

resilience

HALFSTACK LONDON 2024

Hello, my name is

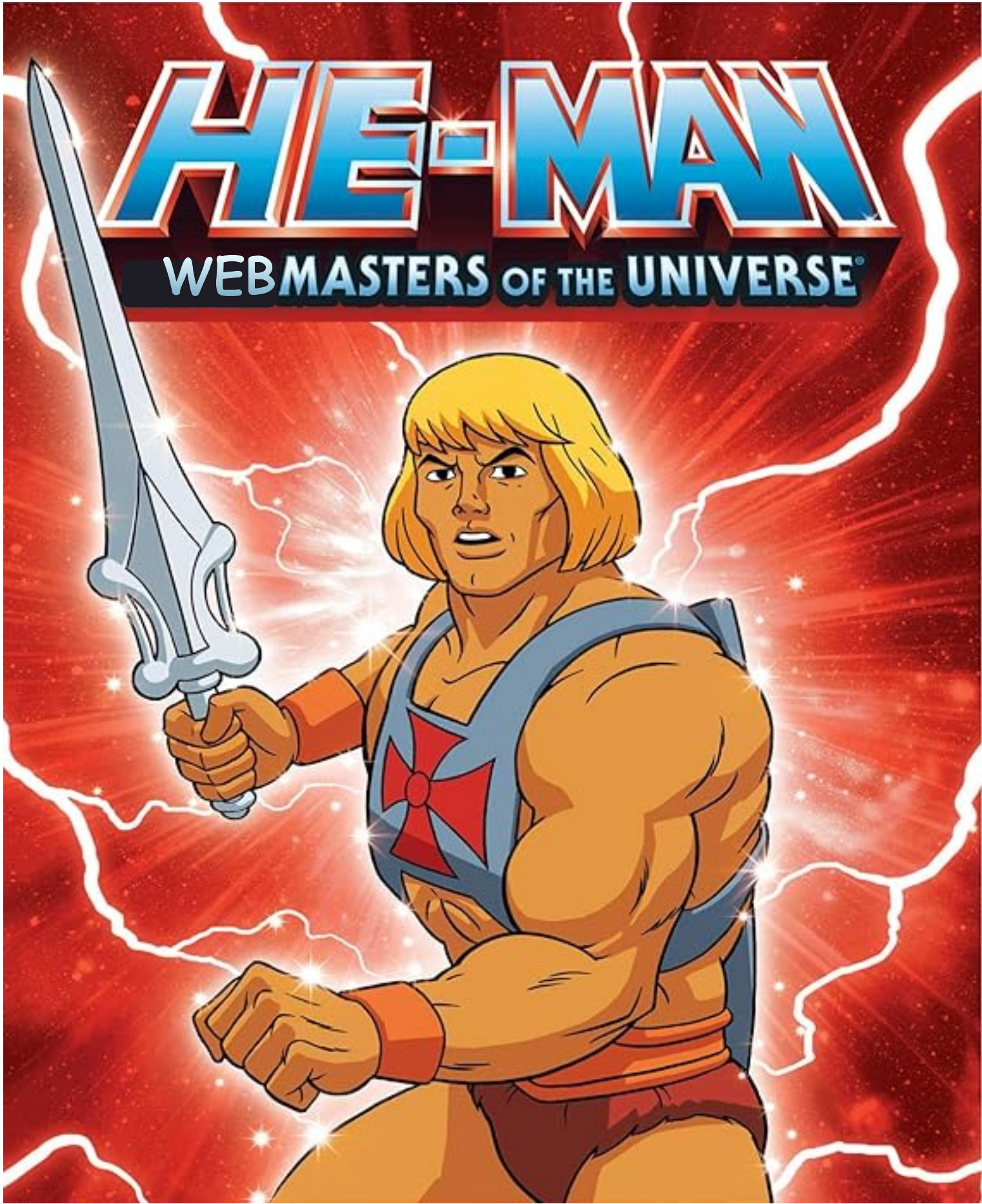
Niels Leenheer



@html5test.com



@html5test@front-end.social





**HALFSTACK**

<script>

~~<script>~~

<noscript>





<noscript>

<noscript>

</noscript>

</noscript>

<noscript style = ".....">

<noscript>

<script>

.....

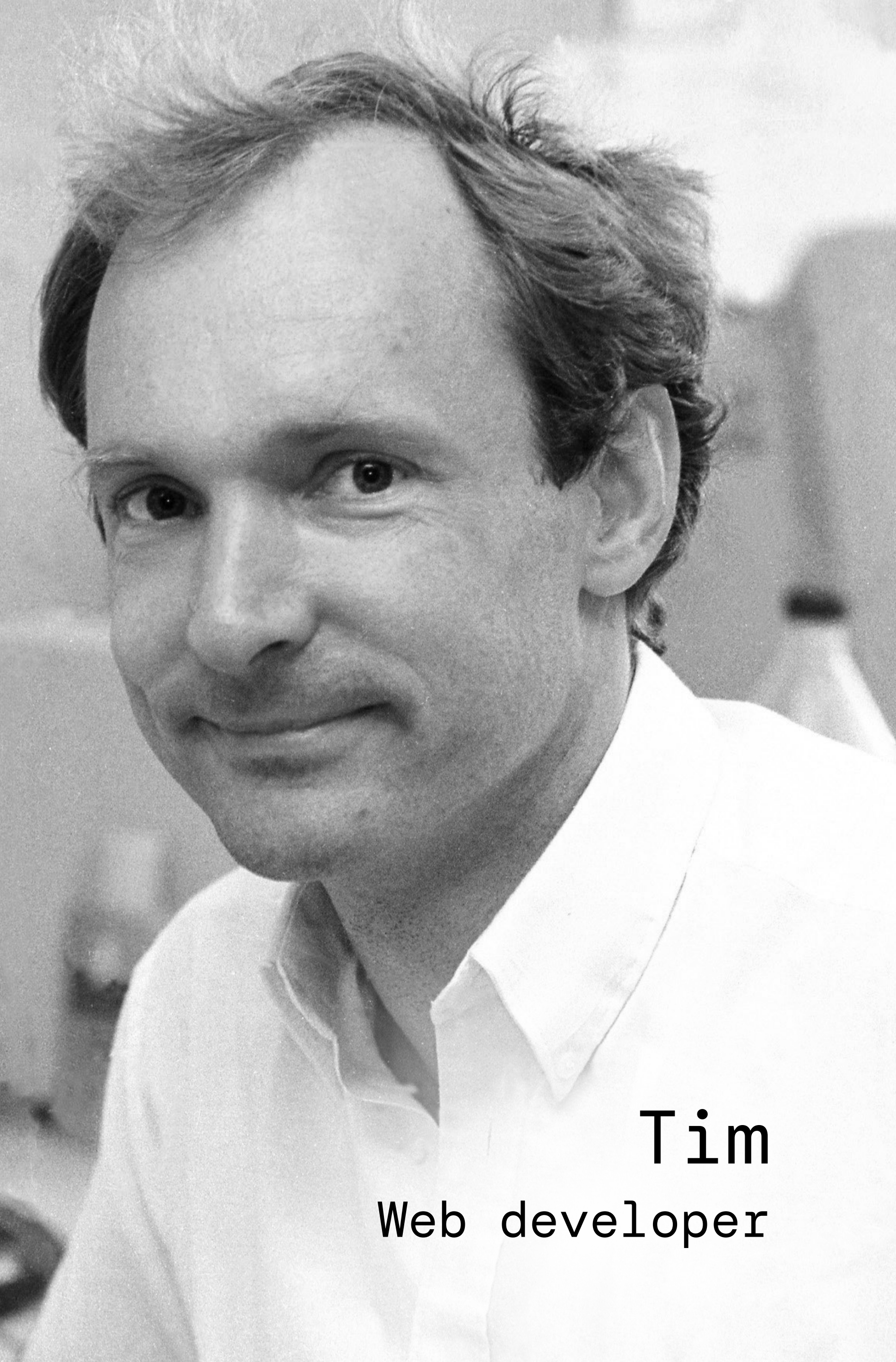
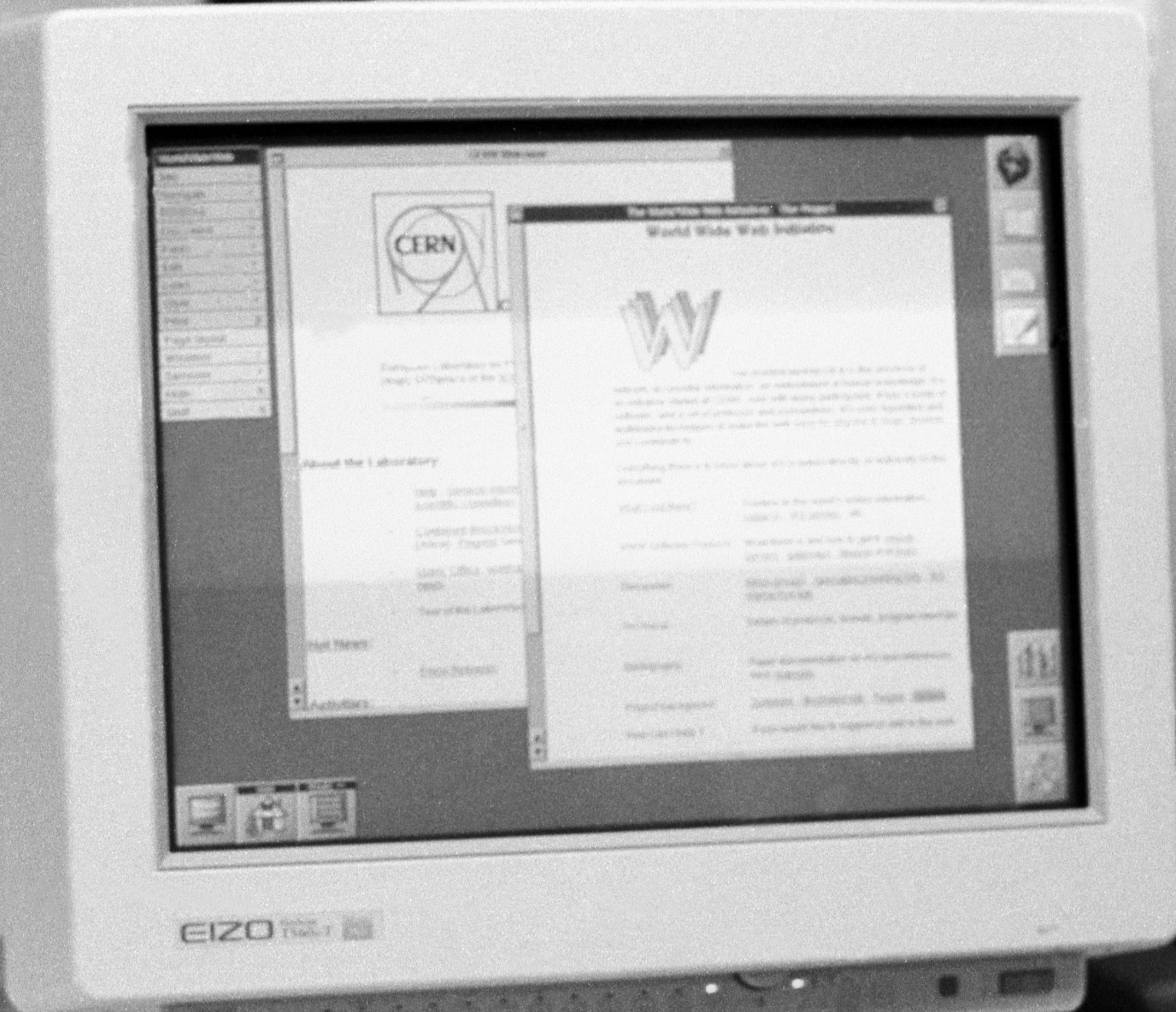
</script>

</noscript>

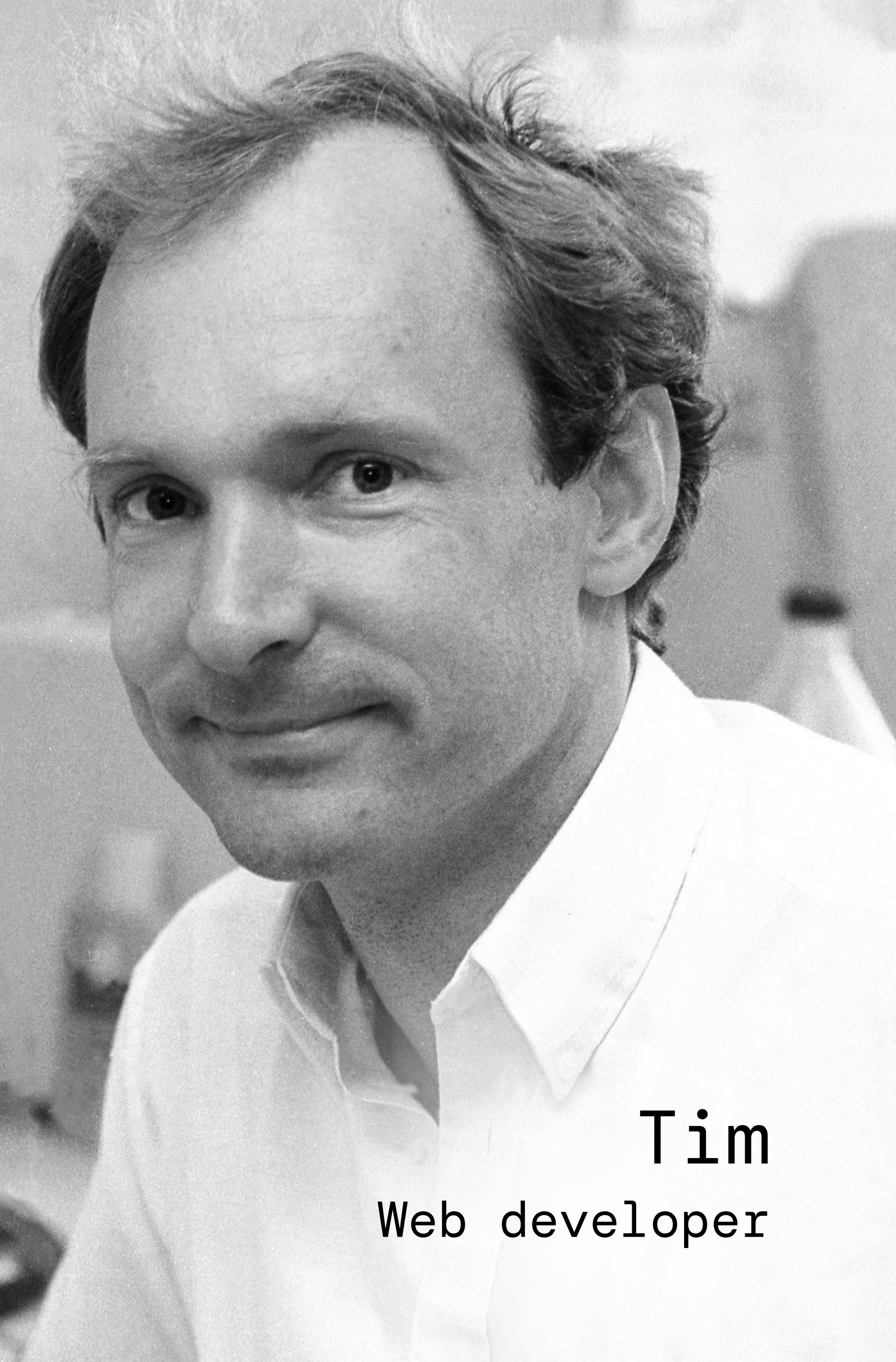
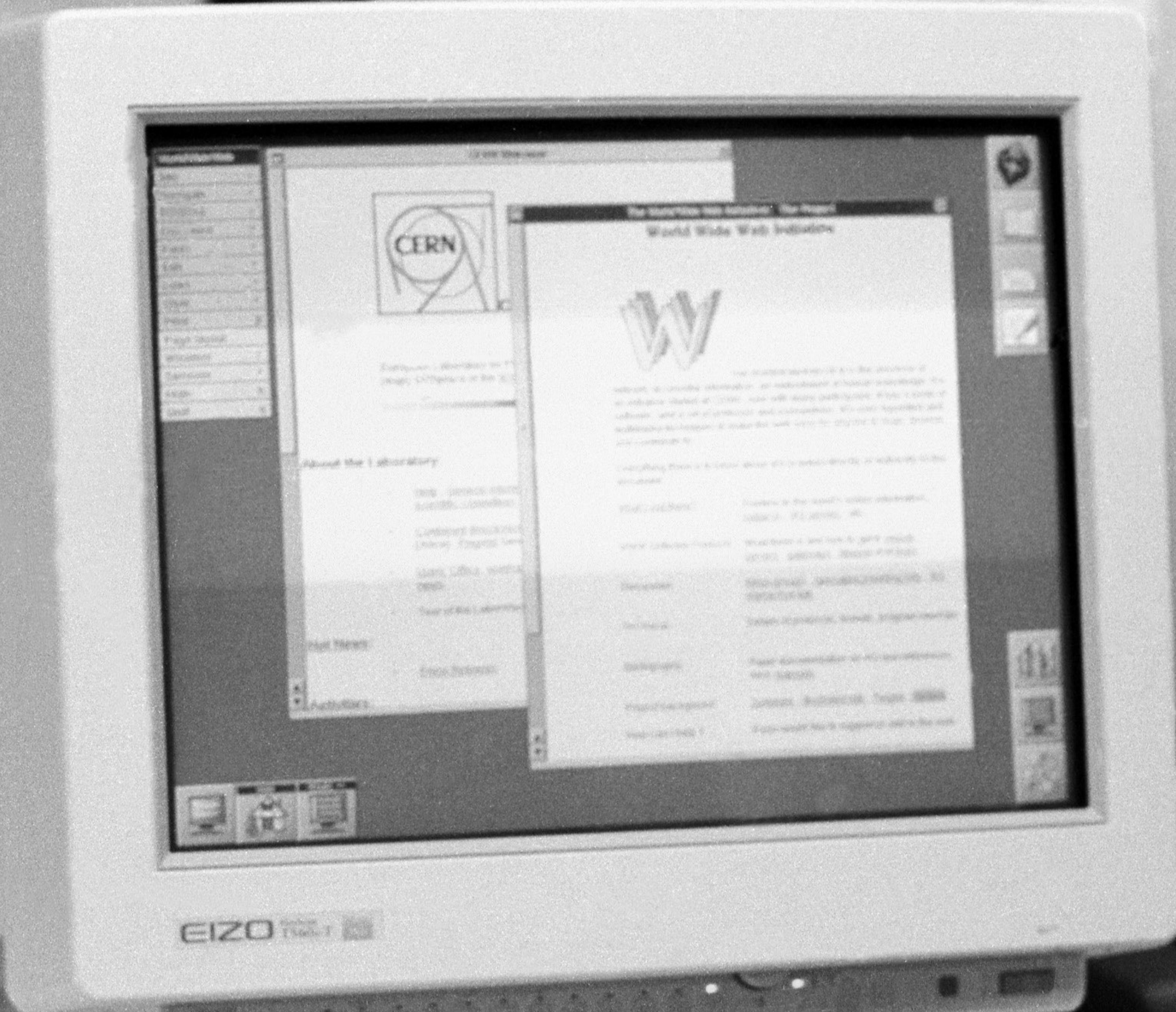


resilience

resilience



Tim  
Web developer



Tim  
Web developer



**CERN DD/OC**

**Tim Berners-Lee, CERN/DD**

**Information Management: A Proposal**

**March 1989**

---

## **Information Management: A Proposal**

### **Abstract**

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a

Vague but exciting ...

**CERN DD/OC**

**Tim Berners-Lee, CERN/DD**

**Information Management: A Proposal**

**March 1989**

---

## **Information Management: A Proposal**

### **Abstract**

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a

<b>WorldWideWeb</b>	
Info	▶
Navigate	▶
Document	▶
Edit	▶
Links	▶
Style	▶
Print	p
Page Layout	
Windows	
Services	
Hide	h
Quit	q

**The World Wide Web project**

## World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists, Policy, November's W3 news, Frequently Asked Questions.

What's out there?      Pointers to the world's online information, subjects, W3 servers, etc.

Help      on the browser you are using

Software Products      A list of W3 project components and their current state. (e.g. Line Mode, X11 Viola, NeXTStep, Servers, Tools, Mail robot, Library.)

Technical      Details of protocols, formats, program internals etc

Bibliography      Paper documentation on W3 and references.

People      A list of some people involved in the project.

History      A summary of the history of the project.

How can I help?      If you would like to support the web..

Getting code      Getting the code by anonymous FTP, etc.





Not Secure — info.cern.ch



# World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

## [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

## [Help](#)

on the browser you are using

## [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) .)

## [Technical](#)

Details of protocols, formats, program internals etc

## [Bibliography](#)

Paper documentation on W3 and references.

## [People](#)

A list of some people involved in the project.

## [History](#)

A summary of the history of the project.

## [How can I help ?](#)

If you would like to support the web..

## [Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

```
<HEADER>
```

```
<TITLE>The World Wide Web project</TITLE>
```

```
<NEXTID N="55">
```

```
</HEADER>
```

```
<BODY>
```

```
<H1>World Wide Web</H1>The WorldWideWeb (W3)
```

```
is a wide-area
```

```
<A NAME=0 HREF="WhatIs.html">
```

```
hypermedia</A> information retrieval
```

```
initiative aiming to give universal
```

```
access to a large universe of documents.<P>
```

```
Everything there is online about
```

```
W3 is linked directly or indirectly
```

```
to this document, including an <A
```

```
<HEADER>
```

```
<TITLE>The World Wide Web project</TITLE>
```

```
<NEXTID N="55">
```

```
</HEADER>
```

```
<BODY>
```

```
<H1>World Wide Web</H1>The WorldWideWeb (W3)
```

```
is a wide-area
```

```
<A NAME=0 HREF="WhatIs.html">
```

```
hypermedia</A> information retrieval
```

```
initiative aiming to give universal
```

```
access to a large universe of documents.<P>
```

```
Everything there is online about
```

```
W3 is linked directly or indirectly
```

```
to this document, including an <A
```

---

WWW parsers should *ignore tags* which they do not understand, and *ignore attributes* which they do not understand of tags which they do understand.

---

```
<HEADER>
```

```
<TITLE>The World Wide Web project</TITLE>
```

```
<NEXTID N="55">
```

```
</HEADER>
```

```
<BODY>
```

```
<H1>World Wide Web</H1>The WorldWideWeb (W3)
```

```
is a wide-area
```

```
<A NAME=0 HREF="WhatIs.html">
```

```
hypermedia</A> information retrieval
```

```
initiative aiming to give universal
```

```
access to a large universe of documents.<P>
```

```
Everything there is online about
```

```
W3 is linked directly or indirectly
```

```
to this document, including an <A
```



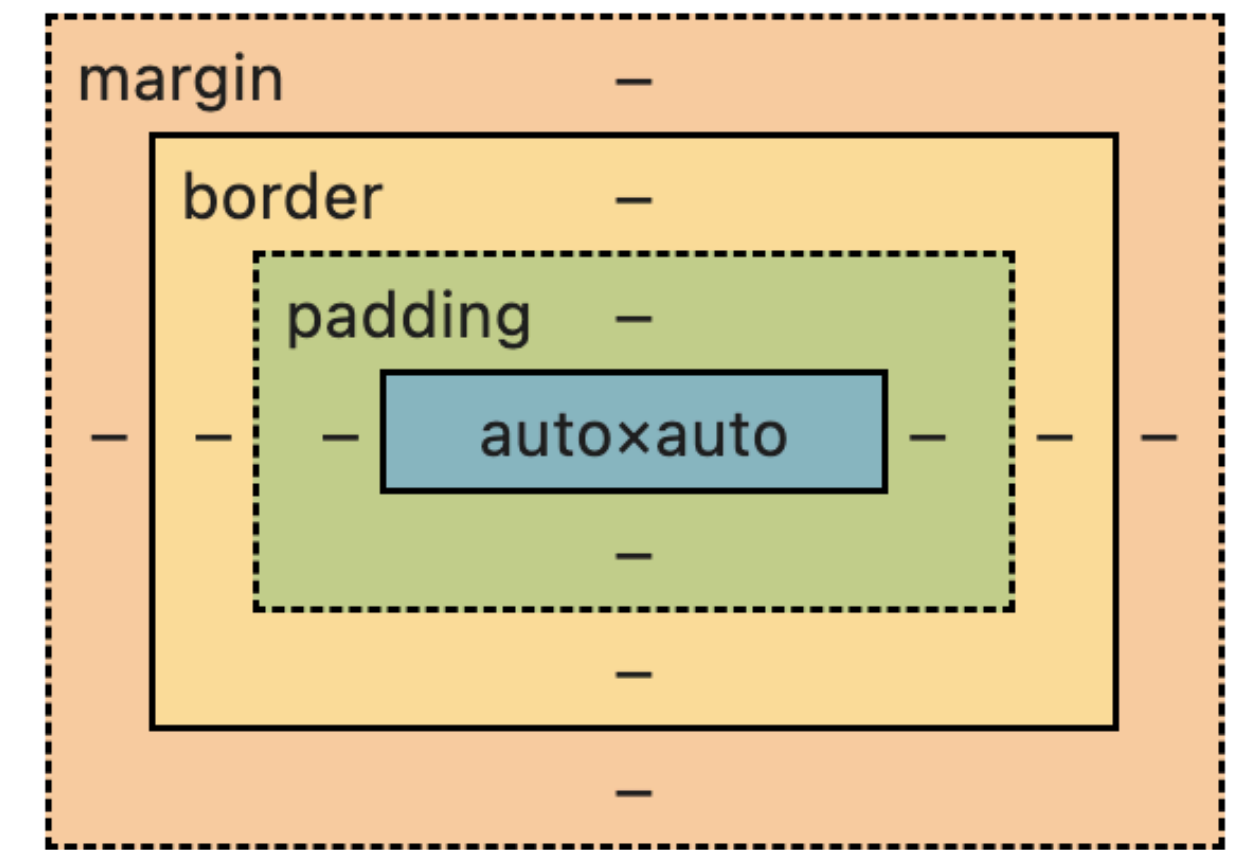


```
<html>
  <head></head>
  <body>
    <header>
      <title>HyperText Mark-up Language</title>
      <nextid n="29"> </nextid> == $0
    </header>
    <h1>...</h1>
    "The "
    <a name="0" href=" ../TheProject.html">WWW</a>
    " system uses marked-up text to represent a hypertext document
    for transmission over the network. The hypertext mark-up language
    is an "
    <a name="7" href="SGML.html">SGML</a>
    " format. WWW parsers should ignore tags which they do not
    understand, and ignore attributes which they do not understand
    of tags which they do understand."
    <p>...</p>
    <dl>
      <dt>...</dt>
      <dd> How text is interpreted when uninfluenced by tags </dd>
      <dt>...</dt>
```

Styles Computed Layout >>

Filter :hov .cls +

```
element.style {
}
```



<noscript>

What is this... 1995?

Enable your JavaScript!

</noscript>

<noscript>

What is this... 1995?

Enable your JavaScript!

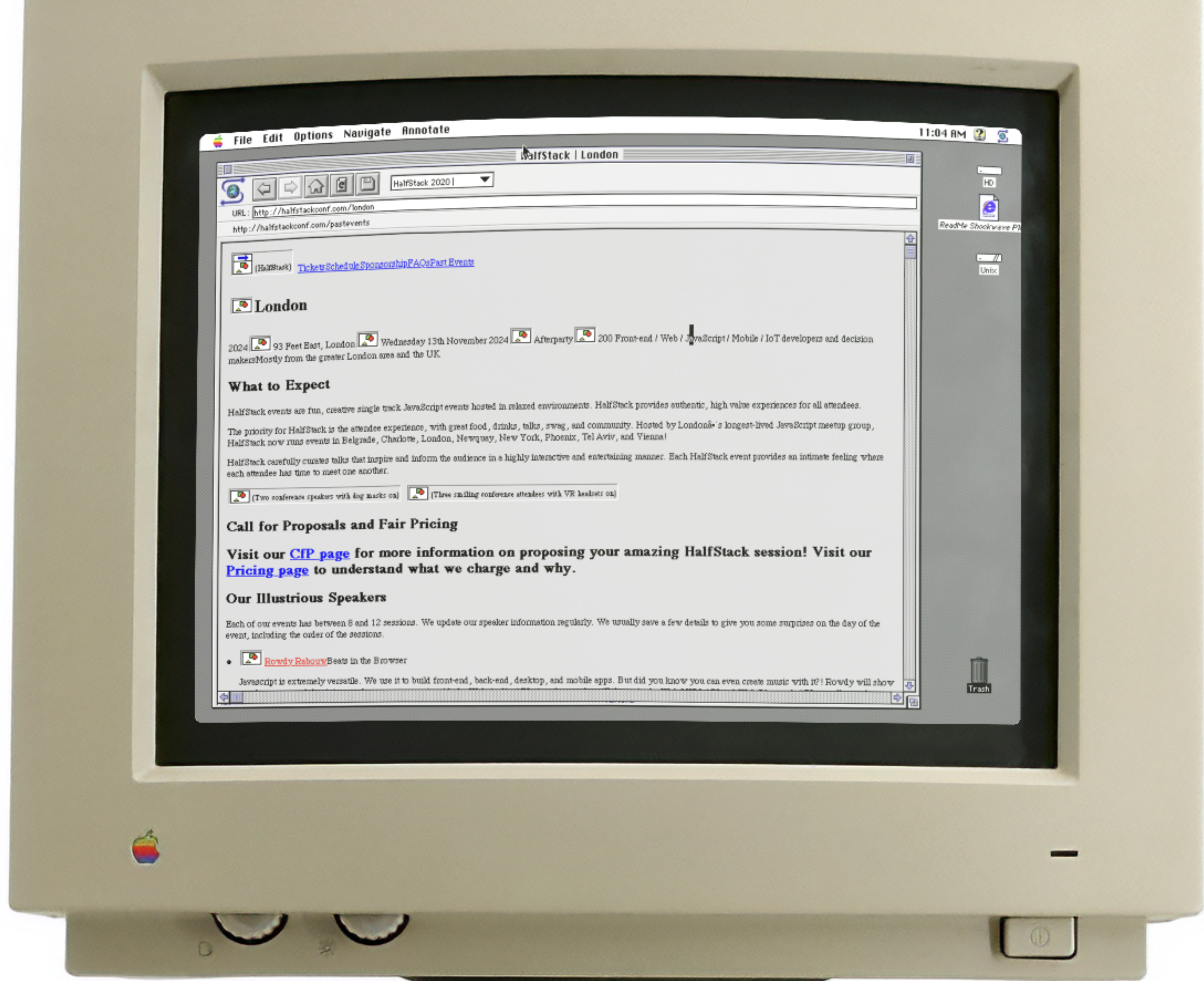
</noscript>

`<noscript>`

What is this... 1995?

Enable your JavaScript!

`</noscript>`



## London

2024 93 Feet East, London Wednesday 13th November 2024 Afterparty 200 Front-end / Web / JavaScript / Mobile / IoT developers and decision makers Mostly from the greater London area and the UK

### What to Expect

HalfStack events are fun, creative single track JavaScript events hosted in relaxed environments. HalfStack provides authentic, high value experiences for all attendees. The priority for HalfStack is the attendee experience, with great food, drinks, talks, swag, and community. Hosted by London's longest-lived JavaScript meetup group, HalfStack now runs events in Belgrade, Charlotte, London, Newquay, New York, Phoenix, Tel Aviv, and Vienna!

HalfStack carefully curates talks that inspire and inform the audience in a highly interactive and entertaining manner. Each HalfStack event provides an intimate feeling where each attendee has time to meet one another.

(Two conference speakers with dog masks on) (Three smiling conference attendees with VR headsets on)

### Call for Proposals and Fair Pricing

Visit our [CFP page](#) for more information on proposing your amazing HalfStack session! Visit our [Pricing page](#) to understand what we charge and why.

### Our Illustrious Speakers

Each of our events has between 8 and 12 sessions. We update our speaker information regularly. We usually save a few details to give you some surprises on the day of the event, including the order of the sessions.

- Rowdy Rabouw** Beats in the Browser  
JavaScript is extremely versatile. We use it to build front-end, back-end, desktop, and mobile apps. But did you know you can even create music with it? Rowdy will show



create polyfills for our favourite tags

<blink>

<blink>

my eyes! please make it stop!

</blink>



```
@keyframes blink {
  0% {
    visibility: visible;
  }
  50% {
    visibility: hidden;
  }
}

blink {
  animation: blink 1s steps(1) infinite;
}
```

<marquee>

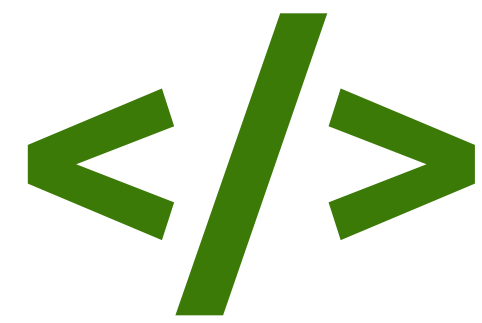
<marquee>



**Baseline 1995**



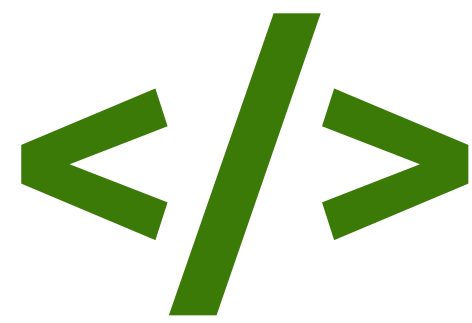




HTML



DOM



HTML

tokenizer  
&  
tree builder



DOM



*tokenizer*

analyse html character by character



`<a href="#">`

`<a href="#">`

data state

<a href="#">

tag open state

< a href="#" >

tag name state

< a href="#" >

before attribute name state

< a href="#" >

attribute name state

< a href="#" >

attribute name state

< a href="#" >

attribute name state



< a href="#" >

attribute name state

<a href=#>

before attribute value state

< a href = "#" >

attribute value (double-quoted) state

<a href="#">

attribute value (double-quoted) state

<a href=#>

after attribute value quoted state

< a href = "#" >

data state

tag name

attribute name

`<a href="#">`

attribute value

`<a href="#">`



```
{
  type: 'tag open'
  name: 'a',
  attributes: [
    {
      name: 'href',
      value: '#'
    }
  ]
}
```



Data state - Character reference in data state - RCDATA state - Character reference in RCDATA state - RAWTEXT state - Script data state - PLAINTEXT state - Tag open state - End tag open state - Tag name state - RCDATA less-than sign state - RCDATA end tag open state - RCDATA end tag name state - RAWTEXT less-than sign state - RAWTEXT end tag open state - RAWTEXT end tag name state - Script data less-than sign state - Script data end tag open state - Script data end tag name state - Script data escape start state - Script data escape start dash state - Script data escaped state - Script data escaped dash state - Script data escaped dash dash state - Script data escaped less-than sign state - Script data escaped end tag open state - Script data escaped end tag name state - Script data double escape start state - Script data double escaped state - Script data double escaped dash state - Script data double escaped dash dash state - Script data double escaped less-than sign state - Script data double escape end state - Before attribute name state - Attribute name state - After attribute name state - Before attribute value state - Attribute value (double-quoted) state - Attribute value (single-quoted) state - Attribute value (unquoted) state - Character reference in attribute value state - After attribute value (quoted) state - Self-closing start tag state - Bogus comment state - Markup declaration open state - Comment start state - Comment start dash state - Comment state - Comment end dash state - Comment end state - Comment end bang state - DOCTYPE state - Before DOCTYPE name state - DOCTYPE name state - After DOCTYPE name state - After DOCTYPE public keyword state - Before DOCTYPE public identifier state - DOCTYPE public identifier (double-quoted) state - DOCTYPE public identifier (single-quoted) state - After DOCTYPE public identifier state - Between DOCTYPE public and system identifiers state - After DOCTYPE system keyword state - Before DOCTYPE system identifier state - DOCTYPE system identifier (double-quoted) state - DOCTYPE system identifier (single-quoted) state - After DOCTYPE system identifier state - Bogus DOCTYPE state - CDATA section state

#### 8.2.4.1 Data state

Consume the [next input character](#):

↳ **U+0026 AMPERSAND (&)**

Switch to the [character reference in data state](#).

↳ **"<" (U+003C)**

Switch to the [tag open state](#).

↳ **U+0000 NULL**

[Parse error](#). Emit the [current input character](#) as a character token.

↳ **EOF**

Emit an end-of-file token.

↳ **Anything else**

Emit the [current input character](#) as a character token.

#### 8.2.4.8 Tag open state

Consume the [next input character](#):

↳ **"!" (U+0021)**

Switch to the [markup declaration open state](#).

↳ **"/" (U+002F)**

Switch to the [end tag open state](#).

↳ **Uppercase ASCII letter**

Create a new start tag token, set its tag name to the lowercase version of the [current input character](#) (add 0x0020 to the character's code point), then switch to the [tag name state](#). (Don't emit the token yet; further details will be filled in before it is emitted.)

↳ **Lowercase ASCII letter**

Create a new start tag token, set its tag name to the [current input character](#), then switch to the [tag name state](#). (Don't emit the token yet; further details will be filled in before it is emitted.)

↳ **"?" (U+003F)**

[Parse error](#). Switch to the [bogus comment state](#).

↳ **Anything else**

[Parse error](#). Switch to the [data state](#). Emit a U+003C LESS-THAN SIGN character token. Reconsume the [current input character](#).

#### 8.2.4.10 Tag name state

Consume the [next input character](#):

↳ **"tab" (U+0009)**

↳ **"LF" (U+000A)**

↳ **"FF" (U+000C)**

↳ **U+0020 SPACE**

Switch to the [before attribute name state](#).

↳ **"/" (U+002F)**

Switch to the [self-closing start tag state](#).

↳ **">" (U+003E)**

Switch to the [data state](#). Emit the current tag token.

↳ **[Uppercase ASCII letter](#)**

Append the lowercase version of the [current input character](#) (add 0x0020 to the character's code point) to the current tag token's tag name.

↳ **U+0000 NULL**

[Parse error](#). Append a U+FFFD REPLACEMENT CHARACTER character to the current tag token's tag name.

↳ **EOF**

[Parse error](#). Switch to the [data state](#). Reconsume the EOF character.

↳ **Anything else**

Append the [current input character](#) to the current tag token's tag name.

#### 8.2.4.10 Tag name state

Consume the [next input character](#):

↳ **"tab" (U+0009)**

↳ **"LF" (U+000A)**

↳ **"FF" (U+000C)**

↳ **U+0020 SPACE**

Switch to the [before attribute name state](#).

↳ **"/" (U+002F)**

Switch to the [self-closing start tag state](#).

↳ **">" (U+003E)**

Switch to the [data state](#). Emit the current tag token.

↳ **[Uppercase ASCII letter](#)**

Append the lowercase version of the [current input character](#) (add 0x0020 to the character's code point) to the current tag token's tag name.

↳ **U+0000 NULL**

[Parse error](#). Append a U+FFFD REPLACEMENT CHARACTER character to the current tag token's tag name.

↳ **EOF**

[Parse error](#). Switch to the [data state](#). Reconsume the EOF character.

↳ **Anything else**

Append the [current input character](#) to the current tag token's tag name.

tag name



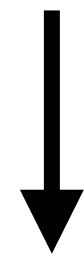
<a<a<a<a<a>

this is a tag!

</a<a<a<a<a>

<umop-əpɪsdn == sɪʏɪ>

tag name



<uMop-əp1sdn

attribute name



= = s147 >



attribute value





Nuclear Power Plant

Spider Farm

```
<my-thing *ngIf="visible">  
  <button (click)="handler()">  
    Click me!  
  </button>  
</my-thing>
```

```
<my-thing v-show="visible">  
  <button @click="handler">  
    Click me!  
  </button>  
</my-thing>
```

<c( J ° □ ° ) ~           >

    this really is a tag too!

</c( J ° □ ° ) ~           >

```
<a href="#">
```

```
    and this one too!
```

```
</a>
```

`<a href="#">`  
and this one too!  
`</a>`

#### 8.2.4.10 Tag name state

Consume the [next input character](#):

↳ **"tab" (U+0009)**

↳ **"LF" (U+000A)**

↳ **"FF" (U+000C)**

↳ **U+0020 SPACE**

Switch to the [before attribute name state](#).

↳ **"/" (U+002F)**

Switch to the [self-closing start tag state](#).

↳ **">" (U+003E)**

Switch to the [data state](#). Emit the current tag token.

↳ **[Uppercase ASCII letter](#)**

Append the lowercase version of the [current input character](#) (add 0x0020 to the character's code point) to the current tag token's tag name.

↳ **U+0000 NULL**


[Parse error](#). Append a U+FFFD REPLACEMENT CHARACTER character to the current tag token's tag name.

↳ **EOF**

[Parse error](#). Switch to the [data state](#). Reconsume the EOF character.

↳ **Anything else**

Append the [current input character](#) to the current tag token's tag name.



`<a href="#">`

and this one too!

`</a>`



not a space



```
<a href="#">
```

and this one too!

```
</a>
```

tag name



<a href="#">

and this one too!

</a>

```
<a href="#">
```

```
    and this one too!
```

```
</a href="#">
```

```
find ~ -type f -name '*.html' -print0 |  
xargs -0 -L1 perl -C -p -i -e 's/<a h/<a\x{2007}h/g'
```

```
find ~ -type f -name '*.html' -print0 |  
xargs -0 -L1 perl -C -p -i -e 's/<a h/<a\x{2007}h/g'
```

```
find ~ -type f -name '*.html' -print0 |  
xargs -0 -L1 perl -C -p -i -e 's/<a h/<a\x{2007}h/g'
```

lol my bank just called me

because you can give your  
accounts nicknames to  
remember which savings acct is  
which

and I put an emoji in one of  
them

and apparently somehow broke  
their entire banking system

so I guess... don't do that

LOLOLOL

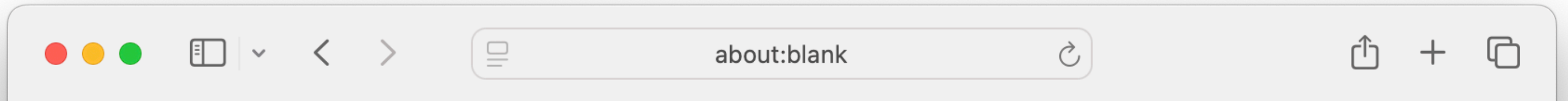
<i->unicode</i->



<i-❤>unicode</i-❤>

```
<i-❤>  
  HTML  
</i-❤>
```

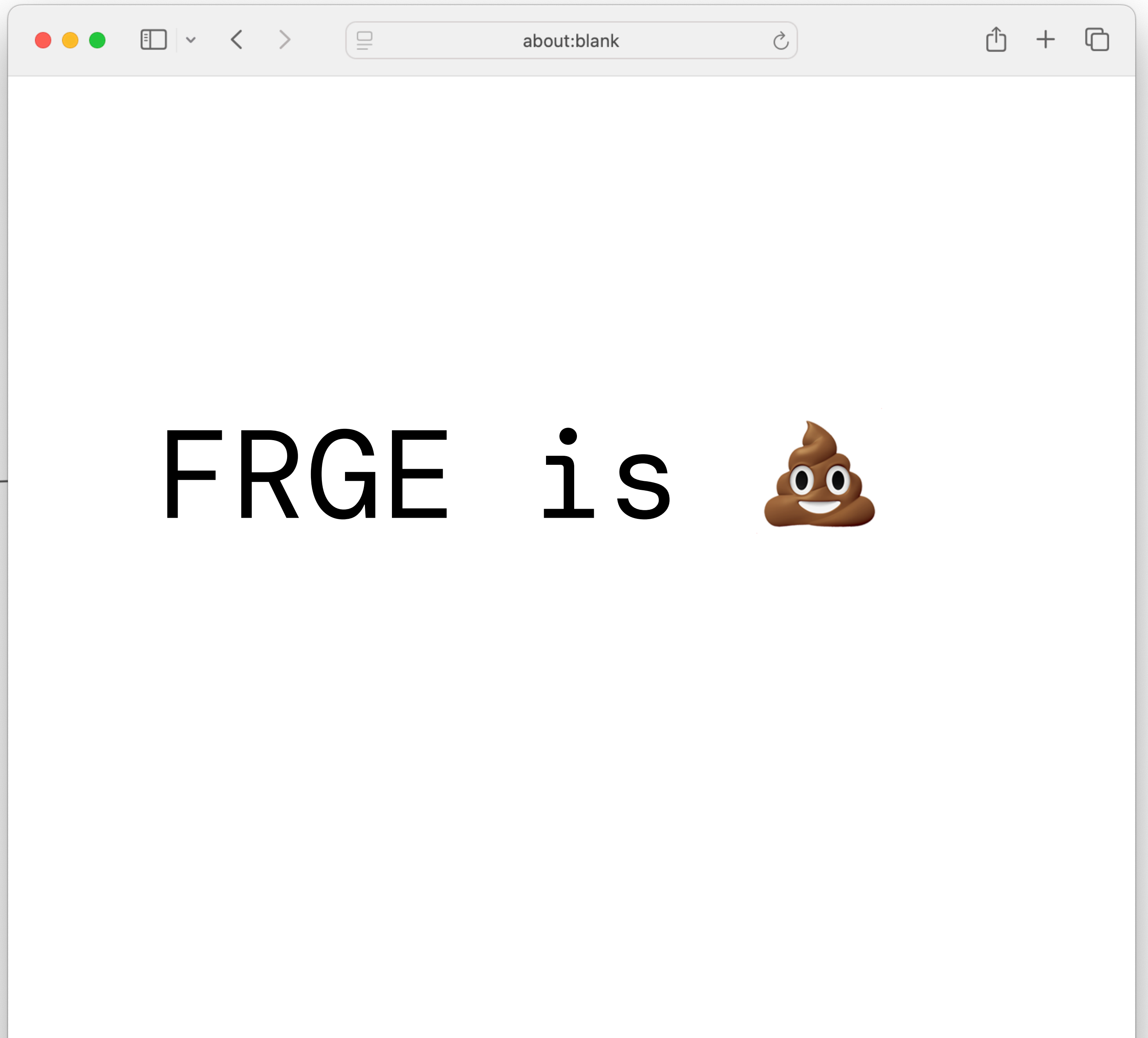
```
i-❤:before {  
  content: 'I ❤';  
}
```



I ❤ HTML

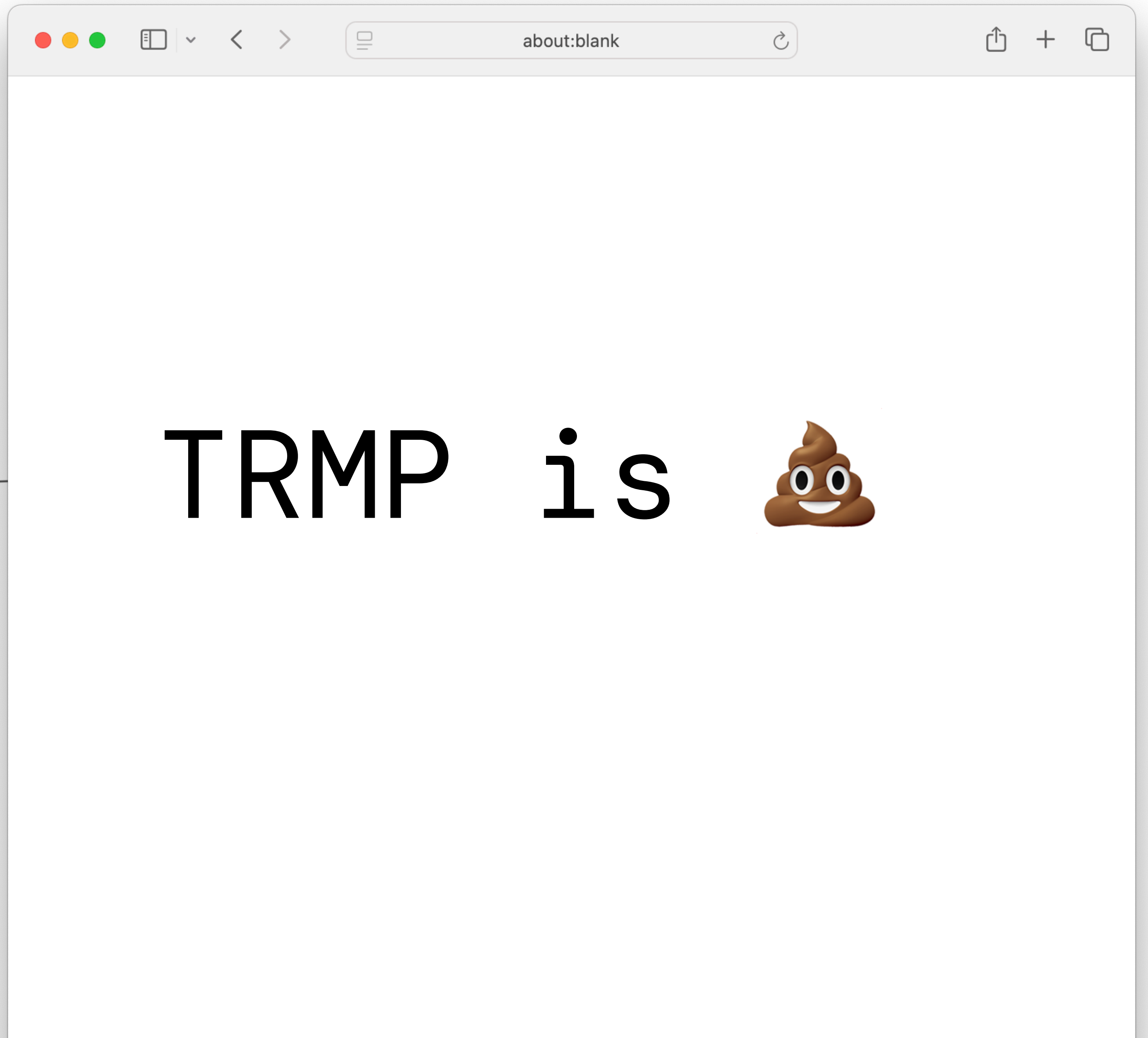
```
<is-💩>  
  FRGE  
</is-💩>
```

```
is-💩:after {  
  content: 'is 💩';  
}
```



```
<is-💩>  
  TRMP  
</is-💩>
```

```
is-💩:after {  
  content: 'is 💩';  
}
```



<html ⚡ >



<my-button></my-button>

```
<div class='button'></div>
```




<button></button>





*tree builder*

`<a href="#">``</a>`

```
{
  type: 'tag open'
  name: 'a',
  attributes: [
    {
      name: 'href',
      value: '#'
    }
  ]
}
```

```
{
  type: 'character'
  value: '👍'
}
```

```
{
  type: 'tag close'
  name: 'a'
}
```

HTMLAnchorElement



Text : 👍

*creating missing elements*

`<a href="#"> 👍 </a>`

initial insertion mode



```
<a href="#"> 👍 </a>
```

before html insertion mode

```
<a href="#"> 👍 </a>
```

HTMLHtmlElement

before head insertion mode

`<a href="#"> 👍 </a>`

HTMLHtmlElement

HTMLHeadElement

in head insertion mode

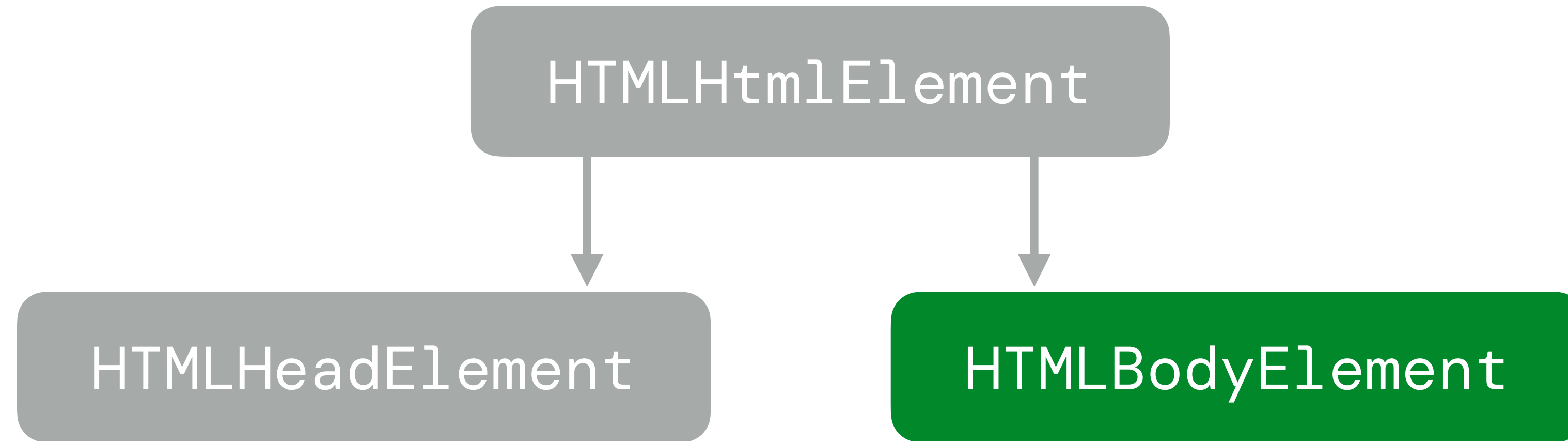
`<a href="#"> 👍 </a>`

HTMLHtmlElement

HTMLHeadElement

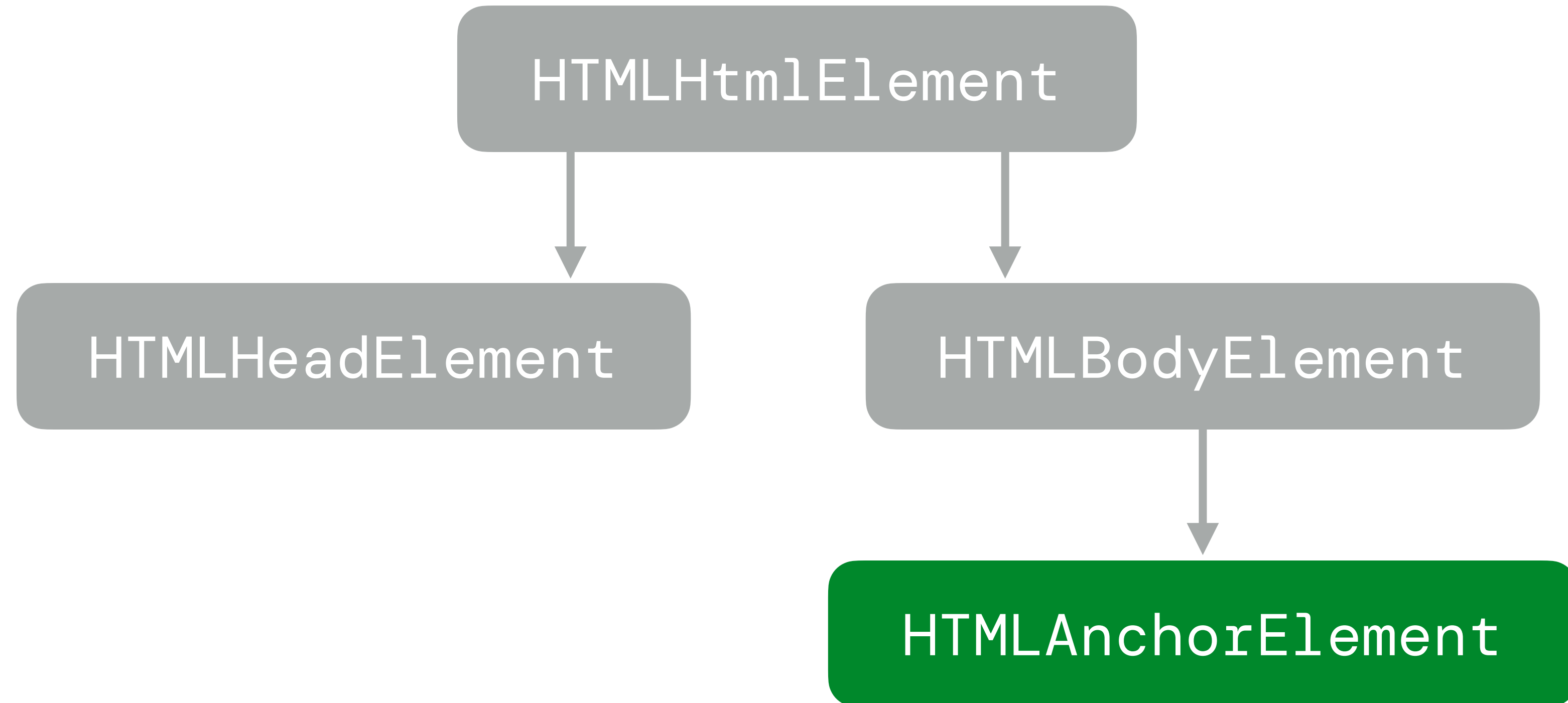
after head insertion mode

`<a href="#"> 👍 </a>`



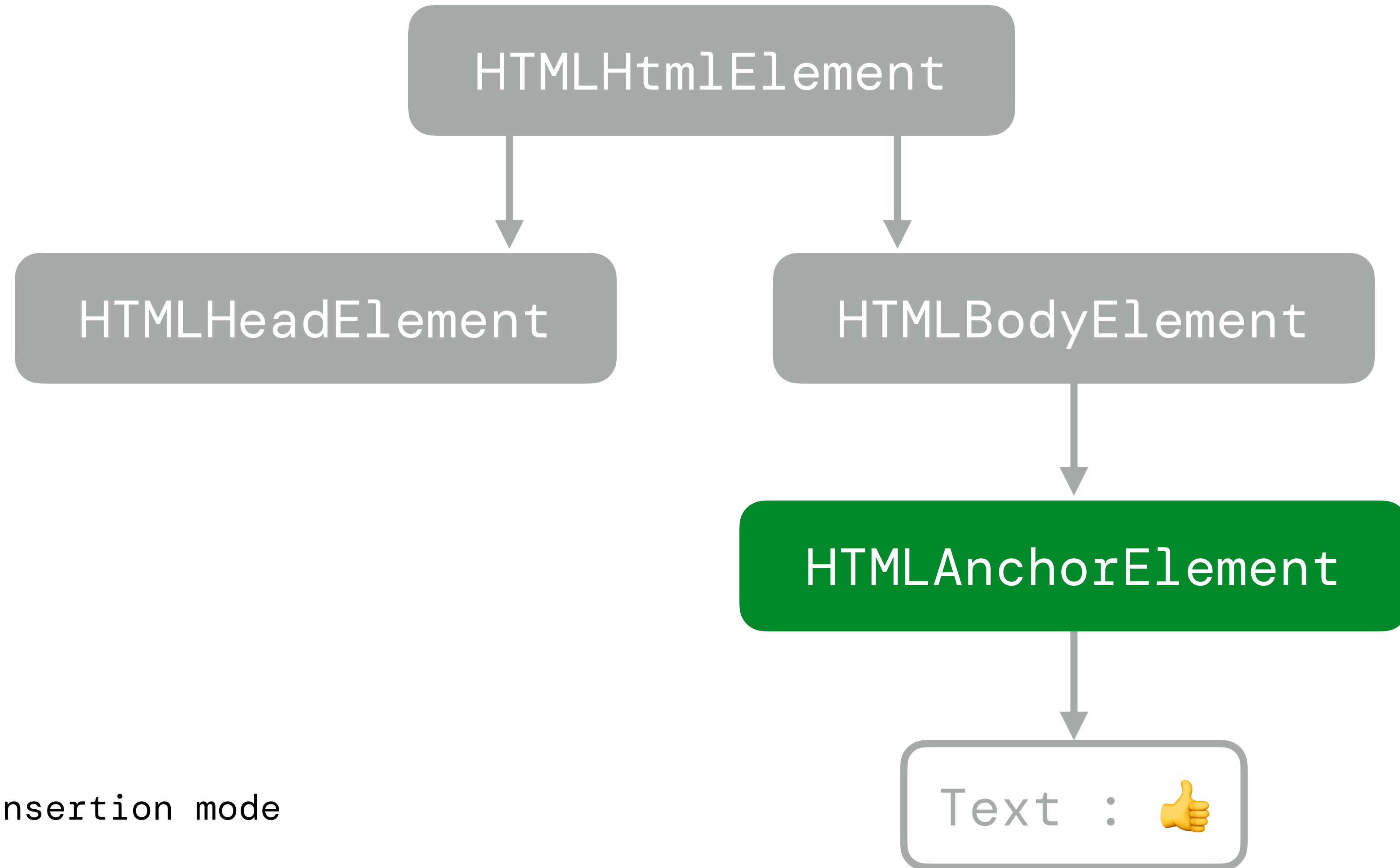
in body insertion mode

<a href="#">👍 </a>



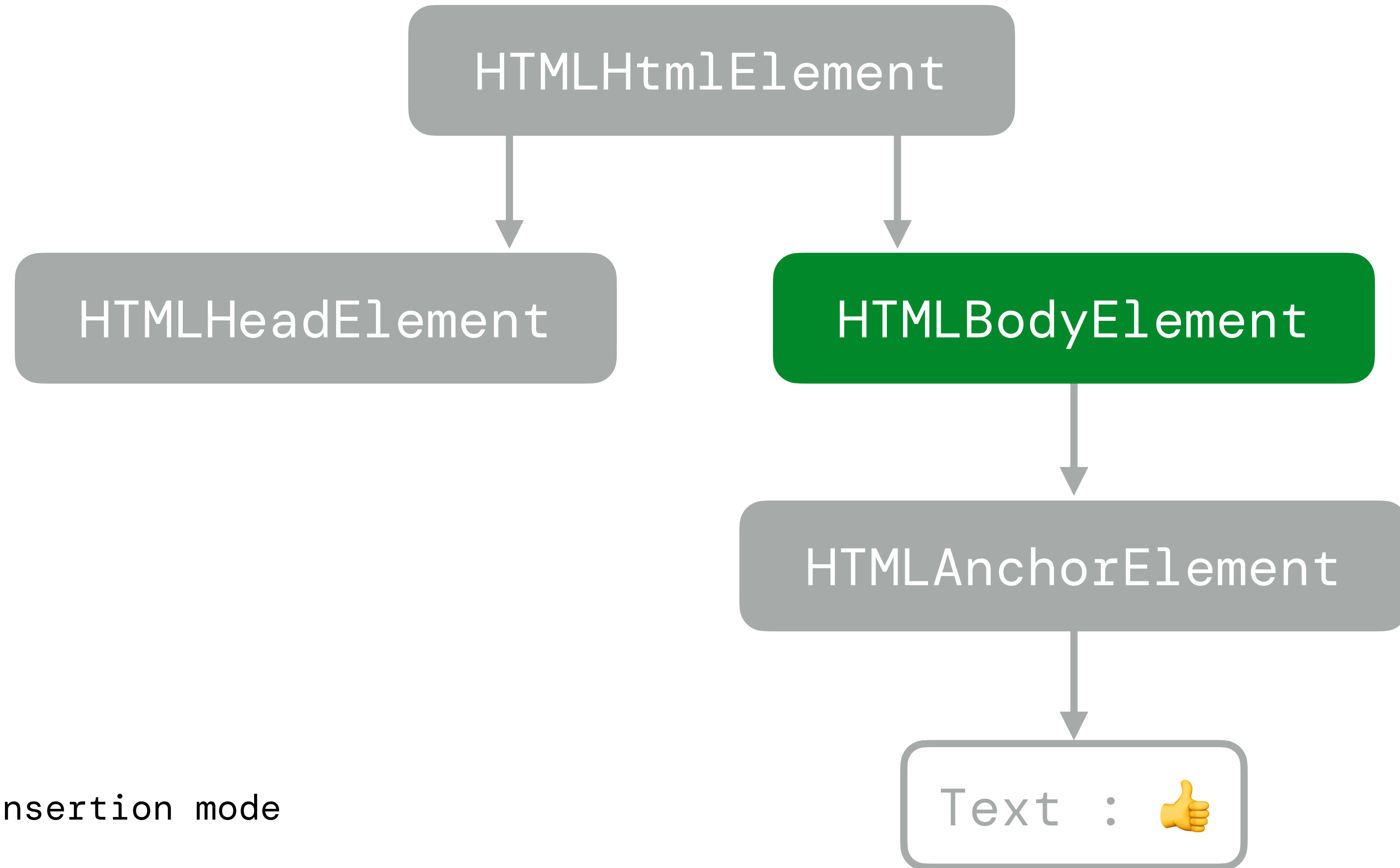
in body insertion mode

`<a href="#">` | `</a>`



in body insertion mode

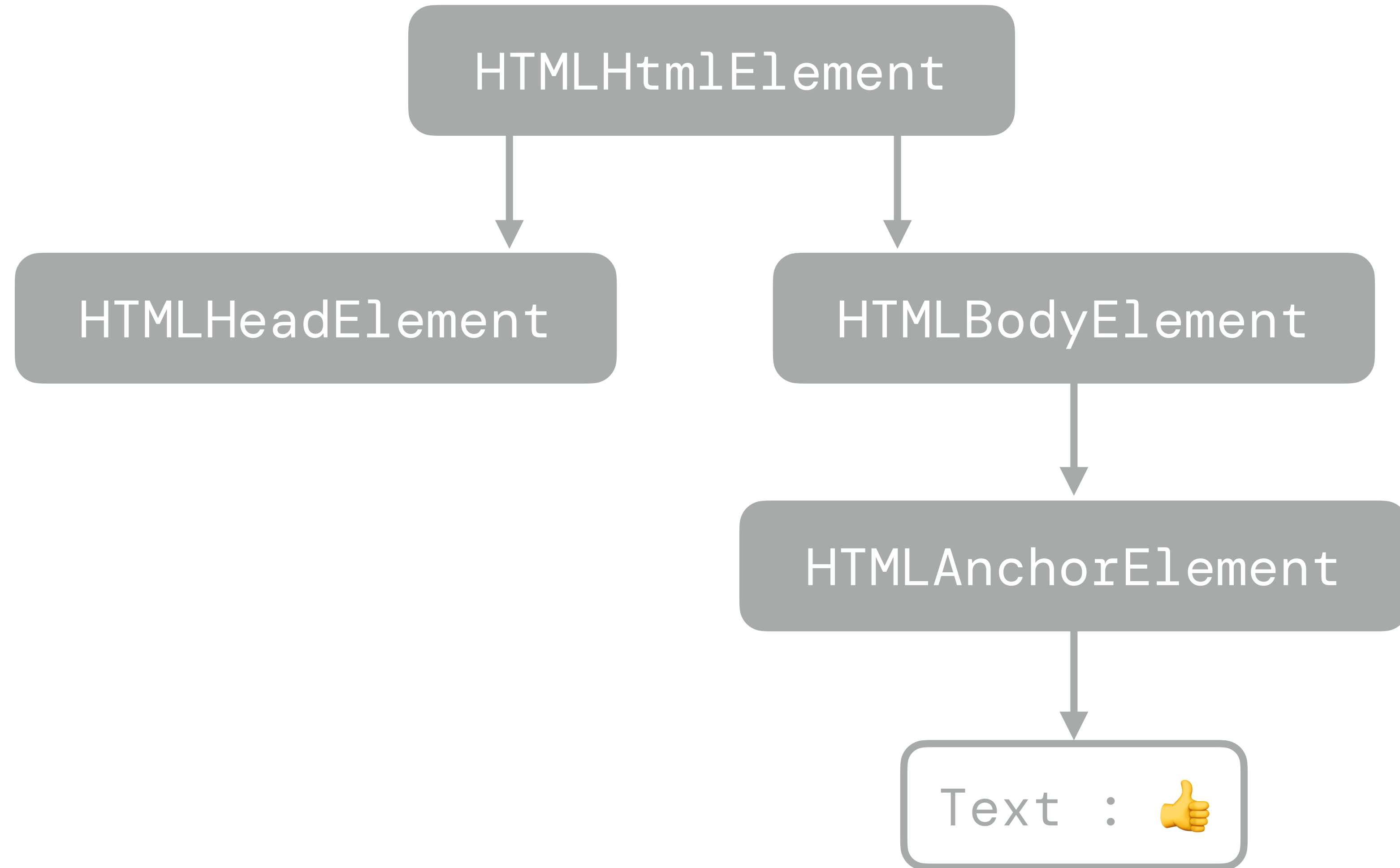
```
<a href="#"> </a>
```



in body insertion mode



`<a href="#"> </a>`



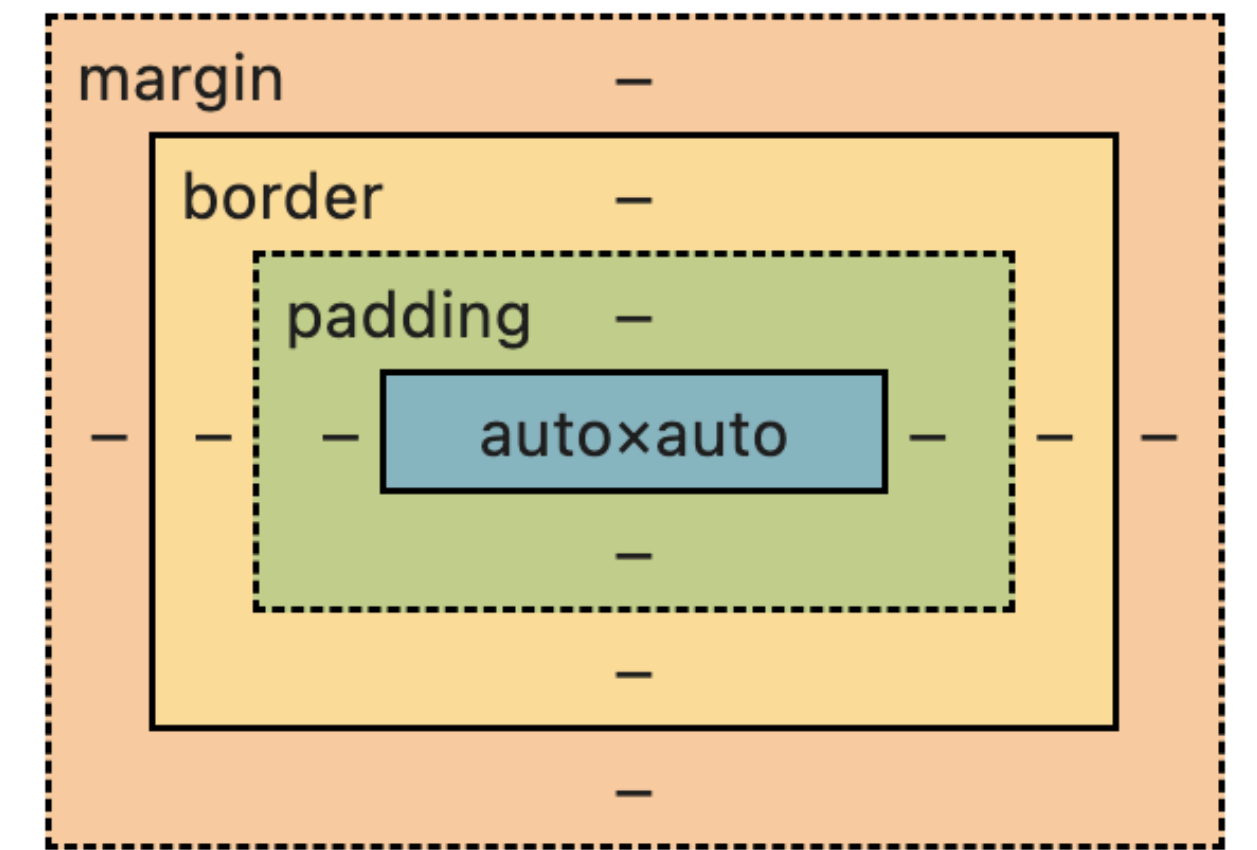


```
<html>
  <head></head>
  <body>
    <header>
      <title>HyperText Mark-up Language</title>
      ... <nextid n="29"> </nextid> == $0
    </header>
    <h1>...</h1>
    "The "
    <a name="0" href=" ../TheProject.html">WWW</a>
    " system uses marked-up text to represent a hypertext document
    for transmission over the network. The hypertext mark-up language
    is an "
    <a name="7" href="SGML.html">SGML</a>
    " format. WWW parsers should ignore tags which they do not
    understand, and ignore attributes which they do not understand
    of tags which they do understand."
    <p>...</p>
    <dl>
      <dt>...</dt>
      <dd> How text is interpreted when uninfluenced by tags </dd>
      <dt>...</dt>
```

Styles Computed Layout >>

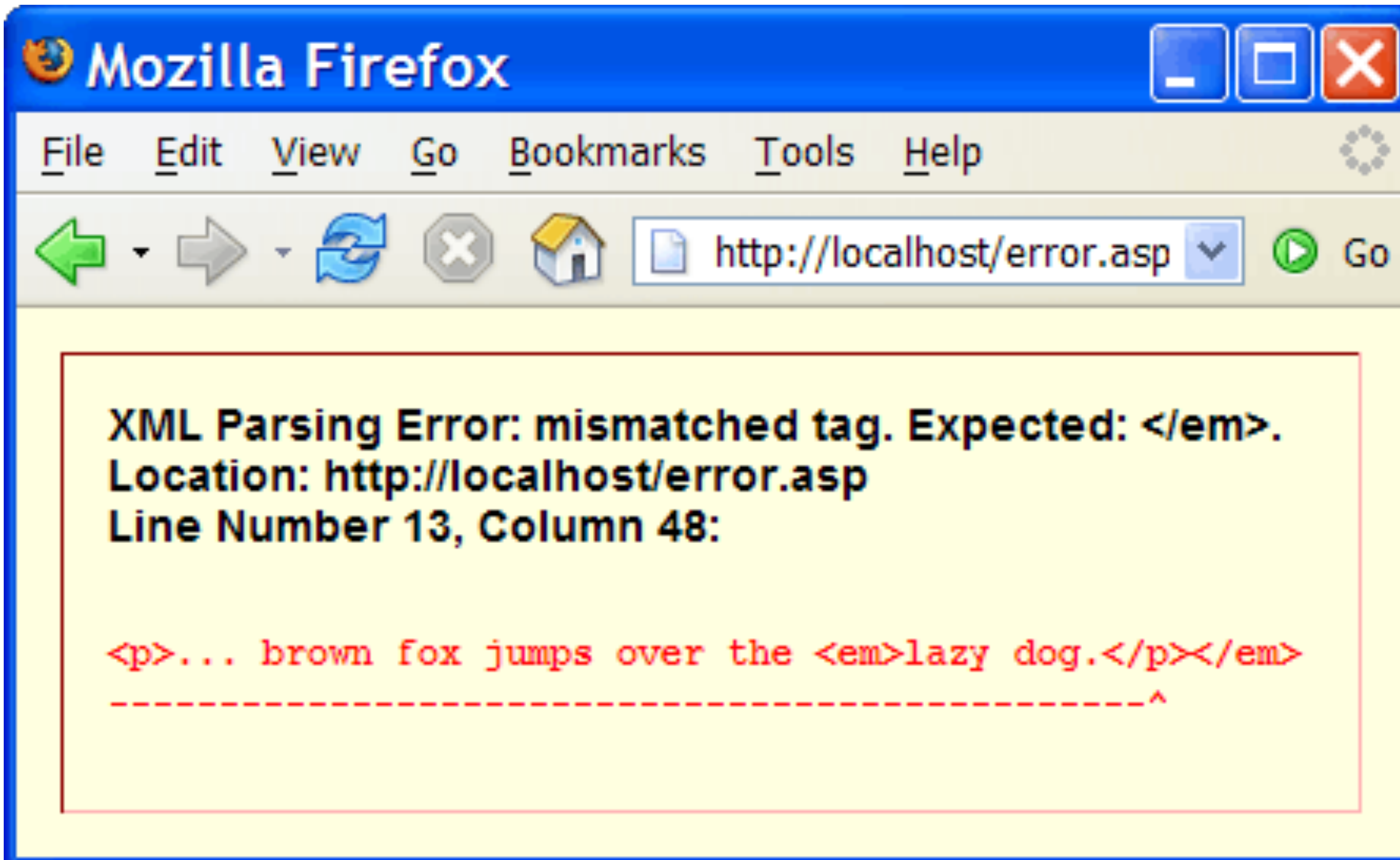
Filter :hov .cls +

```
element.style {
}
```



*fixing mistakes*

<b> 1 <i> 2 </b> 3 </i>



**Mozilla Firefox**

File Edit View Go Bookmarks Tools Help

← → ↻ × 🏠  Go

**XML Parsing Error: mismatched tag. Expected: </em>.**  
**Location: http://localhost/error.asp**  
**Line Number 13, Column 48:**

```
<p>... brown fox jumps over the <em>lazy dog.</p></em>  
-----^
```

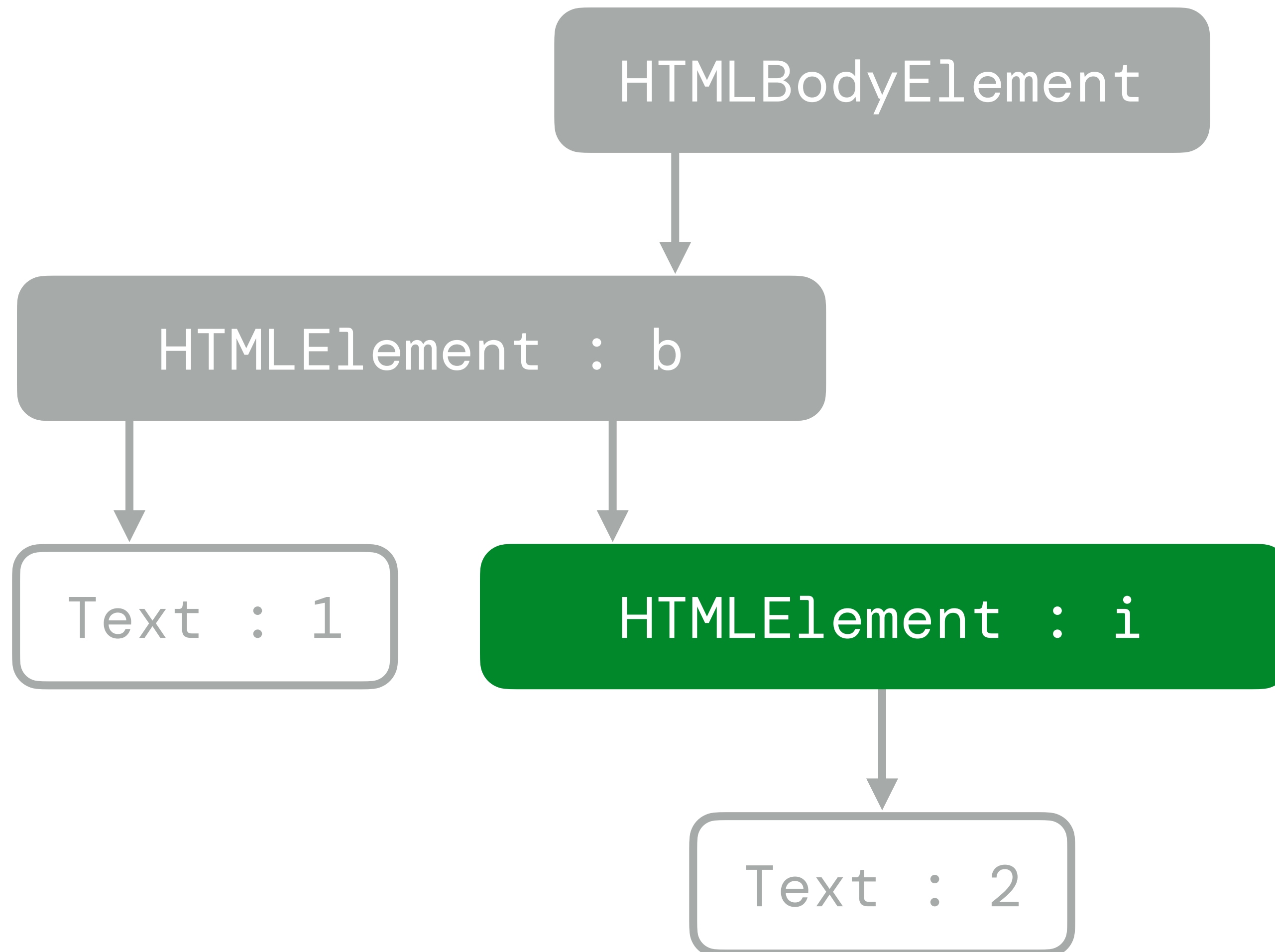
<b> 1 <i> 2 </b> 3 </i>

```
<b> 1 <i> 2 </b> 3 </i>
```

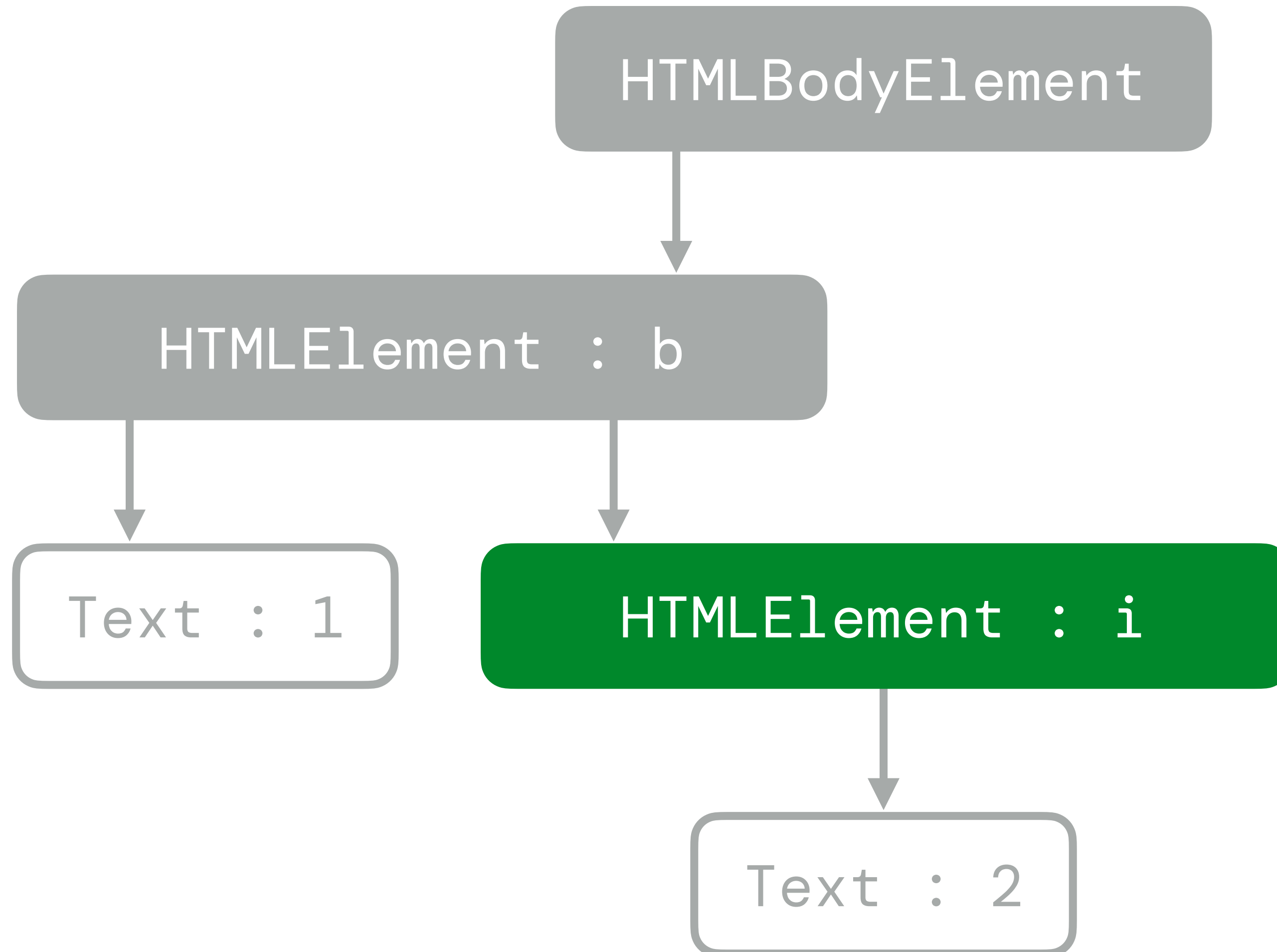
HTMLBodyElement



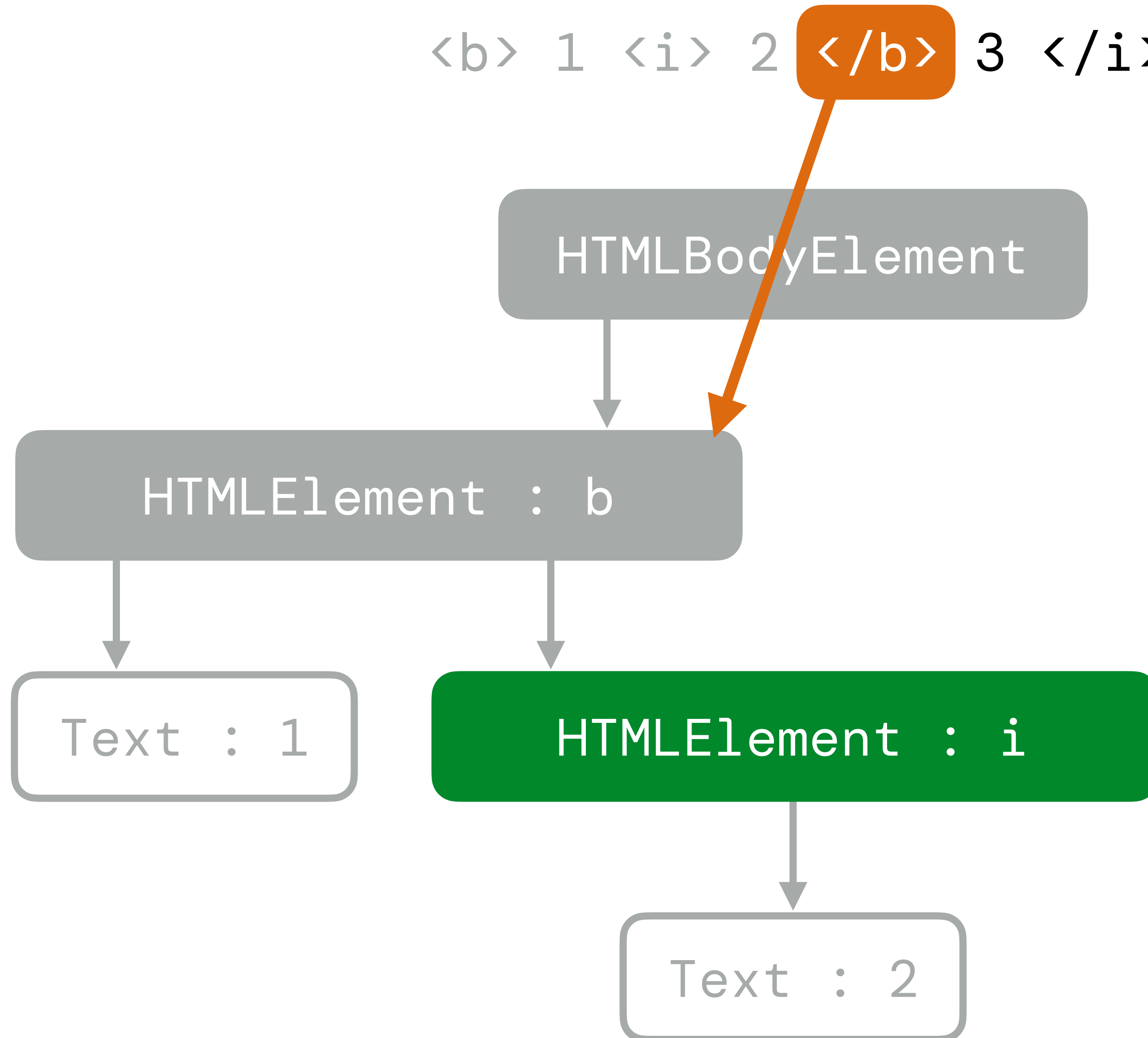
<b> 1 <i> 2 </b> 3 </i>



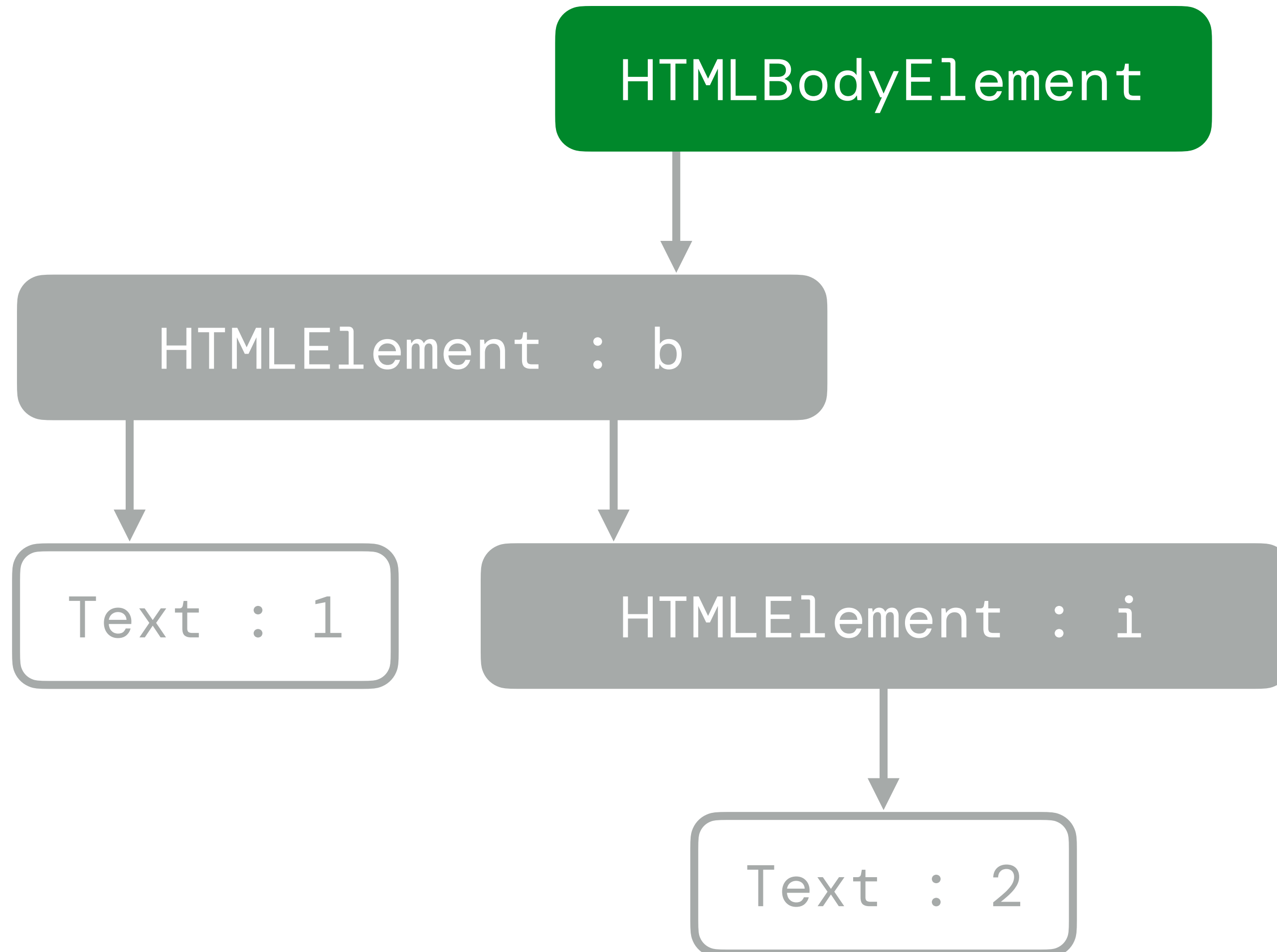
<b> 1 <i> 2 **</b>** 3 </i>



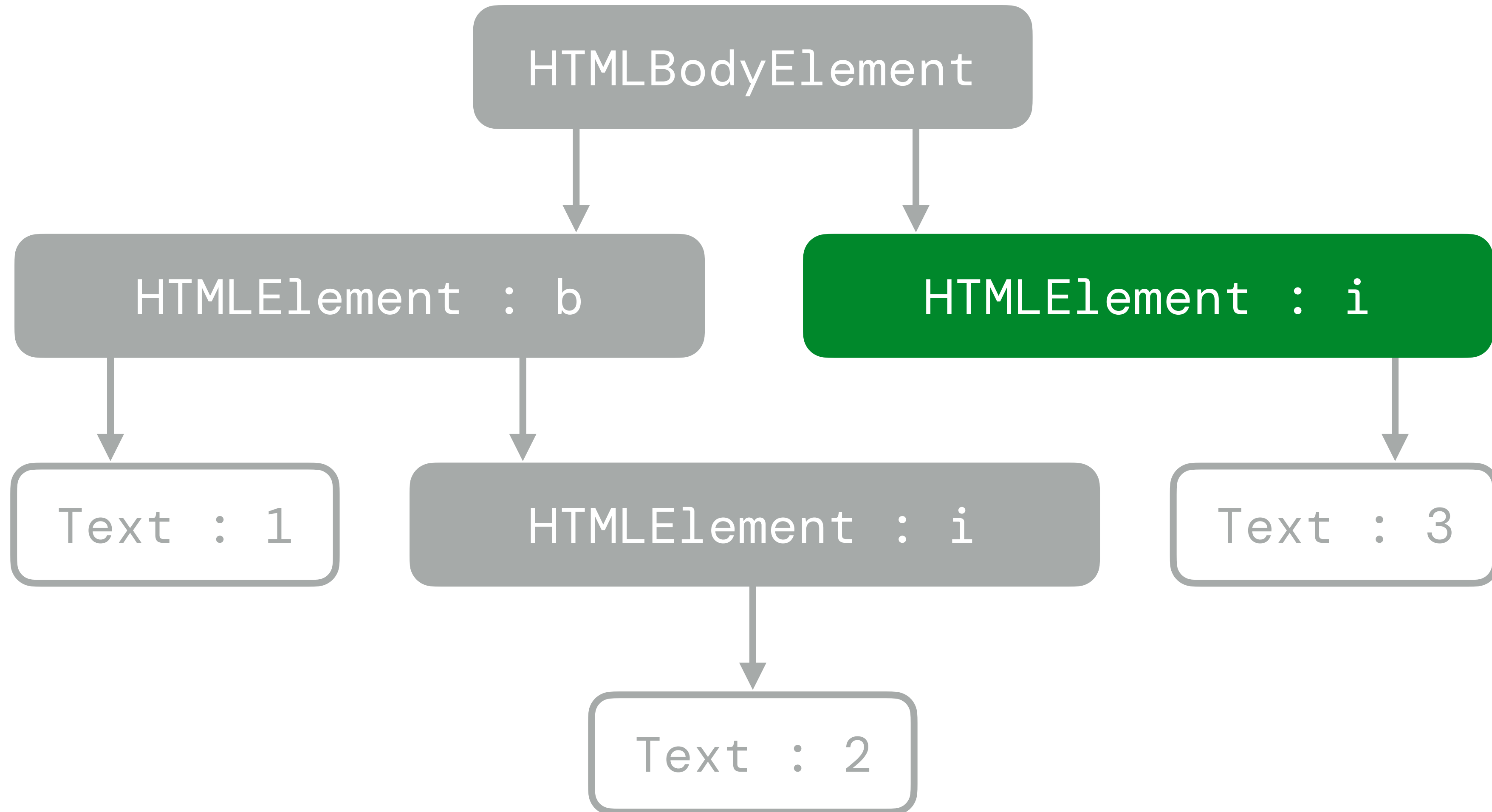
<b> 1 <i> 2 </b> 3 </i>



<b> 1 <i> 2 </b> 3 </i>



<b> 1 <i> 2 </b> 3 </i>





# W3C<sup>®</sup> Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** [Notes and Potential Issues](#) [Validation Output](#)

## Errors found while checking this document as HTML 2.0!

<b>Result:</b>	3 Errors, 3 warning(s)	
<b>Address :</b>	<input type="text" value="http://info.cern.ch/hypertext/WWW/TheProject.html"/>	
<b>Encoding :</b>	utf-8	<input type="text" value="(detect automatically)"/>
<b>Doctype :</b>	HTML 2.0	<input type="text" value="HTML 2.0"/>
<b>Root Element:</b>	html	



The W3C validators rely on community support for hosting and development.

5643

```
NAME=15 HREF="History.html">History</A>
```

```
<DD> A summary of the history  
of the project.
```

```
<DT><A
```

```
NAME=37 HREF="Helping.html">How can I help</A> ?
```

```
<DD> If you would like  
to support the web..
```

```
<DT><A
```

```
NAME=48 HREF=" ../README.html">Getting code</A>
```

```
<DD> Getting the code by<A
```

```
NAME=49 HREF="LineMode/Defaults/Distribution.html">
```

```
anonymous FTP</A> , etc.</A>
```

```
</DL>
```

```
</BODY>
```



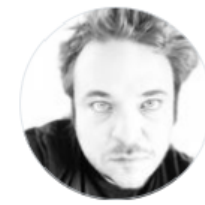












**Thomas Fuchs**



@thomasfuchs.at

The Internet was supposed to democratize knowledge and enable publishing and communication without barriers.

Instead we got a few billionaires controlling propaganda feeds in the form of videos and LLMs.

It's way worse than the newspaper barons of the late 19th century.

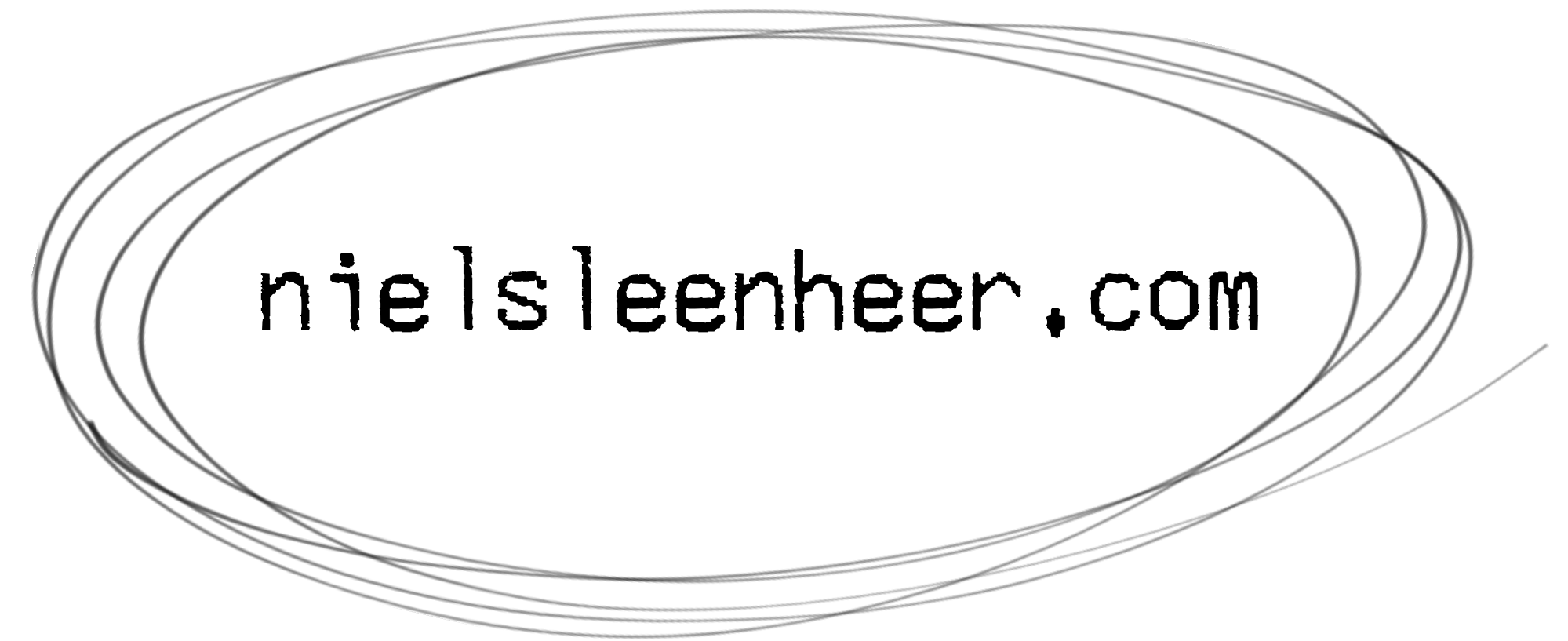
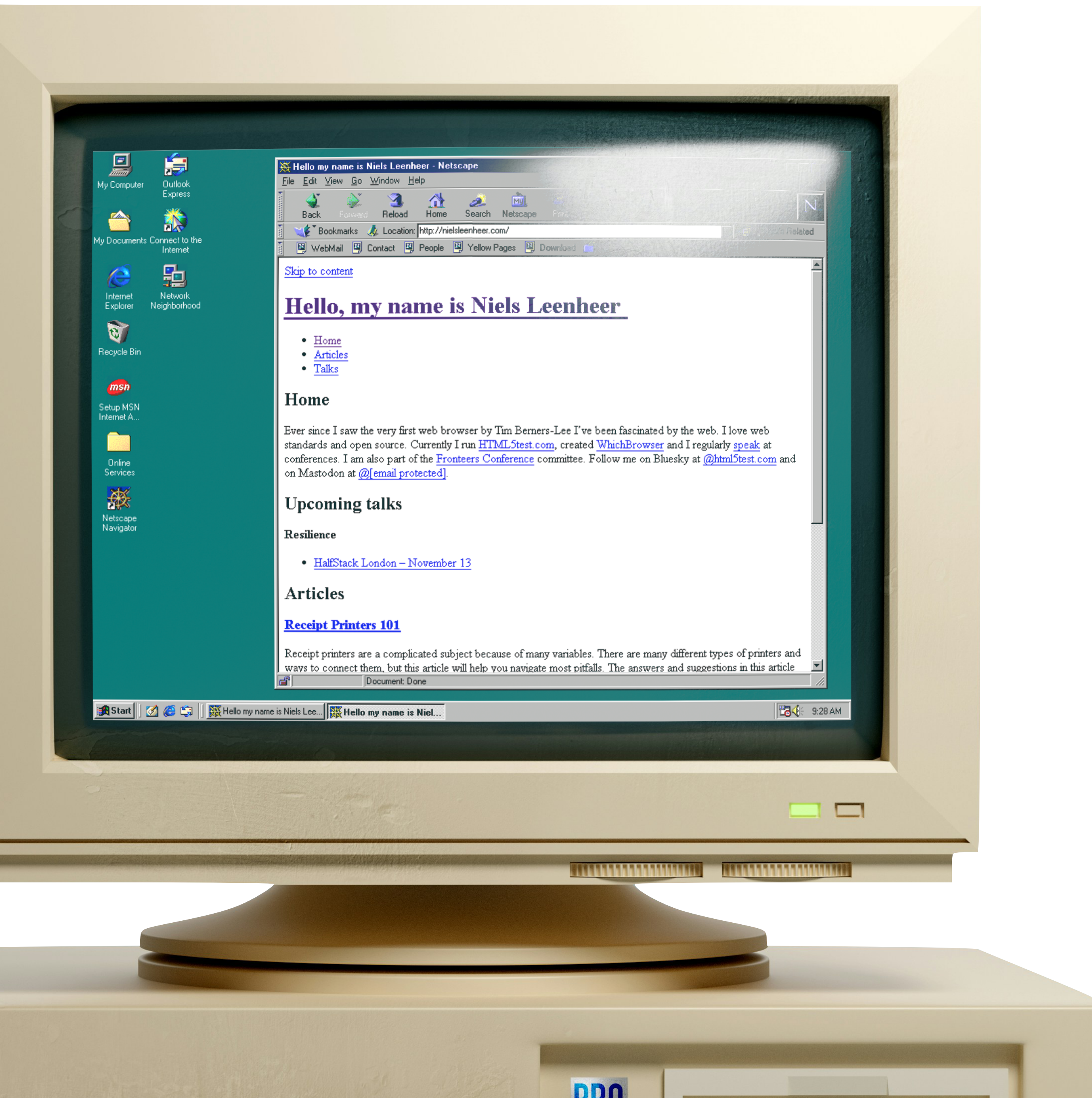
November 7, 2024 at 4:36 AM  Everybody can reply



THIS IS FOR EVERYONE

THIS IS FOR

THIS IS FOR



< / noscript >



<noscript>

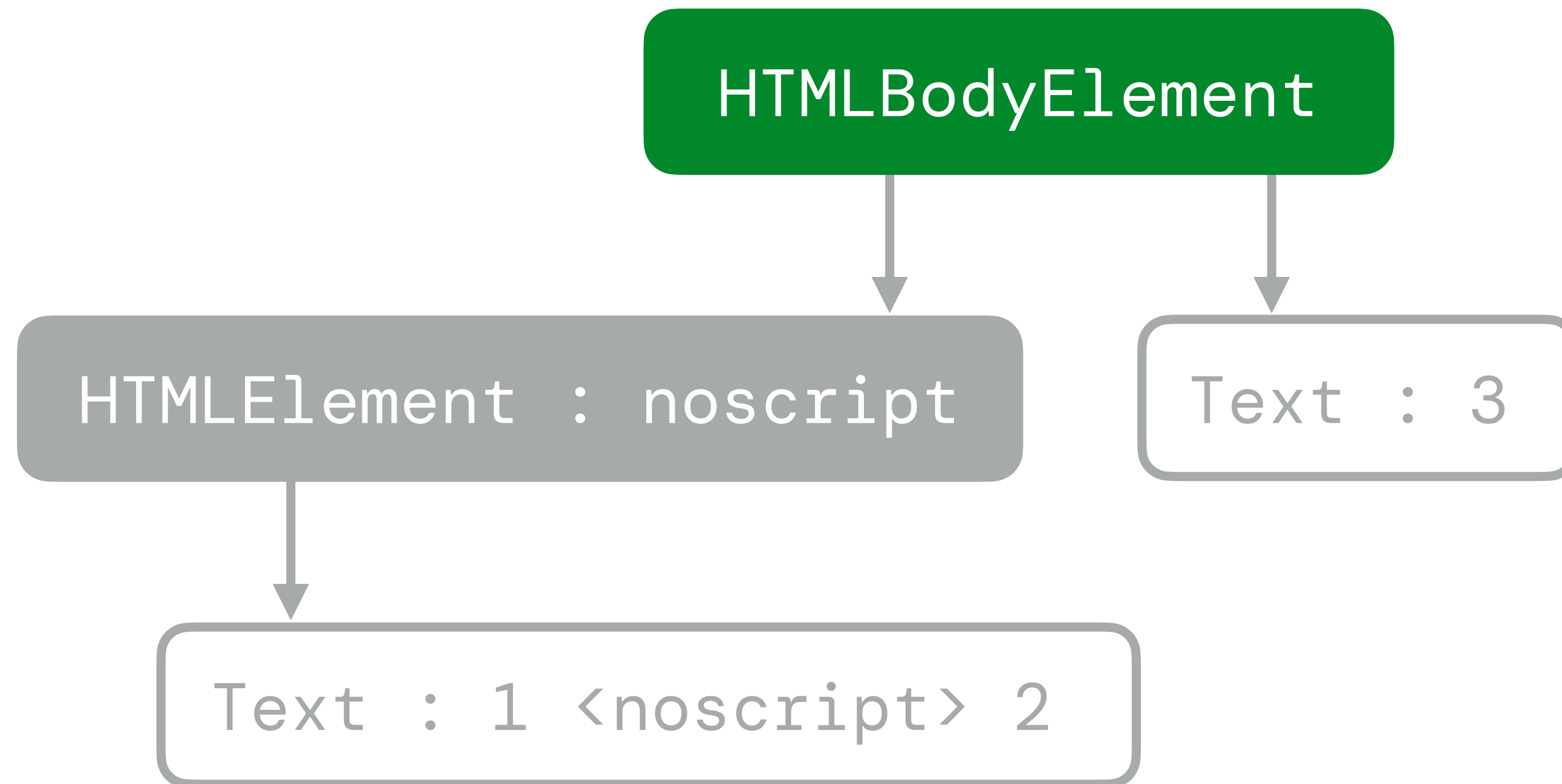
    <noscript>

        ...

    </noscript>

</noscript>

```
<noscript> 1 <noscript> 2 </noscript> 3 </noscript>
```

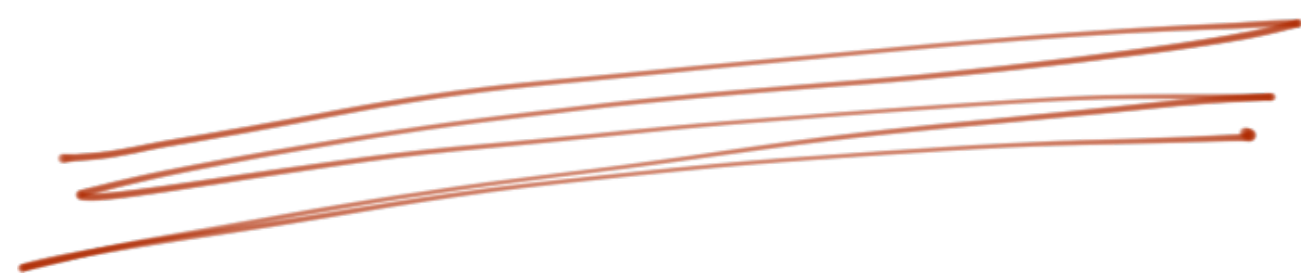


```
<noscript style='... '>
```

```
...
```

```
</noscript>
```

no



```
<noscript>  
  <script>  
    ...  
  </script>  
</noscript>
```

wtf?

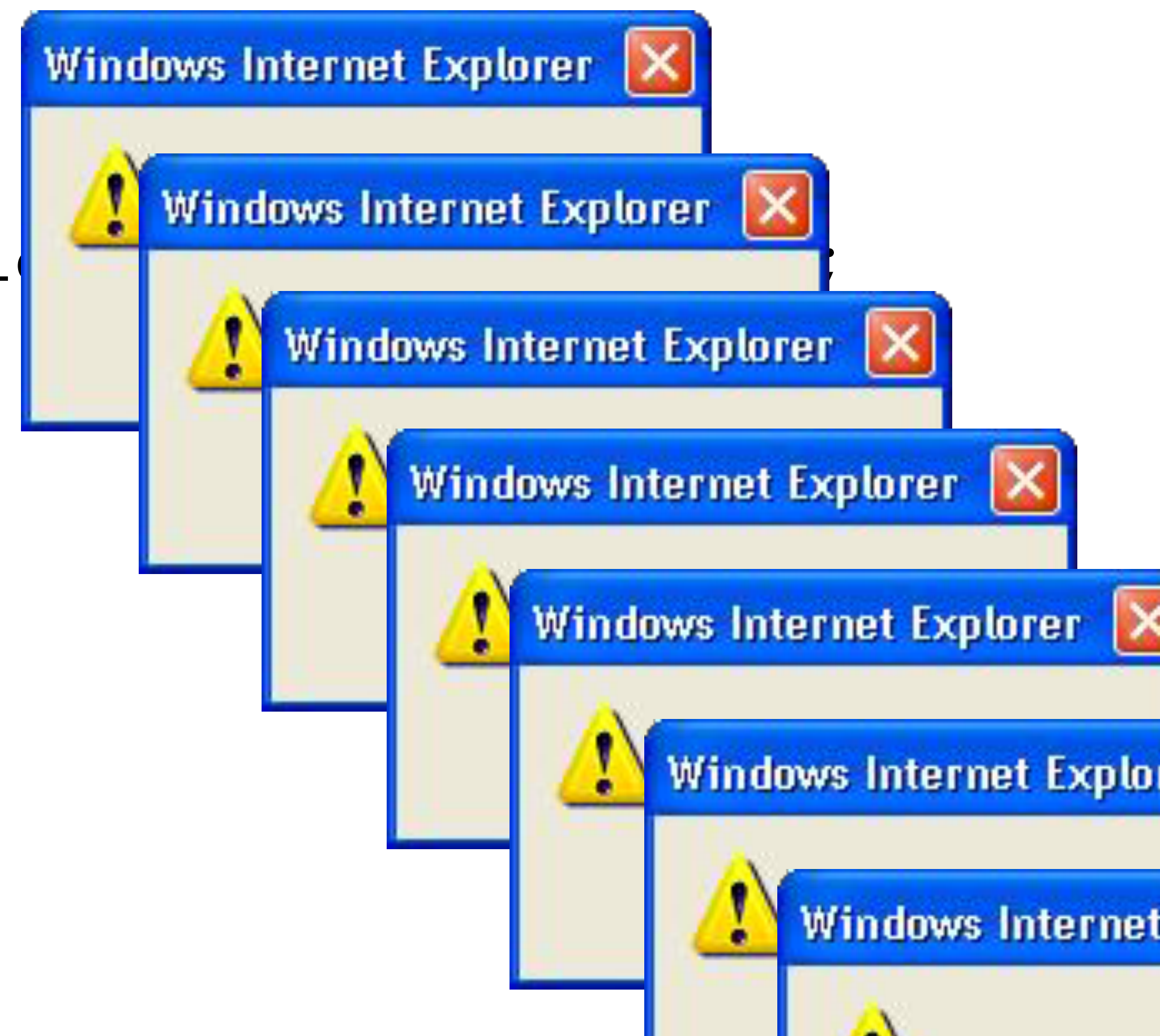
```
<noscript>
  <script>
    <!--
      let html='</noscript> sucks, long live <script><!--';
      alert('Hello World');
    //-->
  </script>
</noscript>
```



<no

<!

```
ipt> sucks, 10  
    alert('Hello World');  
    //-->  
    </script>  
</noscript>
```





Thank you!

