

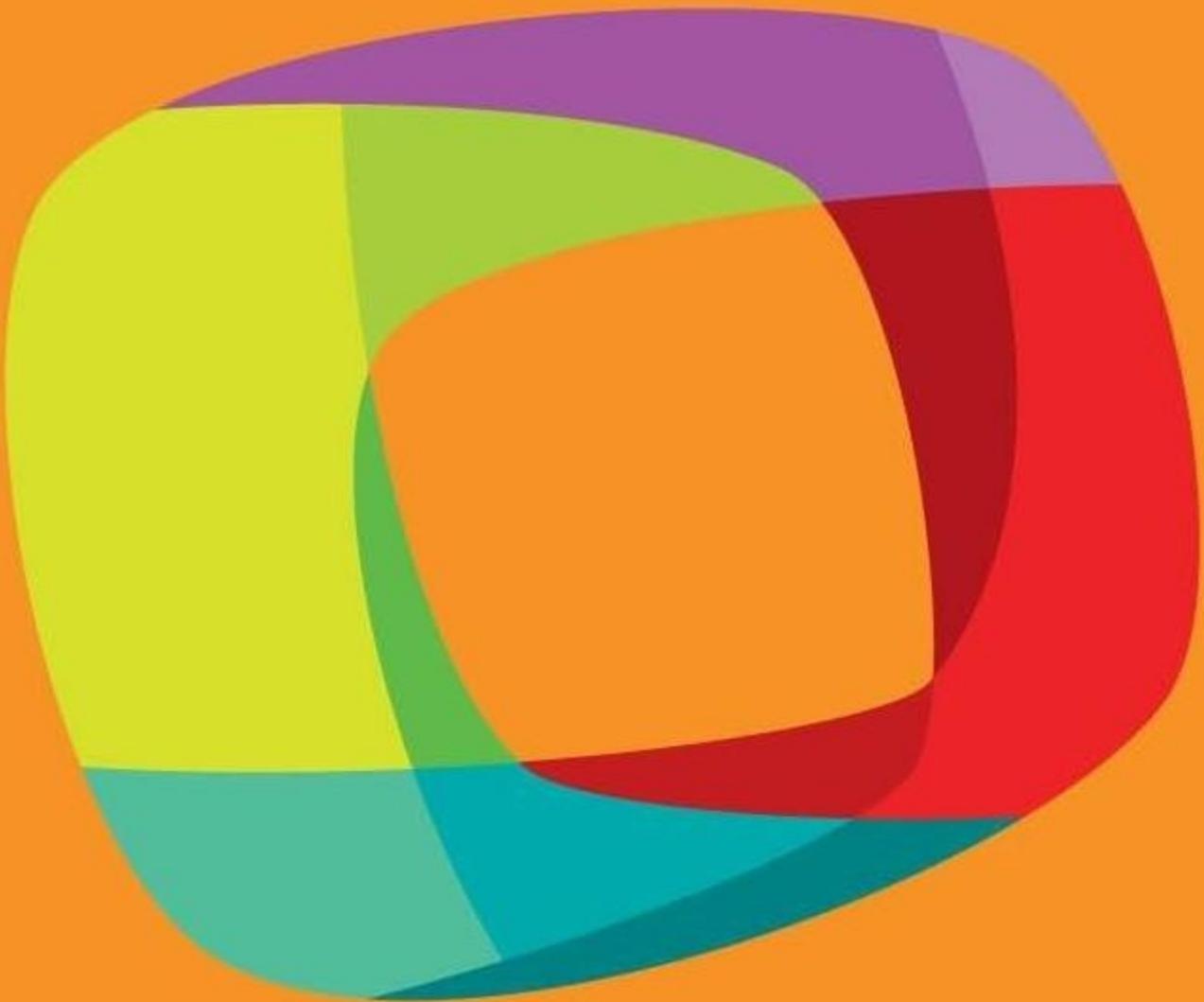
Repaint/Reflow

Renderizando uma página web

Robson Júnior

JS







<https://github.com/JSRocksHQ>

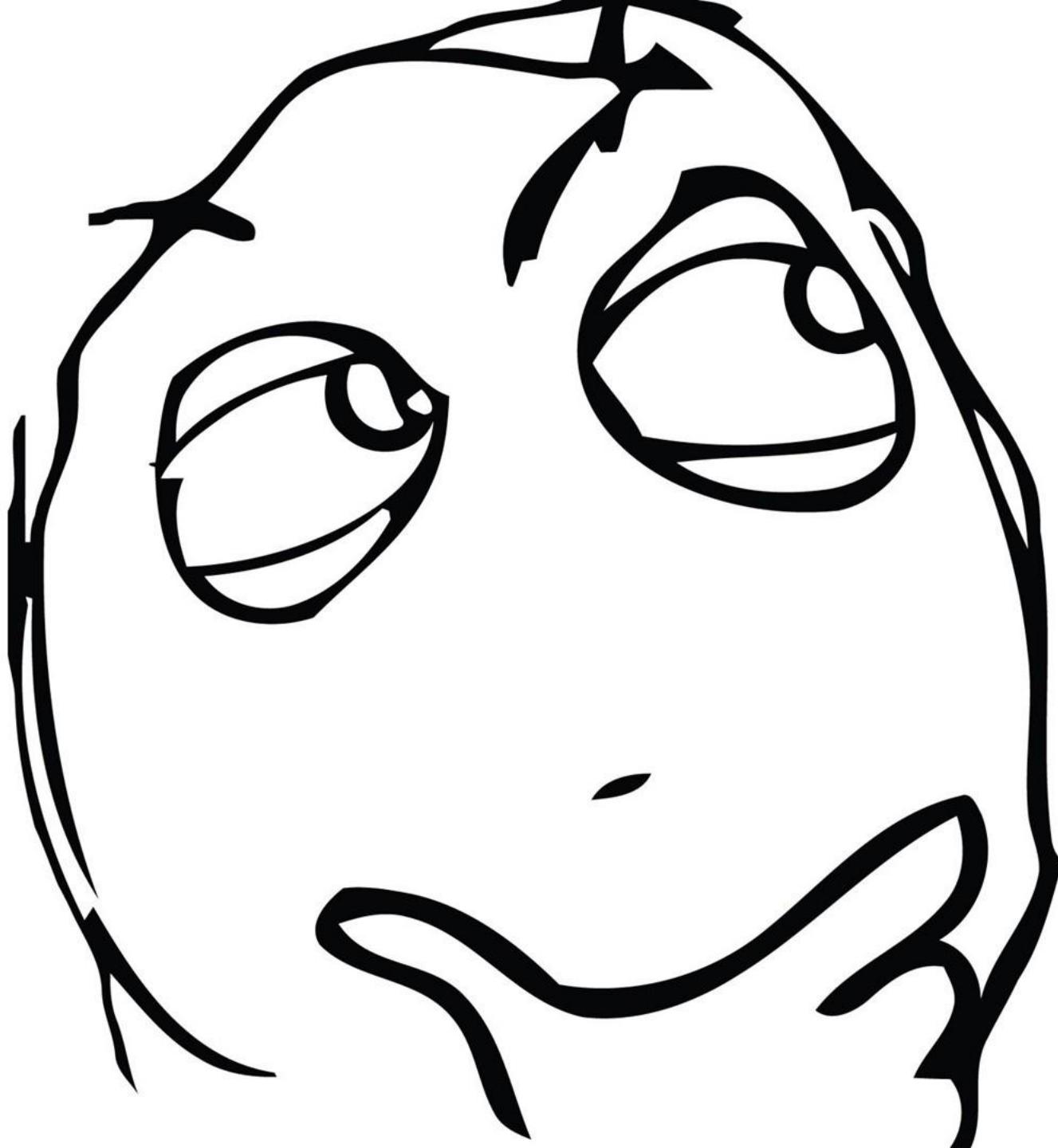


<https://github.com/Webschool-io>

A gold shield-shaped badge with a decorative border of stars. The word "HTML" is written in white, bold, sans-serif capital letters in the center of the badge.

HTML

HyperText Markup Language

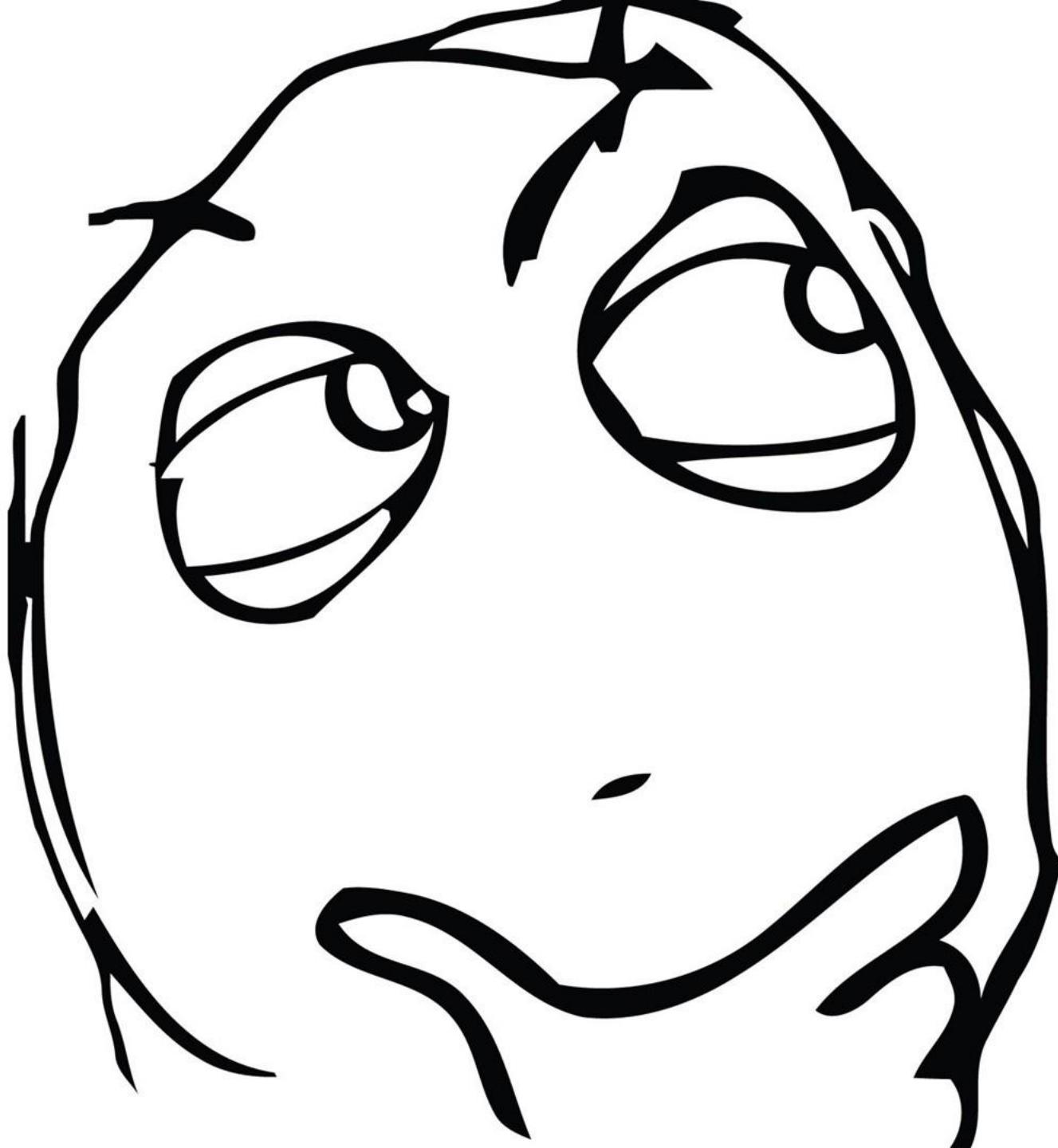


[code]

CSS

font-family:
serif
font-size:
1em
font-weight:
normal
font-style:
italic

Cascading Style Sheets



[code]

**YOU ARE
THE CSS
TO MY
HTML**

DOM / CSSOM



FOCA NO TRABALHO!



DOM

DOM

Document Object Model

- Conversão

- Conversão
- Criação de tokens

- Conversão
- Criação de tokens
- Nodes

- Conversão
- Criação de tokens
- Nodes
- Criação do DOM

Bytes

3C 62 6F 64 79 3E 48 65 6C 6C 6F 2C 20 3C 73 70 61 6E 3E 77 6F 72 6C 64 21 3C 2F 73 70 61
6E 3E 3C 2F 62 6F 64 79 3E

Characters

<html><head>...</head><body><p>Hello web performance...</p>

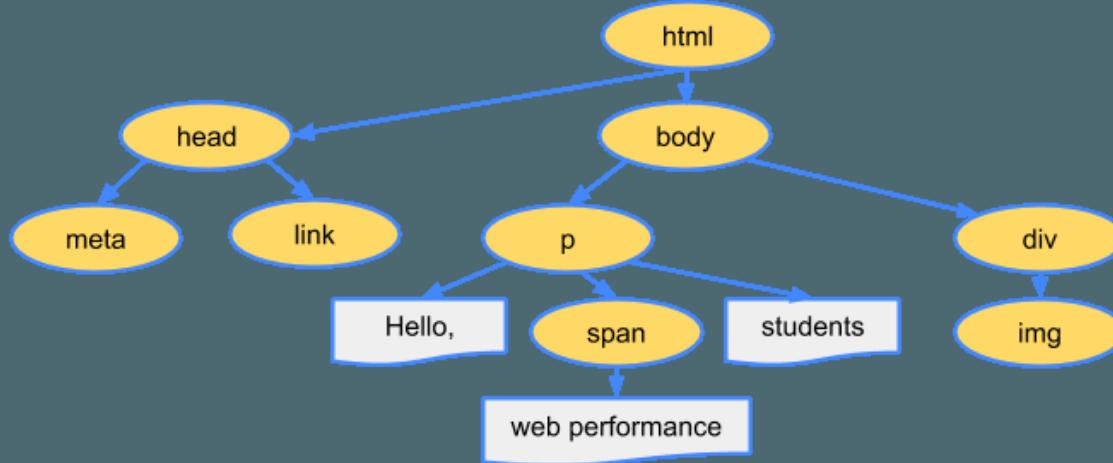
Tokens

StartTag: html StartTag: head ... EndTag: head StartTag: body StartTag: p Hello ...

Nodes

html head meta body p Hello

DOM



O que o
DOM não é ??

- Documento de especificação binária

- Documento de especificação binária
- Setar tipo dos dados

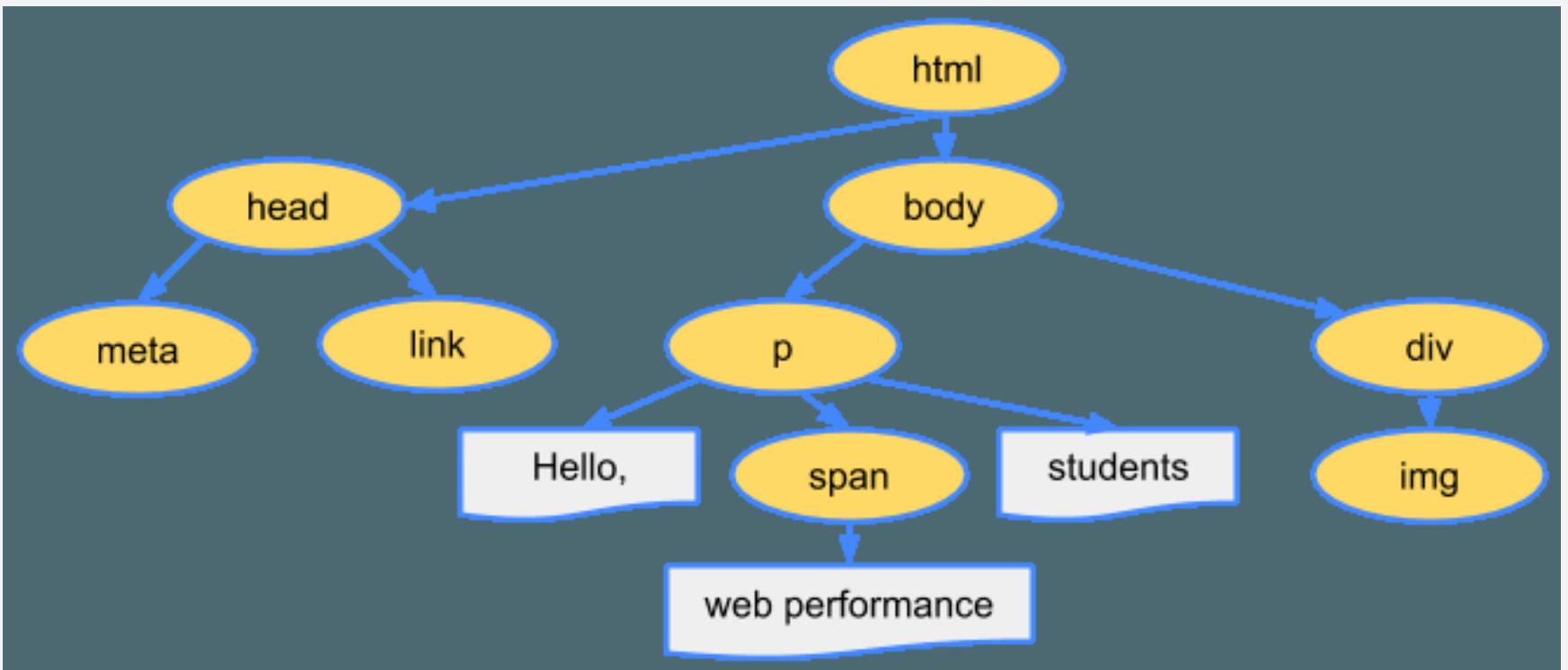
- Documento de especificação binária
- Setar tipo dos dados
- Definir relevância e como conteúdo será estruturado

O que o DOM
realmente é ??

- **Informação Cross Plataform**

- Informação Cross Platform
- Especifica interface

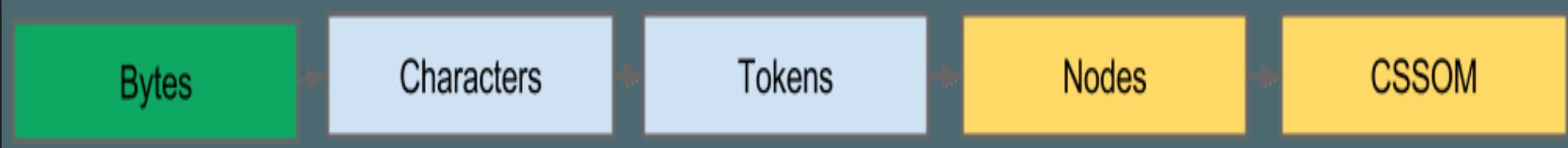
- Informação Cross Platform
- Especifica interface
- API para este conjunto de informações.

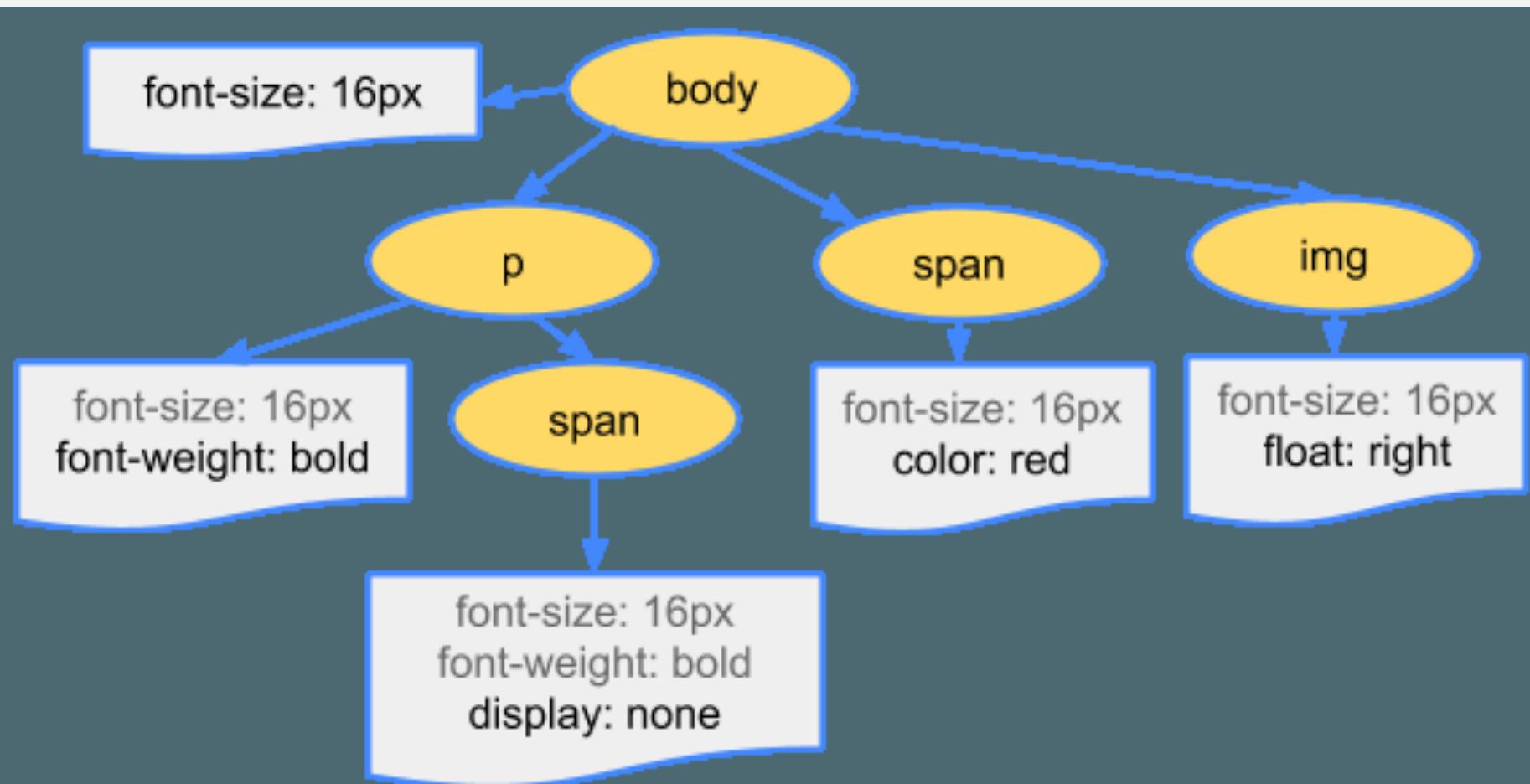


CSSOM

CSSOM

CSS Object Model





- **Bytes → caracteres → tokens → nós → modelo de objeto.**

- **Bytes → caracteres → tokens → nós → modelo de objeto.**
- **Marcação HTML é transformada em DOM**

- **Bytes → caracteres → tokens → nós → modelo de objeto.**
- **Marcação HTML é transformada em DOM**
- **Marcação CSS é transformada em CSSOM**

- **Bytes → caracteres → tokens → nós → modelo de objeto.**
- **Marcação HTML é transformada em DOM**
- **Marcação CSS é transformada em CSSOM**
- **DOM e CSSOM são estruturas de dados independentes.**

Render Tree

- **Analisa todos nodes**

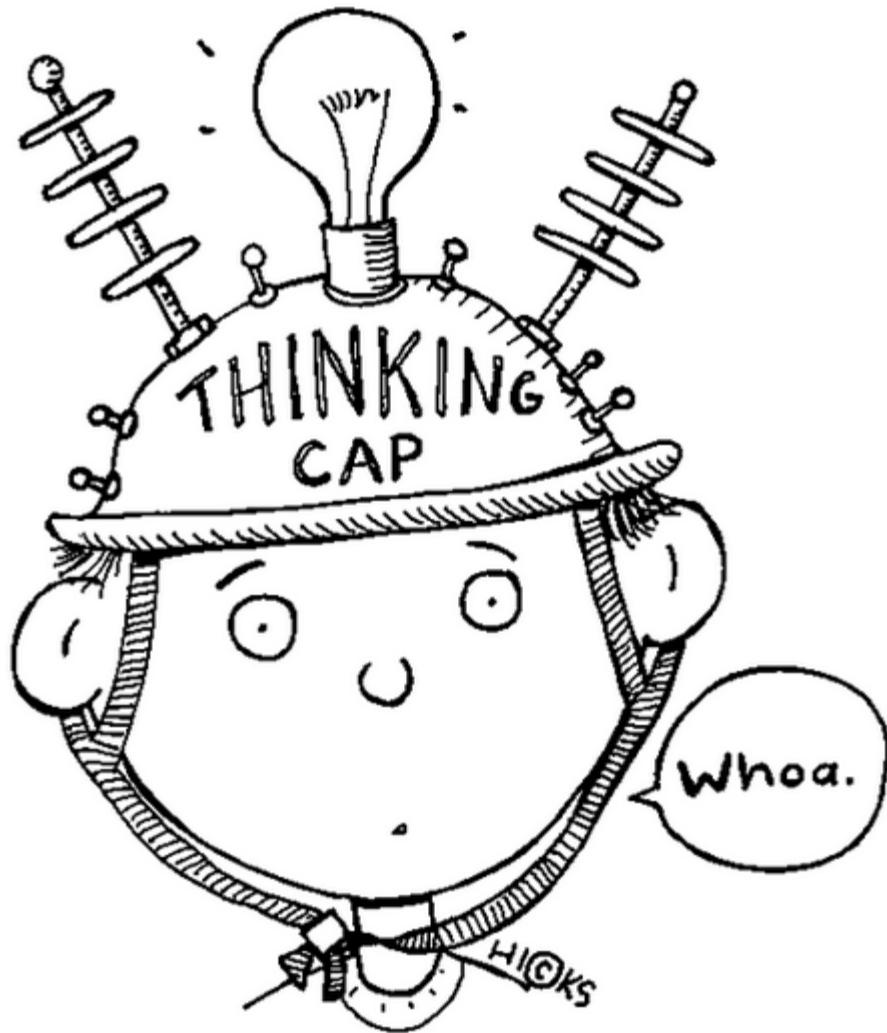
- **Analisa todos nodes**
- **Encontra no CSSOM a regra de cada node**

- **Analisa todos nodes**
- **Encontra no CSSOM a regra de cada node**
- **Devolve os nodes com seus estilos computados**

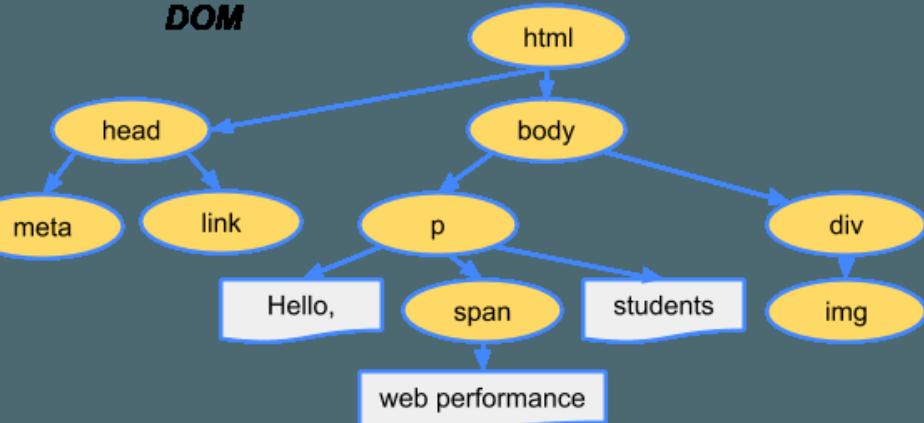
display: none

!==

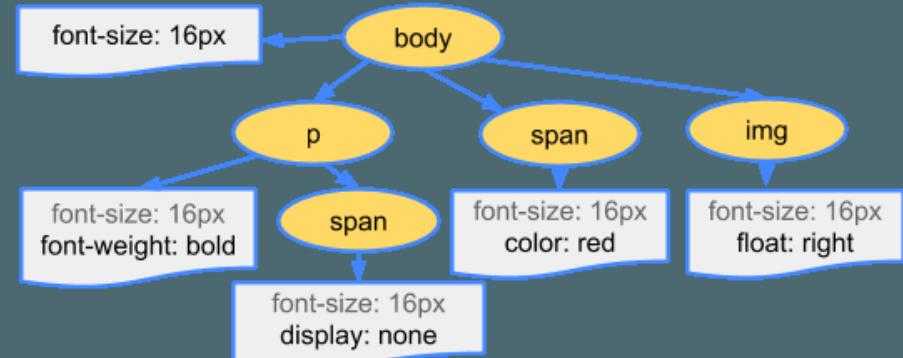
visibility: hidden



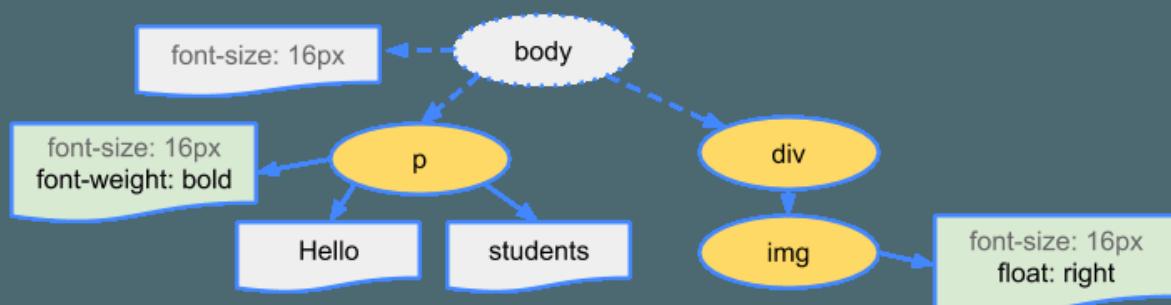
DOM



CSSOM



Render Tree



display: none;

<http://jsfiddle.net/robsongajunior/23wu66zf/2/>

visibility: hidden;

<http://jsfiddle.net/robsongajunior/jr526u8k/3/>

* → **Absoluto**

*viewport
size=device-width*

A diagram illustrating a responsive web layout. A large horizontal red double-headed arrow spans the width of the page, labeled "viewport size=device-width". Inside this area, there is a blue rectangular container divided into two equal-width sections. Each section is labeled "div (50%)". The left section contains the text "Hello world".

*viewport
size=device-width*

div (50%)

Hello world

div (50%)

[video]

- **DOM + CSSOM === Render Tree**

- **DOM + CSSOM === Render Tree**
- **Render Tree === Nodes para renderização**

- **DOM + CSSOM === Render Tree**
- **Render Tree === Nodes para renderização**
- **LAYOUT === posição + tamanho exato**

- **DOM + CSSOM === Render Tree**
- **Render Tree === Nodes para renderização**
- **LAYOUT === posição + tamanho exato**
- **PAINT === Render Tree renderizado**

REPAINT

<http://jsfiddle.net/robsongajunior/qugvk4bx/1/>

REFLOW

REFLOW

sempre tem

REPAINT

<http://jsfiddle.net/gpqlstv/1/>

Test runner

Done. Ready to run again.

Run again

Testing in Firefox 40.0 on Ubuntu 64-bit		
	Test	Ops/sec
DOM manipulation every iteration	<pre>var myList = document.getElementById("myList"); for (var i = 0; i < 100; i++) { myList.innerHTML += "" + i + ""; }</pre>	2.84 ±56.22% 100% slower
DOM manipulation after all iterations	<pre>var myList = ""; for (var i = 0; i < 100; i++) { myList += "" + i + ""; } document.getElementById("myList").innerHTML = myList;</pre>	3,342 ±4.49% 98% slower
DOM manipulation after caching innerHTML	<pre>var myListHTML = document.getElementById("myList").innerHTML; for (var i = 0; i < 100; i++) { myListHTML += "" + i + ""; }</pre>	144,980 ±4.26% fastest

<http://jsperf.com/browser-diet-dom-manipulation/11>

A gold shield-shaped badge with a decorative border of stars. The word "HTML" is written in white, bold, sans-serif capital letters in the center of the badge.

HTML

CSS

font-family:
serif
font-size:
1em
font-weight:
normal
font-style:
italic

DOM

**DOM
CSSOM**

**DOM
CSSOM
Render Tree**









Referências

- <http://www.w3.org/TR/DOM-Level-2-Core/introduction.html>
- <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/constructing-the-object-model?hl=en>
- <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/constructing-the-object-model?hl=pt-br#css-object-model-cssom>
- <http://dev.w3.org/csswg/cssom/>
- <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/render-tree-construction>
- <http://www.phpied.com/rendering-repaint-reflowrelayout-restyle/>
- <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/measure-crp?hl=en>
- <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/analyzing-crp?hl=en>
- <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/optimizing-critical-rendering-path?hl=en>
- <http://browserdiet.com/pt/>
- <https://www.youtube.com/watch?v=jw4tVn7CRcl>
- https://www.youtube.com/watch?v=a2_6bGNZ7bA
- https://developer.mozilla.org/pt-BR/docs/Tools/Paint_Flashing_Tool
- <https://www.youtube.com/watch?v=gZH1d2Co1X0>
- <https://www.youtube.com/watch?v=aH9eVa2cTcM>

OBRIGADO

@robsongajunior