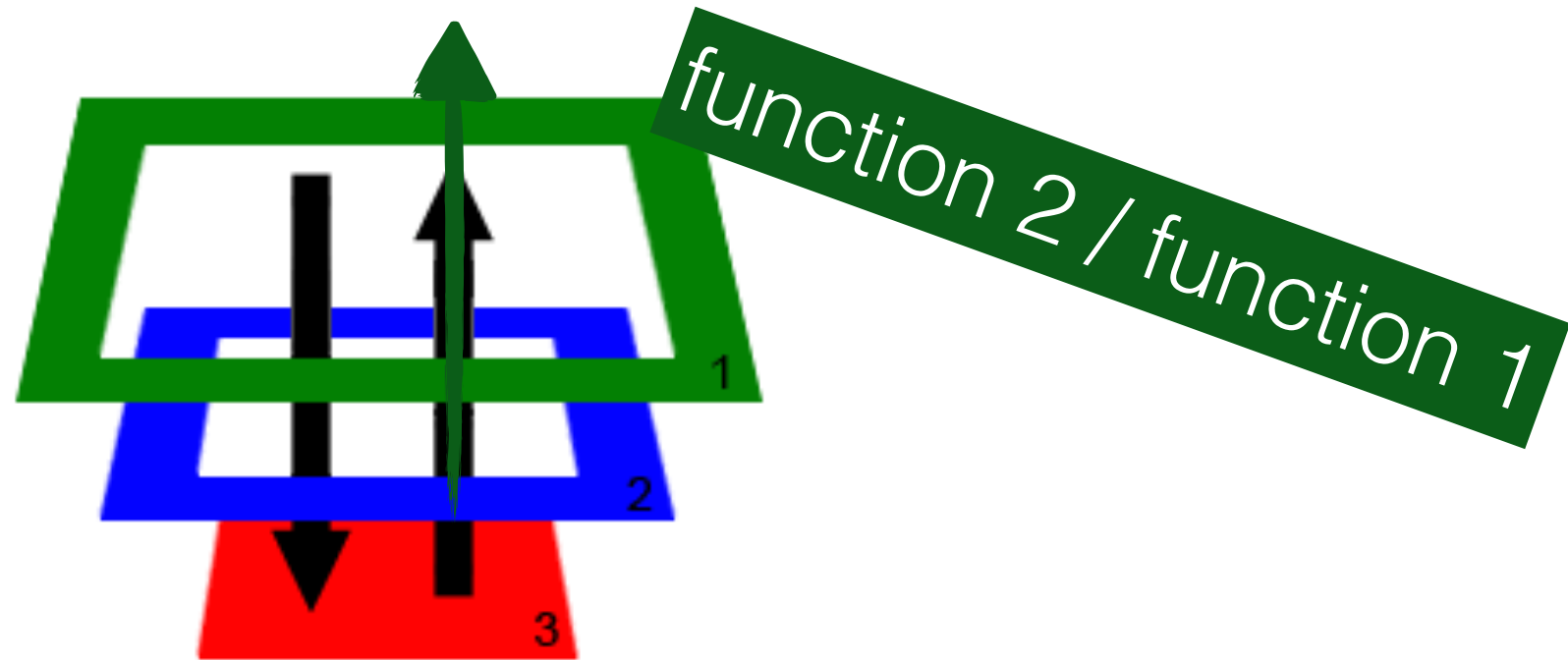
function 1 / function 2 ?  
function 2 / function 1 ?

# HTML DOM - API

To monitor objects

```
<div id="demo1">  
  <p id="demo2"></p>  
</div>
```

```
<script>  
  document.getElementById("demo1")  
    .addEventListener("mouseover", myFunction1, false);  
  document.getElementById("demo2")  
    .addEventListener("mouseover", myFunction2, false);  
</script>
```



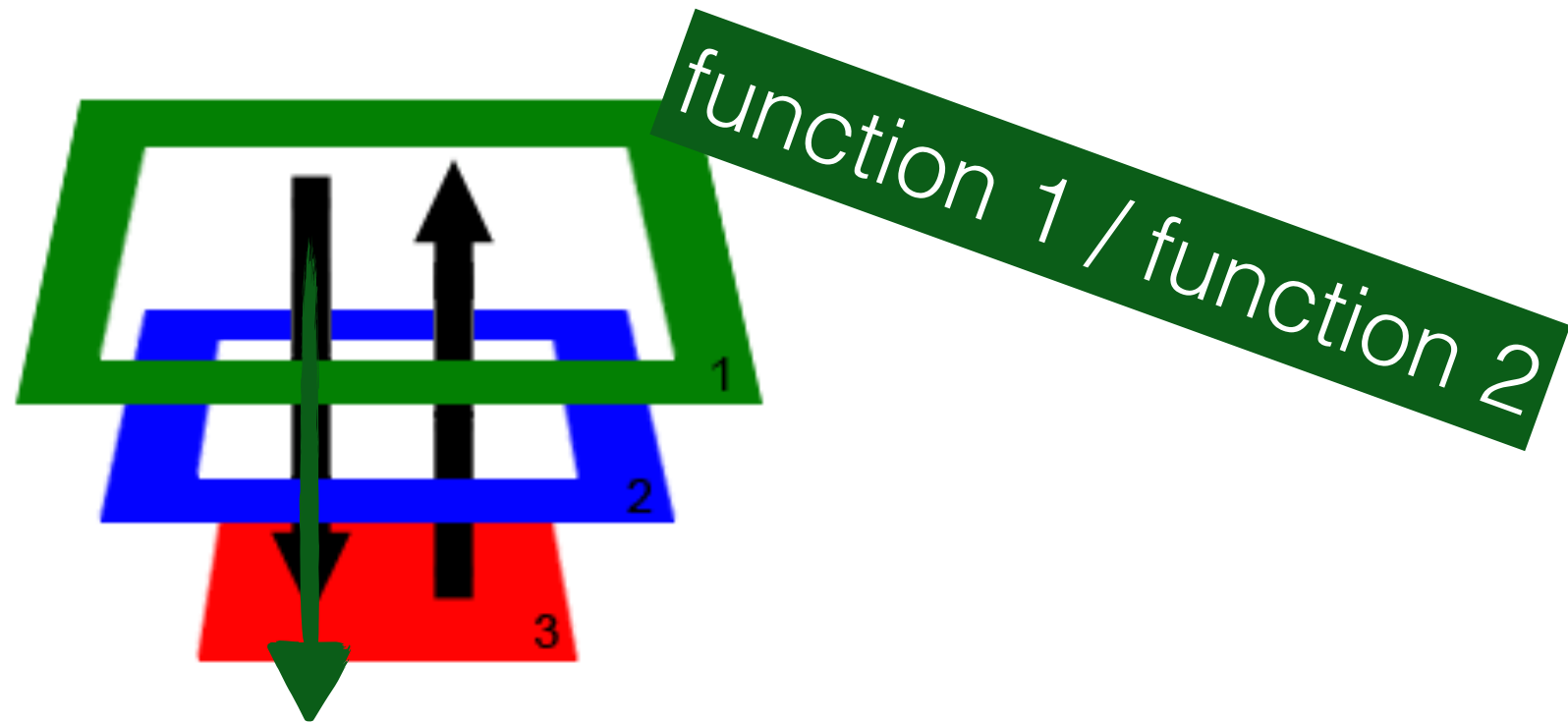
bubbling: default behavior

# HTML DOM - API

To monitor objects

```
<div id="demo1">  
  <p id="demo2"></p>  
</div>
```

```
<script>  
  document.getElementById("demo1")  
    .addEventListener("mouseover", myFunction1, true);  
  document.getElementById("demo2")  
    .addEventListener("mouseover", myFunction2, true);  
</script>
```



capturing!

# HTML DOM - API

To change objects

*element.style.property = new style*



style?

# HTML DOM - API

```
<html>
```

```
<body>
```

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Hello World!";
```

```
document.getElementById("demo").style.color = "red";
```

```
</script>
```

```
</body>
```

```
</html>
```

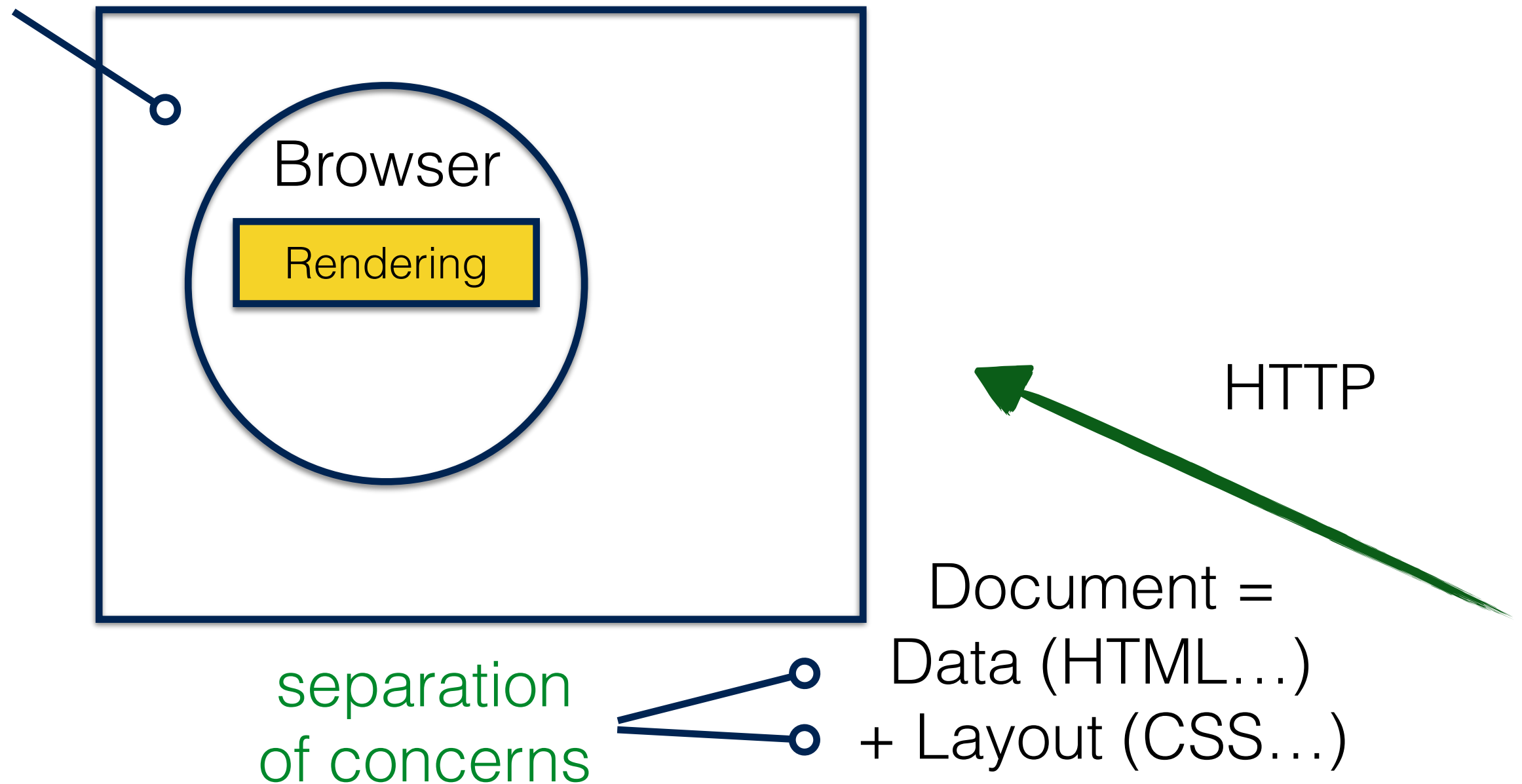


a color?



# Zoom Client

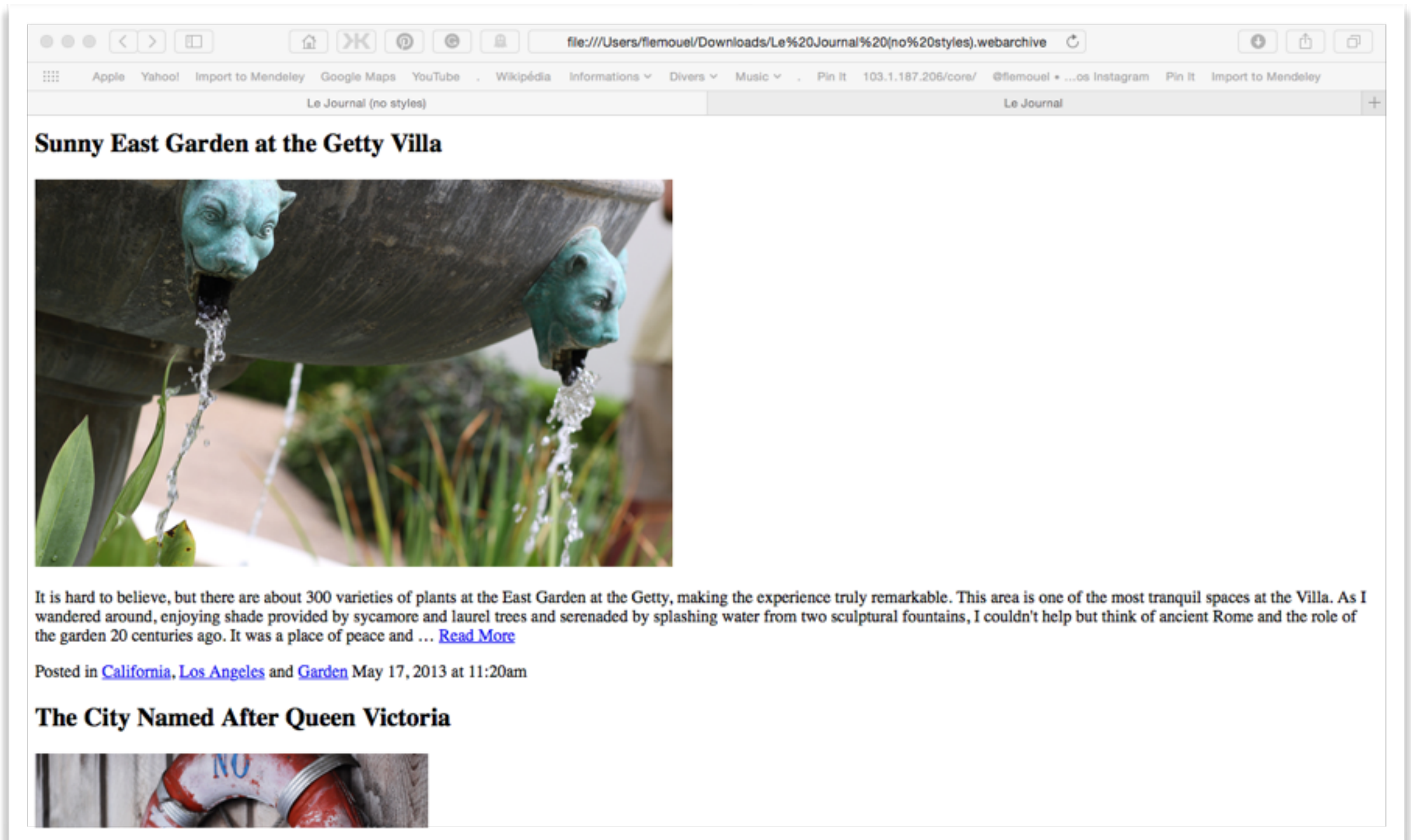
Hardware  
Machine



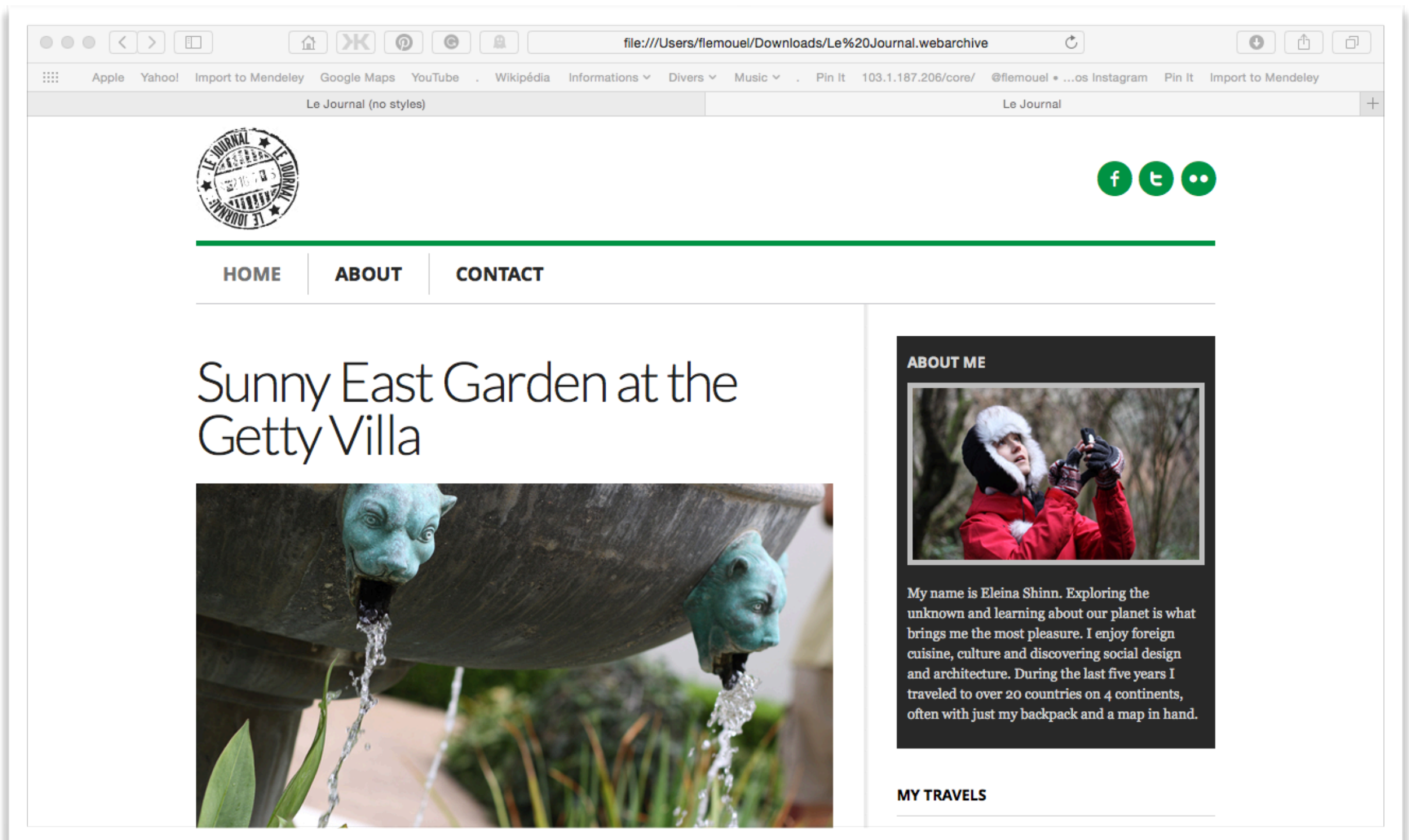
# Cascading Style Sheets (CSS)

- CSS describes the data **layout**
- i.e. the way HTML elements will be displayed on a **media**, e.g. a large screen (laptop), a smaller screen (mobile), a printer, etc.

# CSS - Example - without



# CSS - Example - with



# From HTML 3

```
<html>  
<body>
```

```
<style type="text/css">  
  p.first {color:blue}  
  p.second {color:green}  
</style>
```

style in the  
header or the body

Caution! Oldies!

```
<p class="first">Hello World</p>  
<p class="second">Hello World</p>
```

static style  
in the element

```
<h1 style="color:blue;text-align:center">This is a title</h1>
```

```
<font size="3" color="red">This is some text!</font>
```

```
</body>  
</html>
```

style tags

# From HTML 3 to CSS

```
<html>  
<body>
```

```
<style>  
p.fi  
p.s  
</st
```

```
<p>  
<p>
```

```
<h1 style="color: red;">The element title</h1>
```

```
<font size="3" color="red">
```

```
</body>  
</html>
```

style tags

HTML 5!  
Some elements  
still compatible!  
But globally deprecated!



# From HTML 3

```
<html>  
<body>
```

```
<style type="text/css">  
  p.first {color:blue}  
  p.second {color:green}  
</style>
```

```
<p class="first">Hello World</p>  
<p class="second">Hello World</p>
```

```
<h1 style="color:blue;text-align:center">This is a
```

```
<font size="3" color="red">This is some text!</font>
```

```
</body>  
</html>
```

internal style:  
not recommended!  
style in the  
header or the body

inline style:  
to modify  
dynamically  
in javascript

deprecated  
style tag

# HTML / CSS

Separation  
of concerns!

link between  
data and layout

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="theme.css">
```

```
</head>
```

```
<body>
```

```
...
```

```
</body>
```

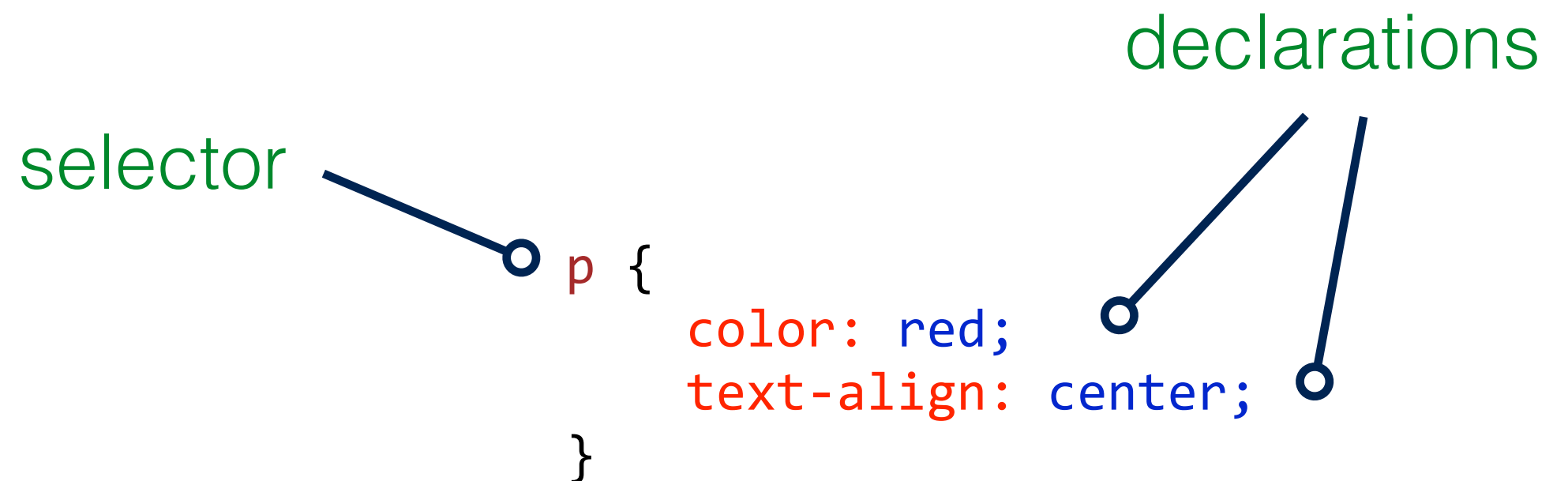
```
</html>
```

external file



# CSS Structure

```
selector {  
  declaration;  
  declaration;  
}
```



# CSS Example

HTML element

body {  
}

h1 {  
}

p {  
}

background-color: #d0e4fe;

color: orange;  
text-align: center;

font-family: "Times New Roman";  
font-size: 20px;

values

properties

plenty  
of design  
properties!

# CSS Example

```
<html>
```

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="theme.css">
</head>
```

```
<body>
```

```
<p id="demo1">Hello World!</p>
```

```
</body>
```

```
</html>
```

id selector

```
#demo1 {
  color: orange;
  text-align: center;
}
```

# CSS Example

```
<html>
```

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="theme.css">
</head>
```

```
<body>
```

```
<h1 class="center">Hello World1!</h1>
```

```
<p class="notcenter">Hello World2!</p>
```

```
<p class="center">Hello World3!</p>
```

```
</body>
```

```
</html>
```

class selector

```
.center {
  color: orange;
  text-align: center;
}
```

# CSS Example

```
<html>
```

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="theme.css">
</head>
```

```
<body>
```

```
<h1 class="center">Hello World1!</h1>
```

```
<p class="notcenter">Hello World2!</p>
```

```
<p class="center">Hello World3!</p>
```

```
</body>
```

```
</html>
```

element &  
class selector

```
p.center {
  color: orange;
  text-align: center;
}
```

# CSS Example

```
<html>
```

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="theme.css">
</head>
```

```
<body>
```

```
<h1 class="center">Hello World1!</h1>
```

```
<p class="notcenter">Hello World2!</p>
```

```
<p class="center big">Hello World3!</p>
```

```
</body>
```

```
</html>
```

element &  
multi-class selector

```
p.center {
  color: orange;
  text-align: center;
}

.big {
  font-size: 250%;
}
```

# CSS 3 - Responsive



desktop  
/ laptop

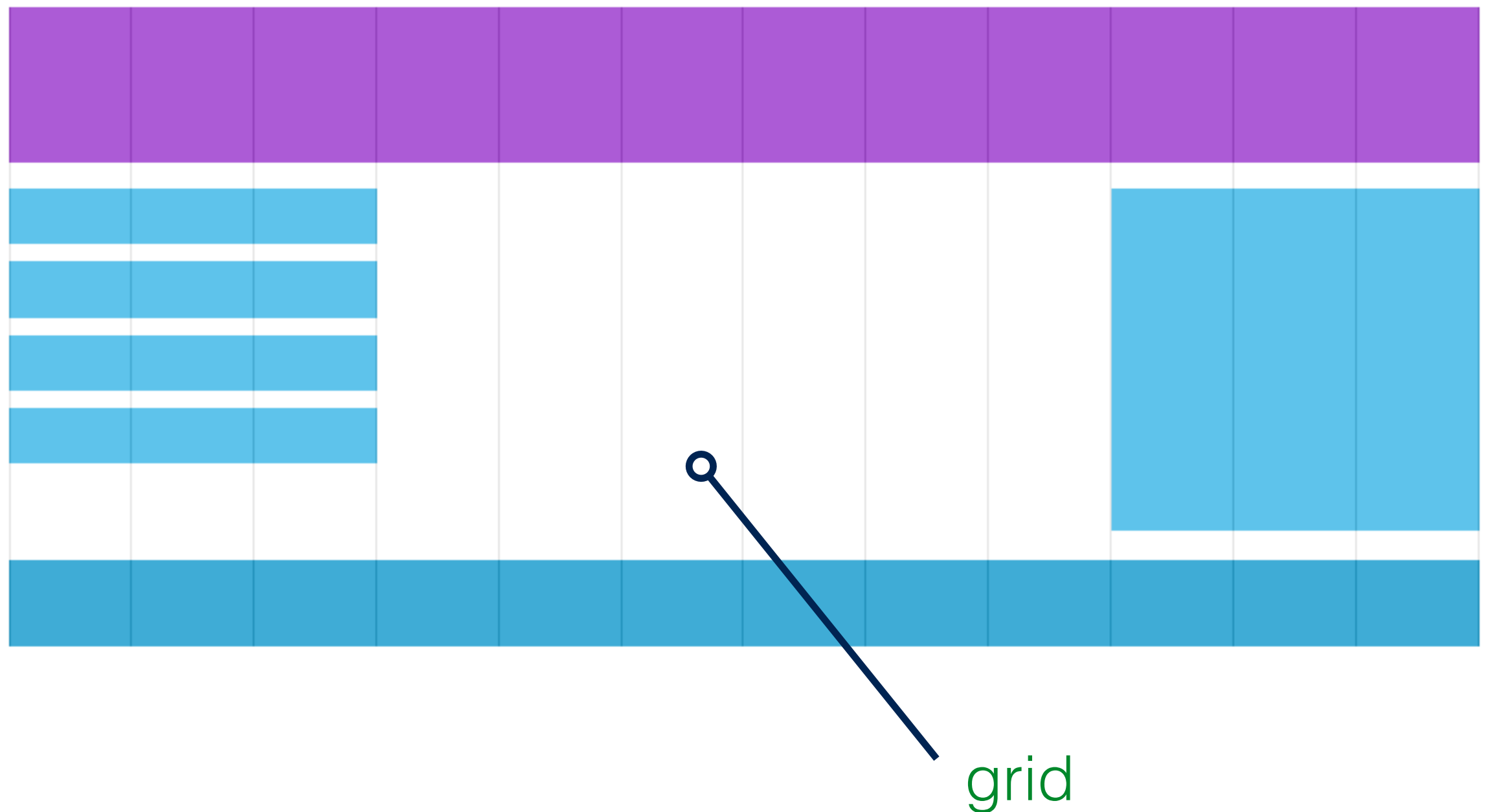


tablet



smartphone

# CSS - Responsive





# CSS - Responsive

```
/* For desktop: */  
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

```
@media only screen and (max-width: 768px) {  
    /* For mobile phones: */  
    [class*="col-"] {  
        width: 100%;  
    }  
}
```

# Webography

- W3C Schools
  - <http://www.w3schools.com>
- HTML and CSS - Elisabeth Castro - Eighth Edition
  - <http://htmlcssvqs.com/8ed/examples/>