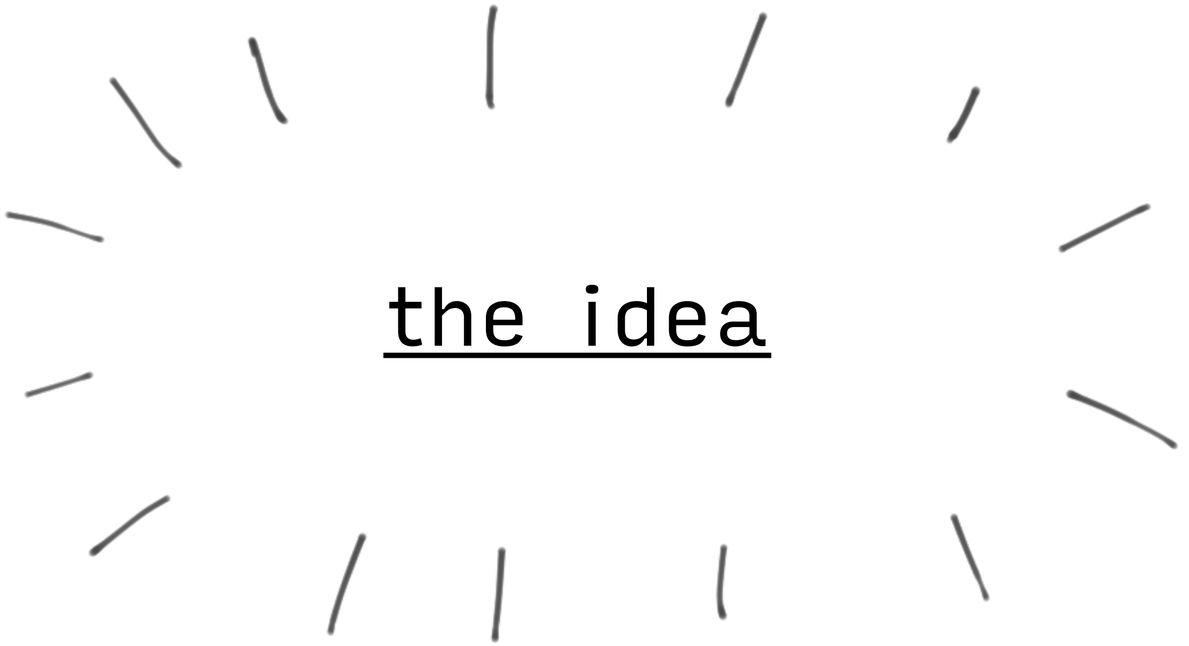
<noscript>
    The most useless tag of them all
</noscript>

CSS Day 2016 — June 16th

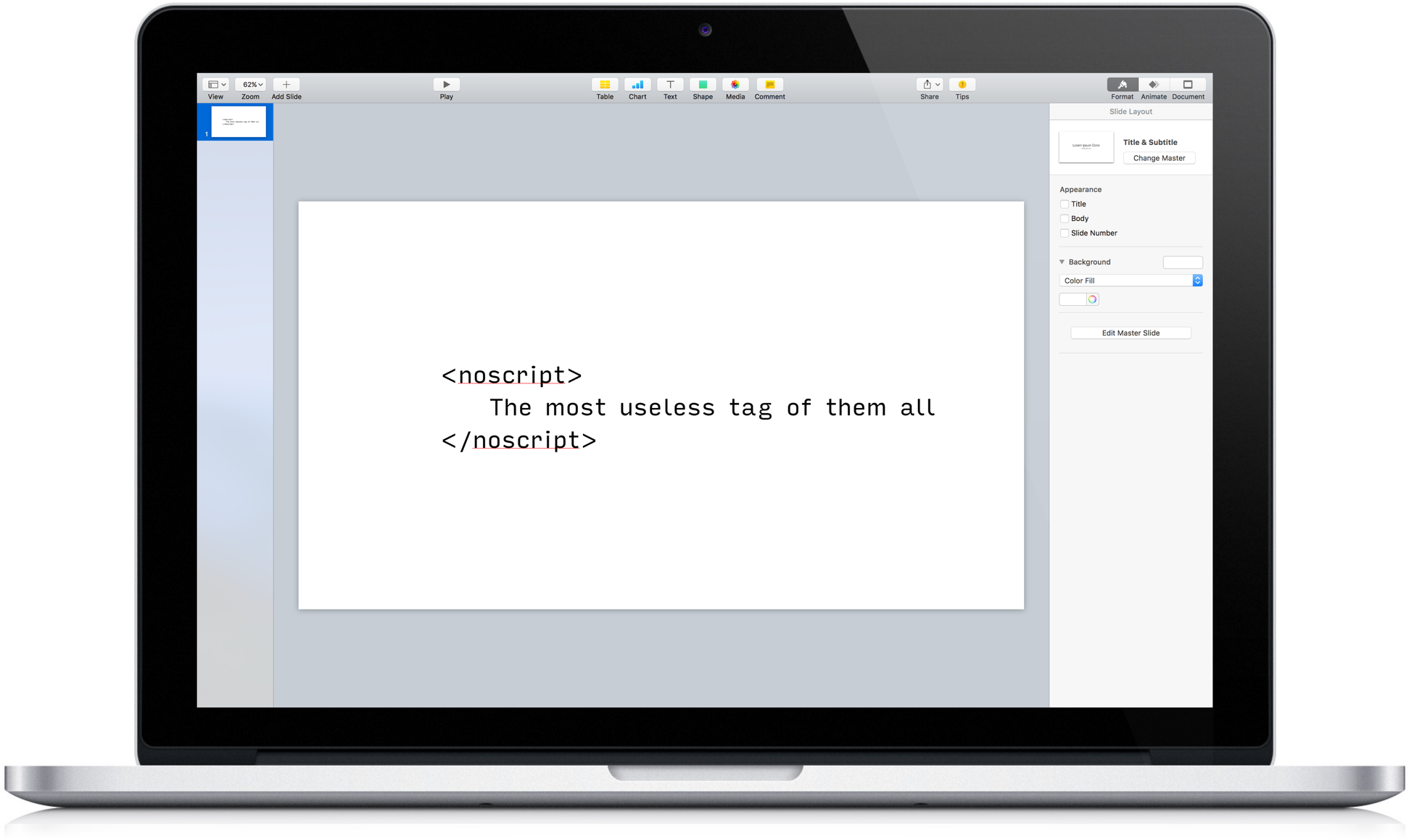
your name here:


the idea

<noscript>

```
<noscript>
    The most useless tag of them all
</noscript>
```

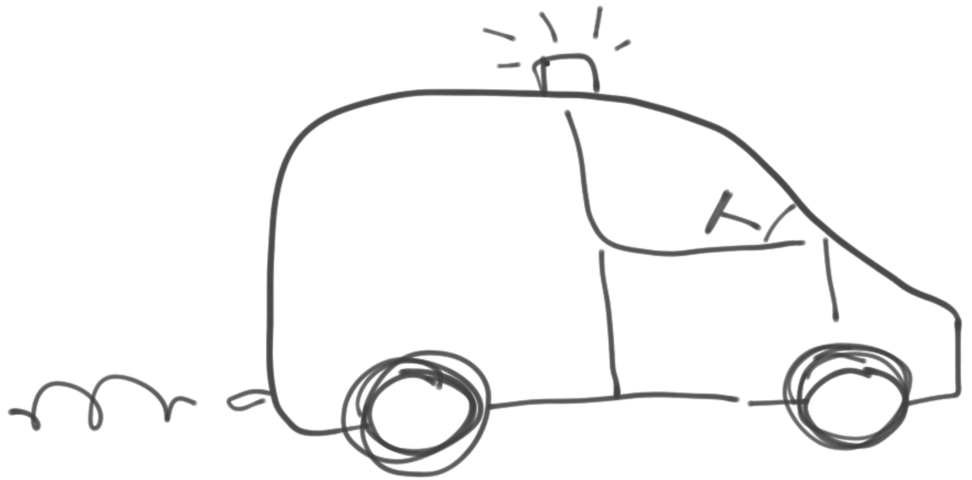
HTML is special. Unlike many other languages the browser won’t show an error message when you make a mistake. Sure, that makes it easy to write bad code, but it also allows HTML to be both backwards and future compatible. It allowed the HTML5 specification to extend the existing form field types. It allowed the RICG to create the <picture> element. And it forms the basis of Web Components because it makes custom elements possible. And most importantly, it allows the <noscript> tag, which by definition does absolutely nothing. This talk will explain the concepts behind graceful degradation, progressive enhancement and feature detection, and focus on how to solve practical problems with these concepts.


<noscript>
The most useless tag of them all
</noscript>

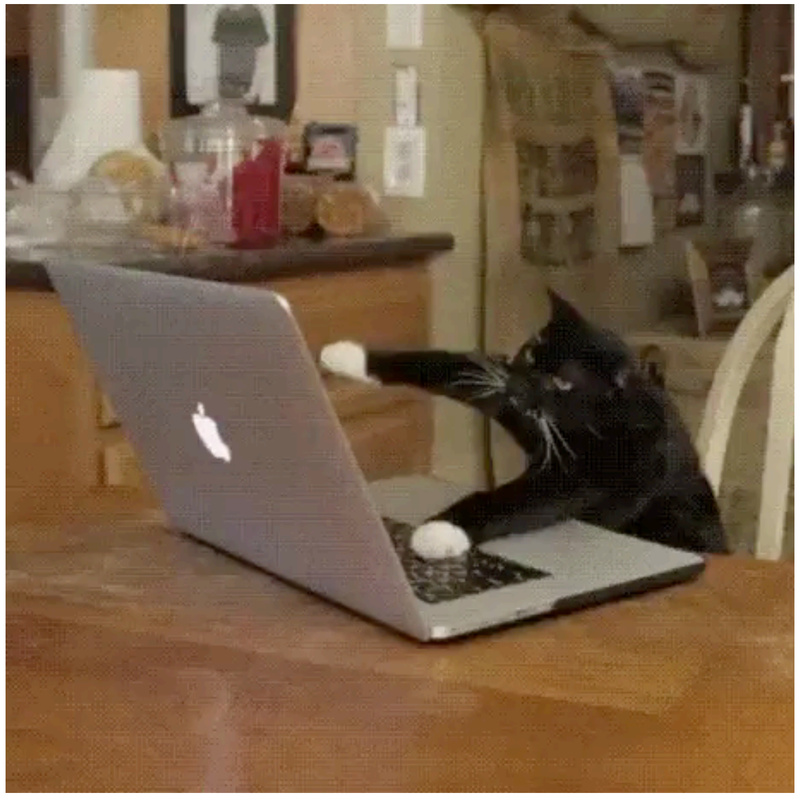
other talks

watching the
star wars trilogy

watching the
star wars trilogy
4 x

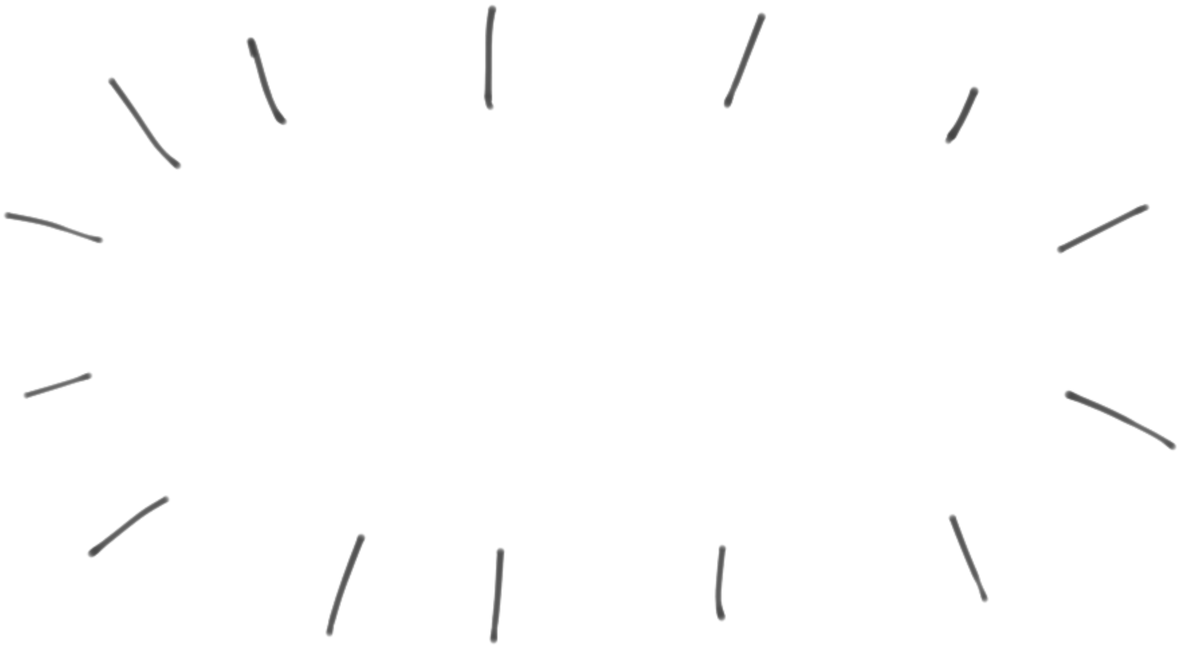


visit conferences



doing actual
work

procrastination



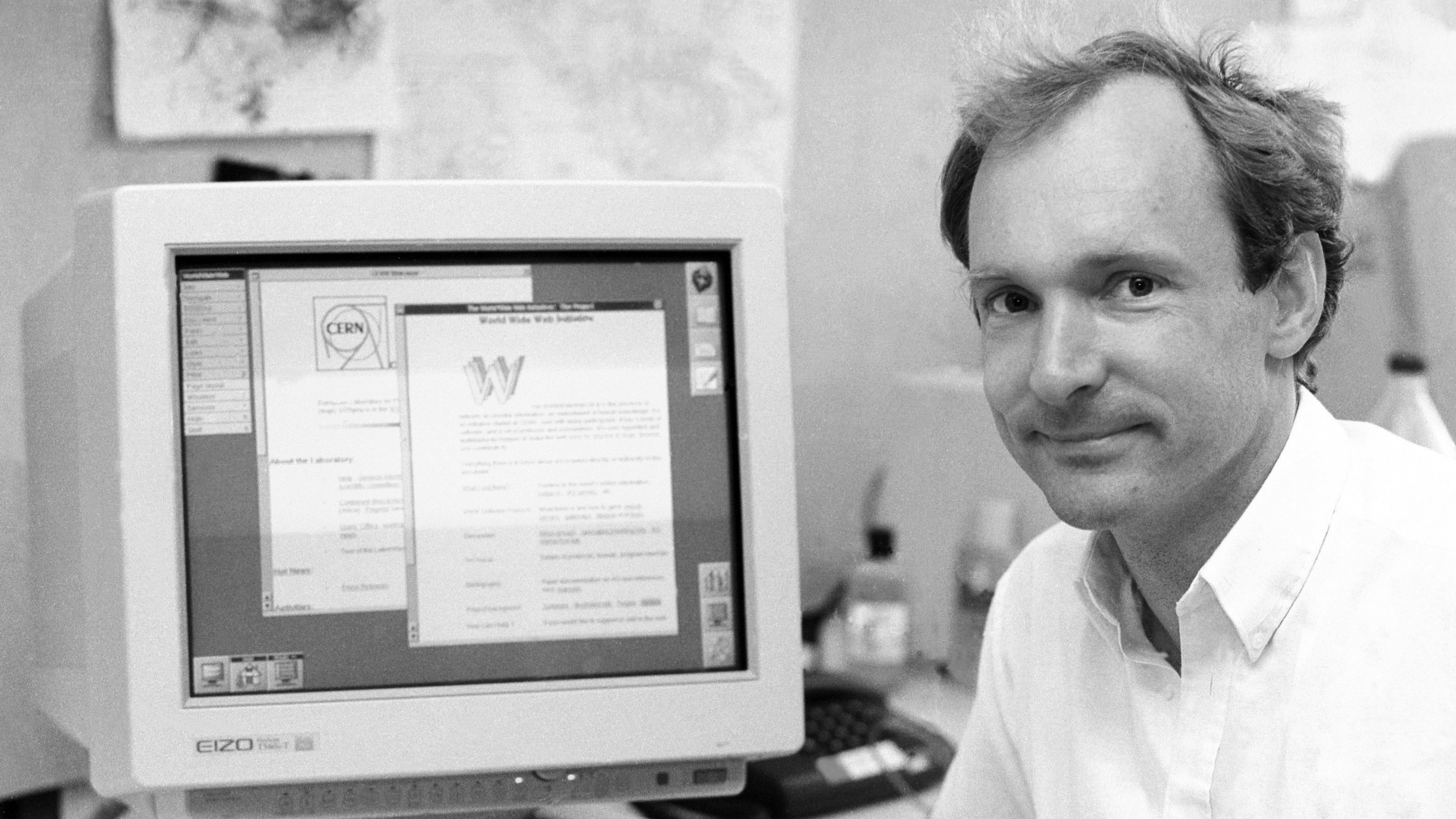
<noscript>
<noscript>
</noscript>
</noscript>

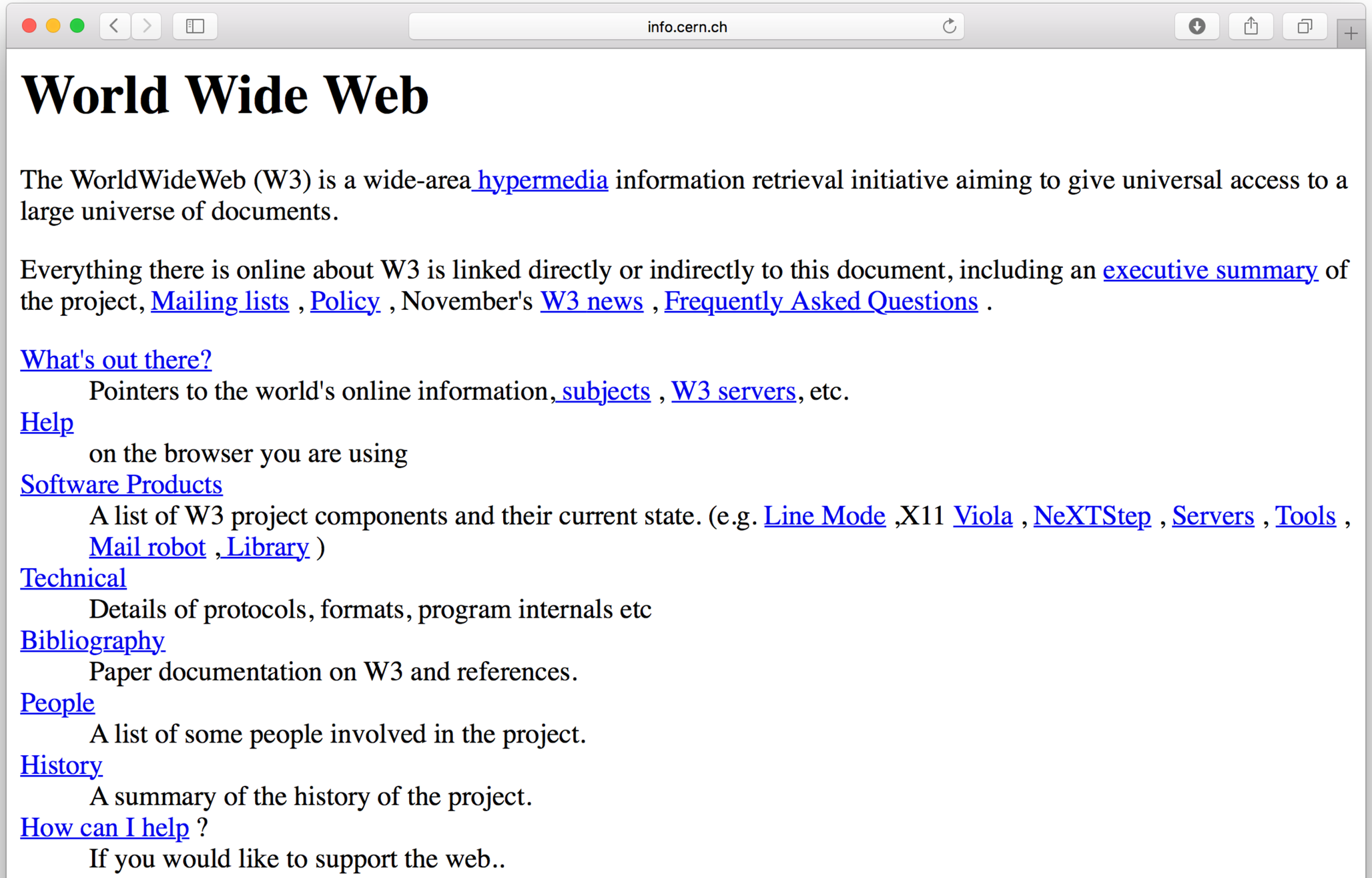
<noscript style="....">

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

resilience

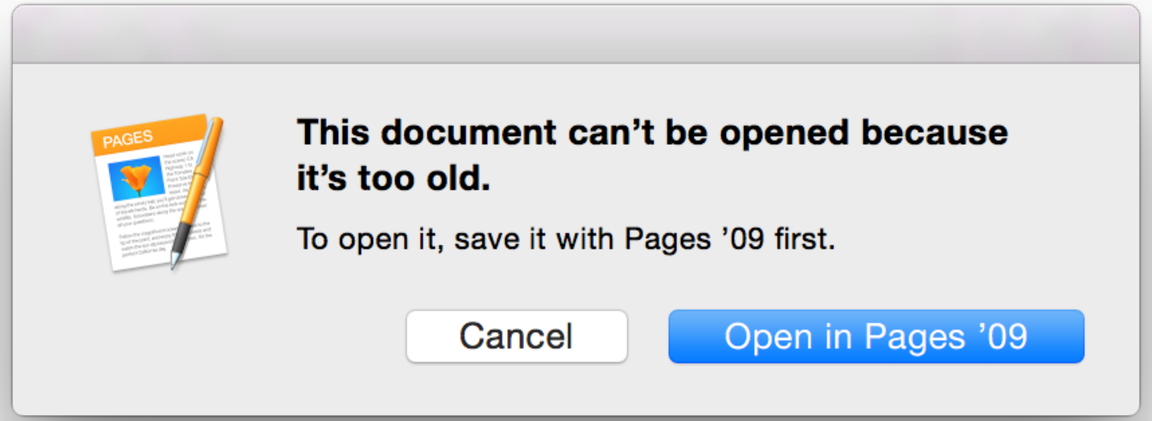
HTML is both backwards and future compatible


CERN
World Wide Web Initiative
EIZO


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..

```
<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
```

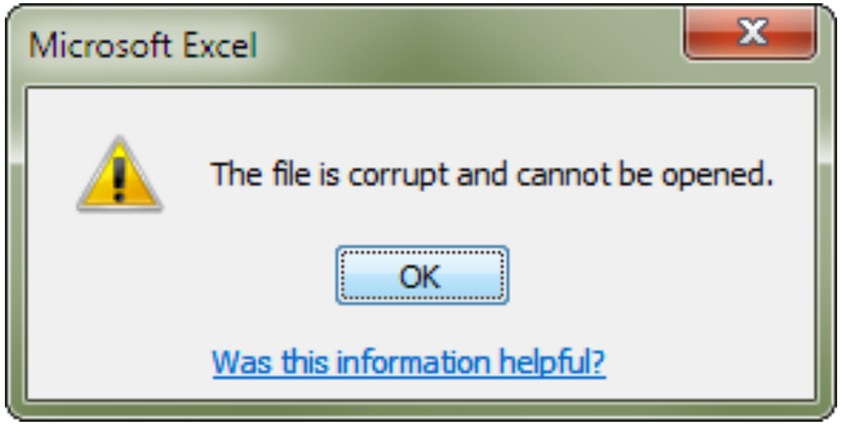
<nextid n="55">

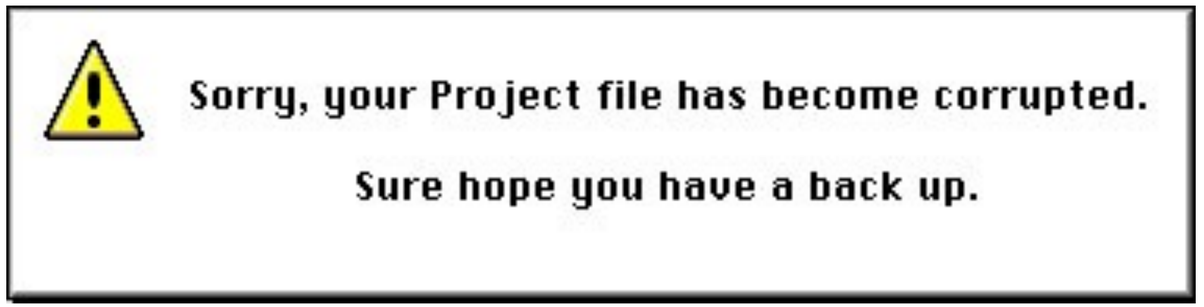

PAGES
This document can’t be opened because it’s too old.
To open it, save it with Pages ’09 first.
Cancel
Open in Pages ’09

<nextid n="55">

<nextid n="55">

```
<header>
   <title>…</title>
</header>
```


Microsoft Excel
The file is corrupt and cannot be opened.
OK
Was this information helpful?


Sorry, your Project file has become corrupted.
Sure hope you have a back up.

```
<header>
    <title>…</title>
</header>
```

```
<picture>
   <source srcset='cat.jpg, cat@2x.jpg 2x'>
   <img src='catastrophe.jpg'>
</picture>
```

```
<picture>
  <source srcset='cat.jpg, cat@2x.jpg 2x'>
  <img src='catastrophe.jpg'>
</picture>
```

```
<picture>
  <source srcset='cat.jpg, cat@2x.jpg 2x'>
  <img src='catastrophe.jpg'>
</picture>
```

```
<noscript>
   What is this, 1995? Take a chill pill!
   Enable your JavaScript! Booyah!
</noscript>
```

<noscript>
What is this, 1995? Take a chill pill!
Enable your JavaScript! Booyah!
</noscript>

```
<script>
    confirm(
      'You will lose unsaved changes ' +
      'if you cancel, is this OK?'
    );
</script>
```

```
<script>
    confirm(
      'You will lose unsaved changes ' +
      'if you cancel, is this OK?'
    );
</script>
```

```
<script>
<!--
    confirm(
      'You will lose unsaved changes ' +
      'if you cancel, is this OK?'
    );
//-->
</script>
```

```
<script>
<!--
    confirm(
      'You will lose unsaved changes ' +
      'if you cancel, is this OK?'
    );
//-->
</script>
```

create polyfills for our favourite tags

<blink>

```
<blink>
    my eyes! please make it stop!
</blink>
```

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

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

<marquee>

<marquee>

supported by all browsers

HTML

DOM

HTML

tokenizer

&

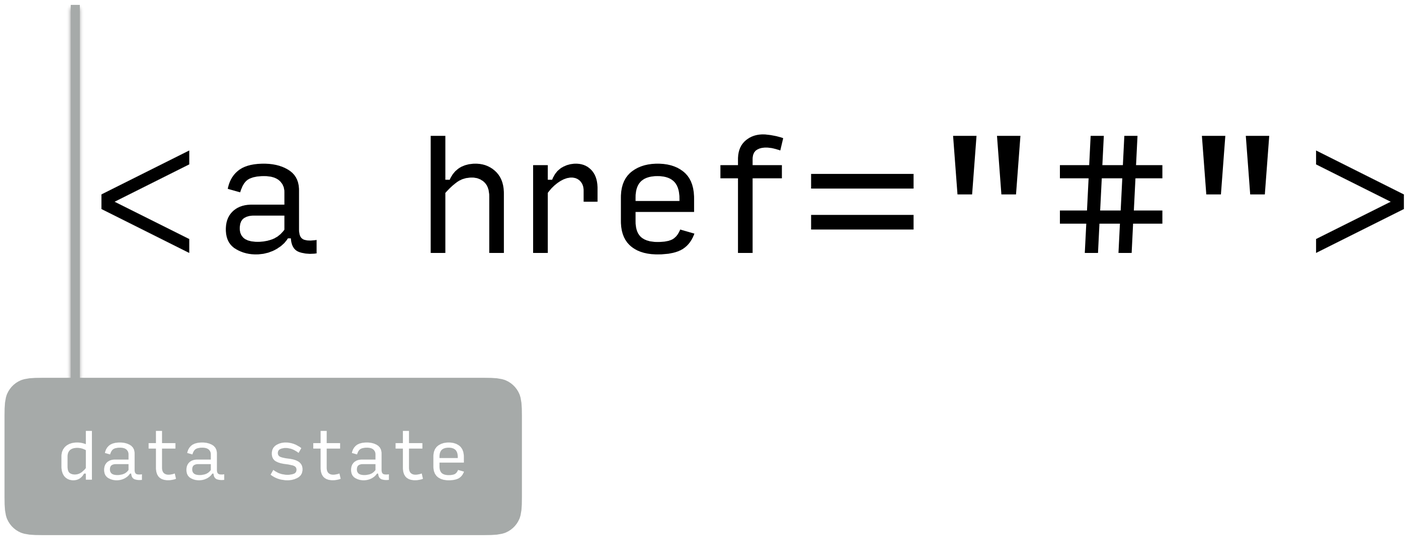
tree builder

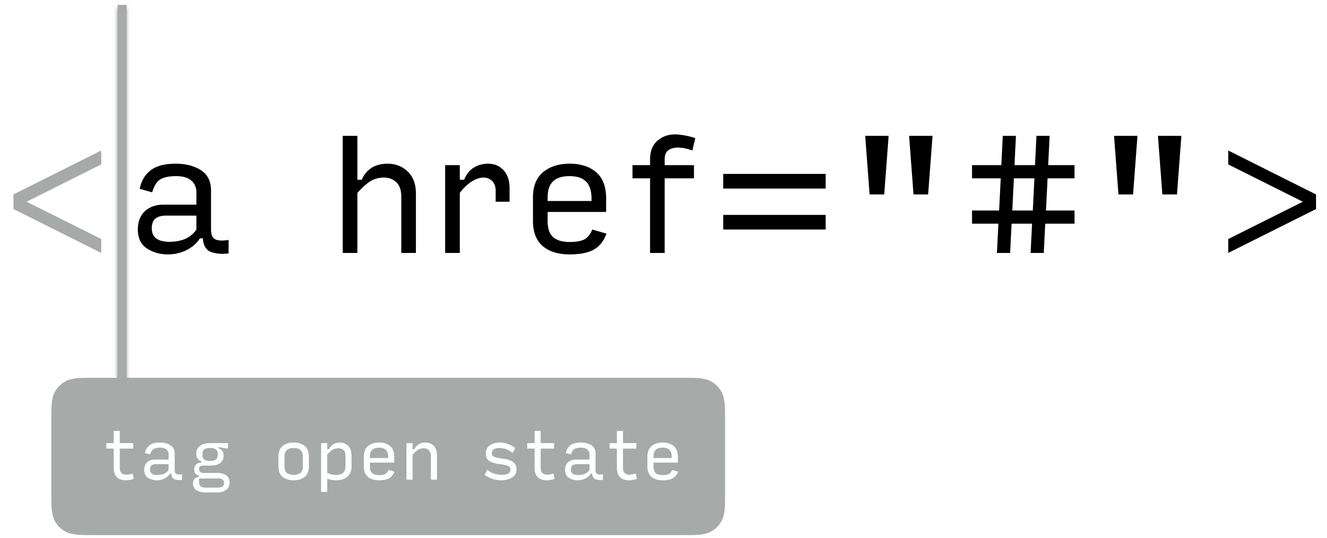
DOM


tokenizer

analyse html character by character

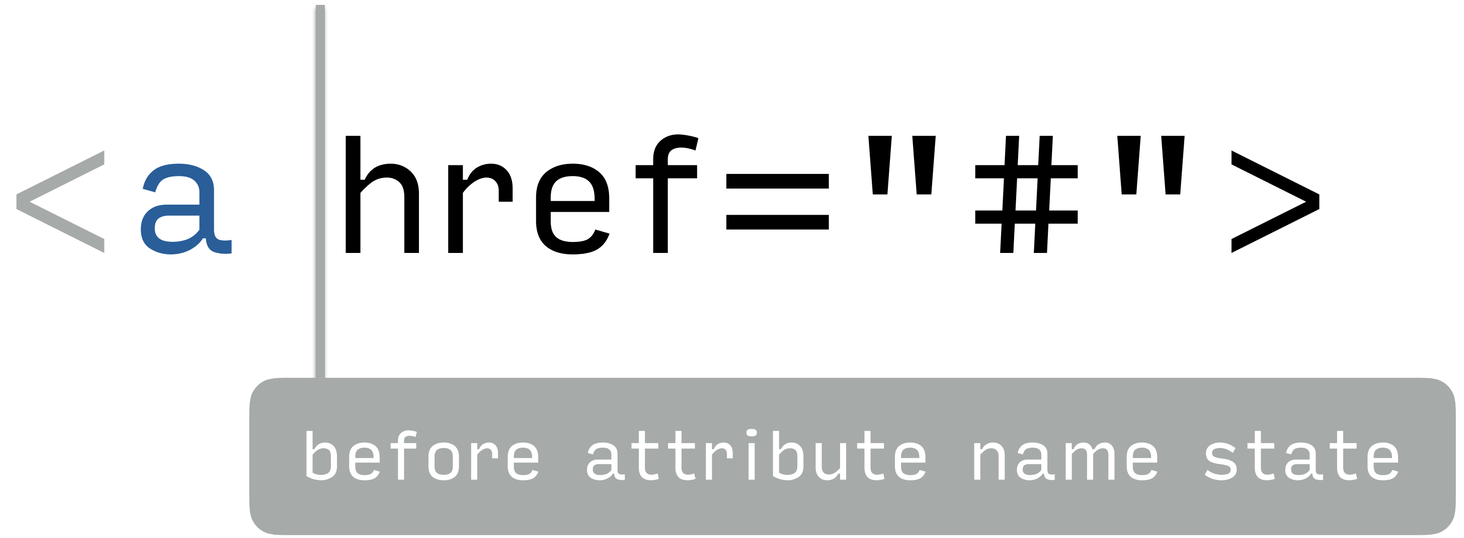
<a href="#">

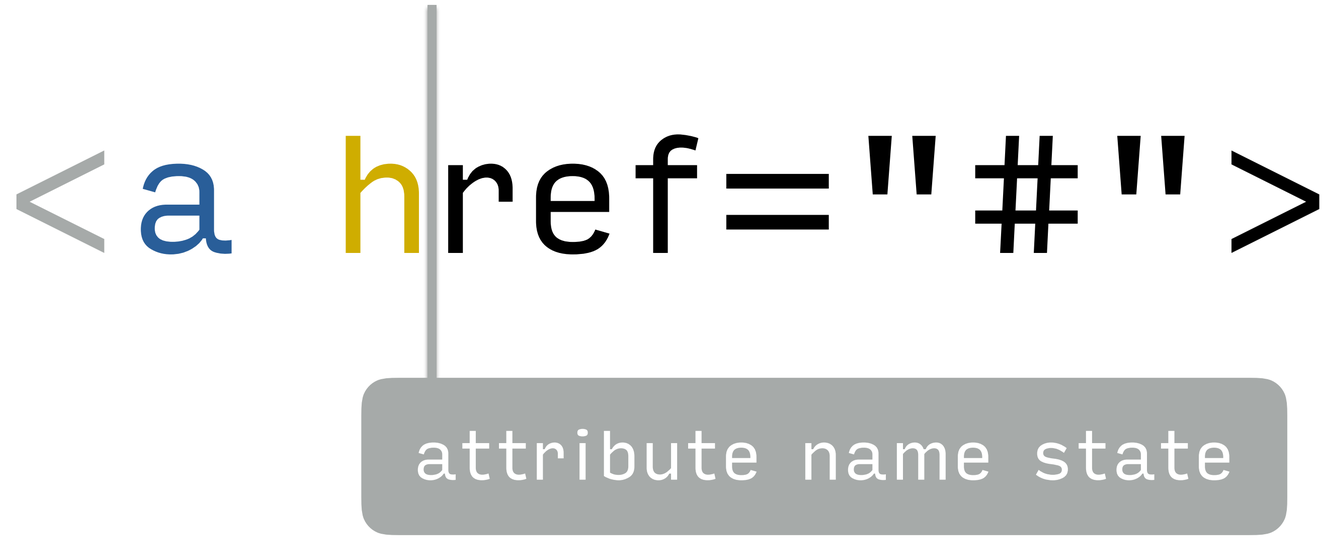

<a href="#">
data state

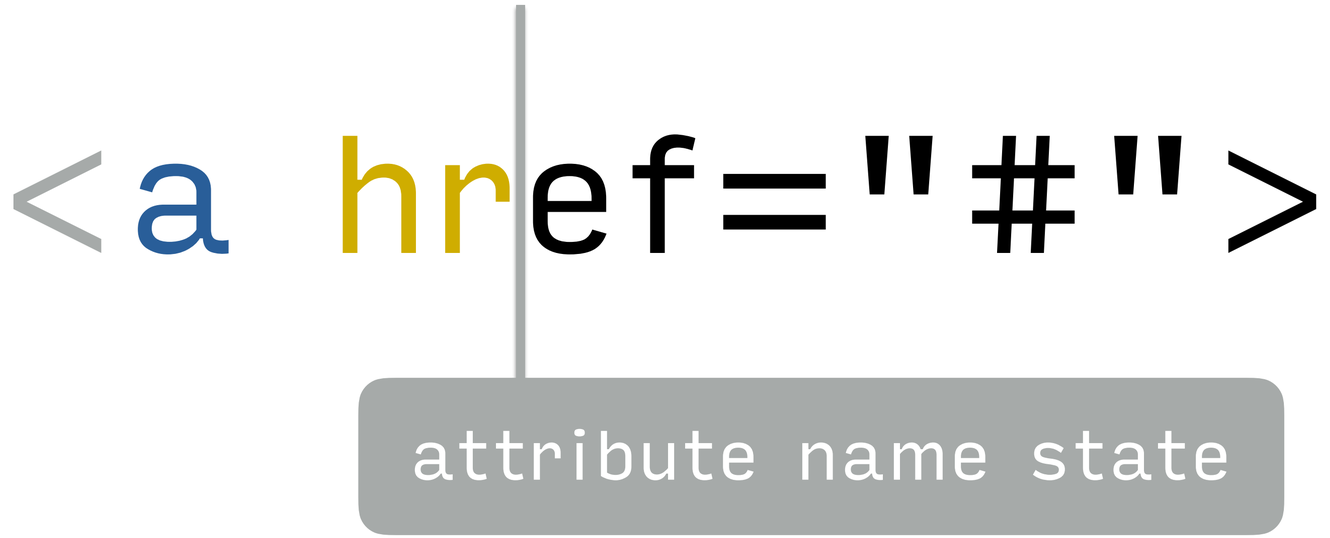

<a href="#">
tag open state

<a href="#">

tag name state


<a href="#">
before attribute name state

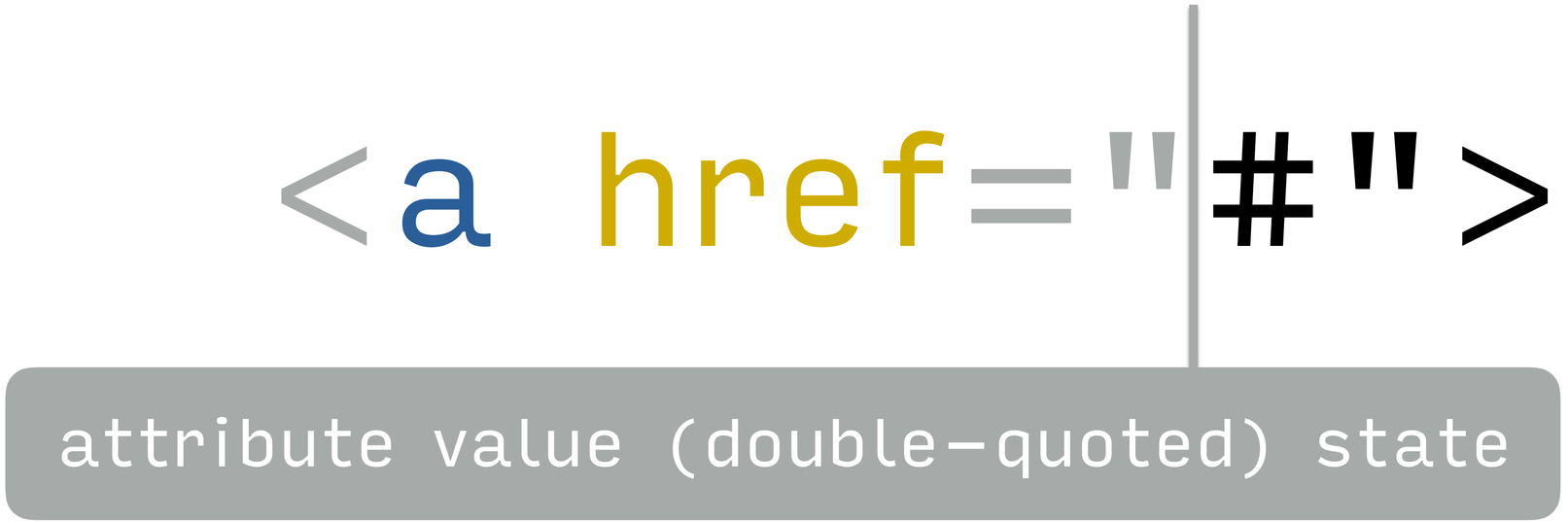

<a href="#">
attribute name state

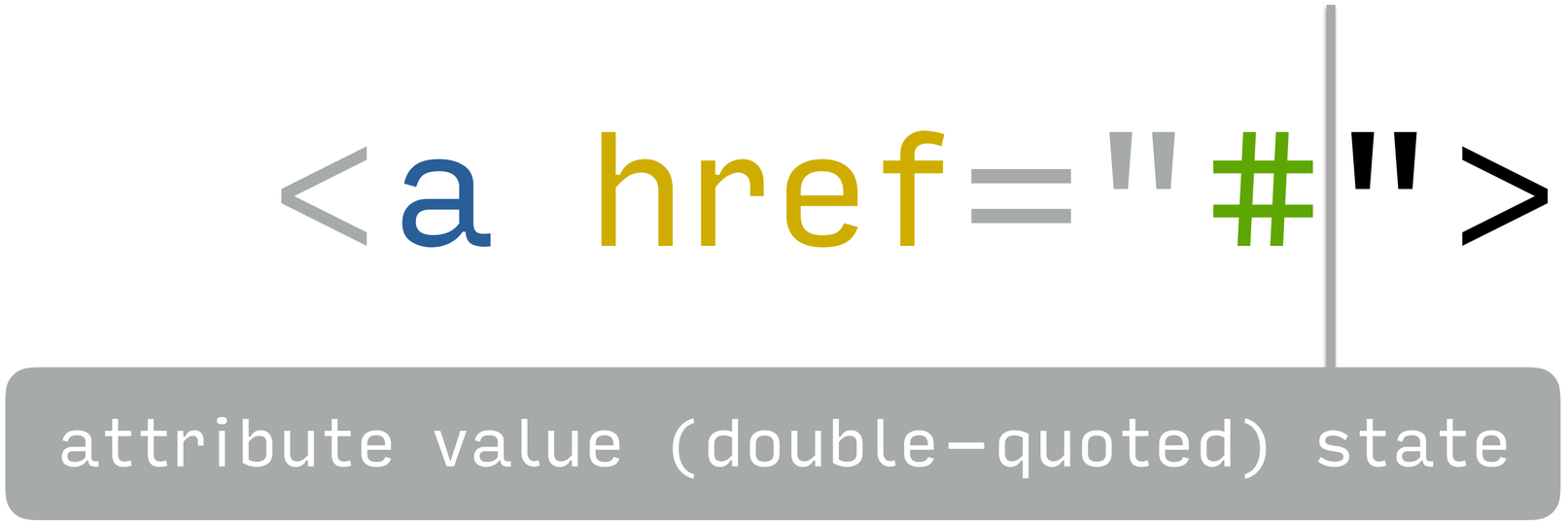

<a href="#">
attribute name state


<a href="#">
attribute name state


<a href="#">
attribute name state

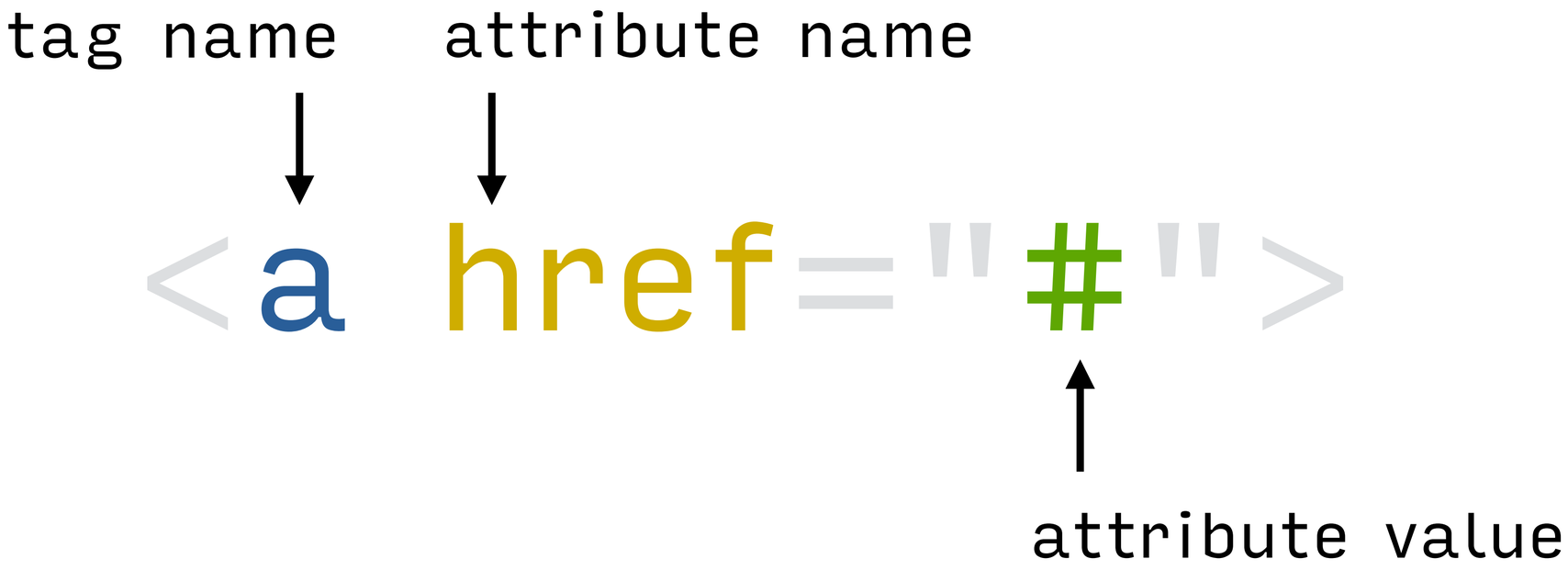

<a href="#">
before attribute value state

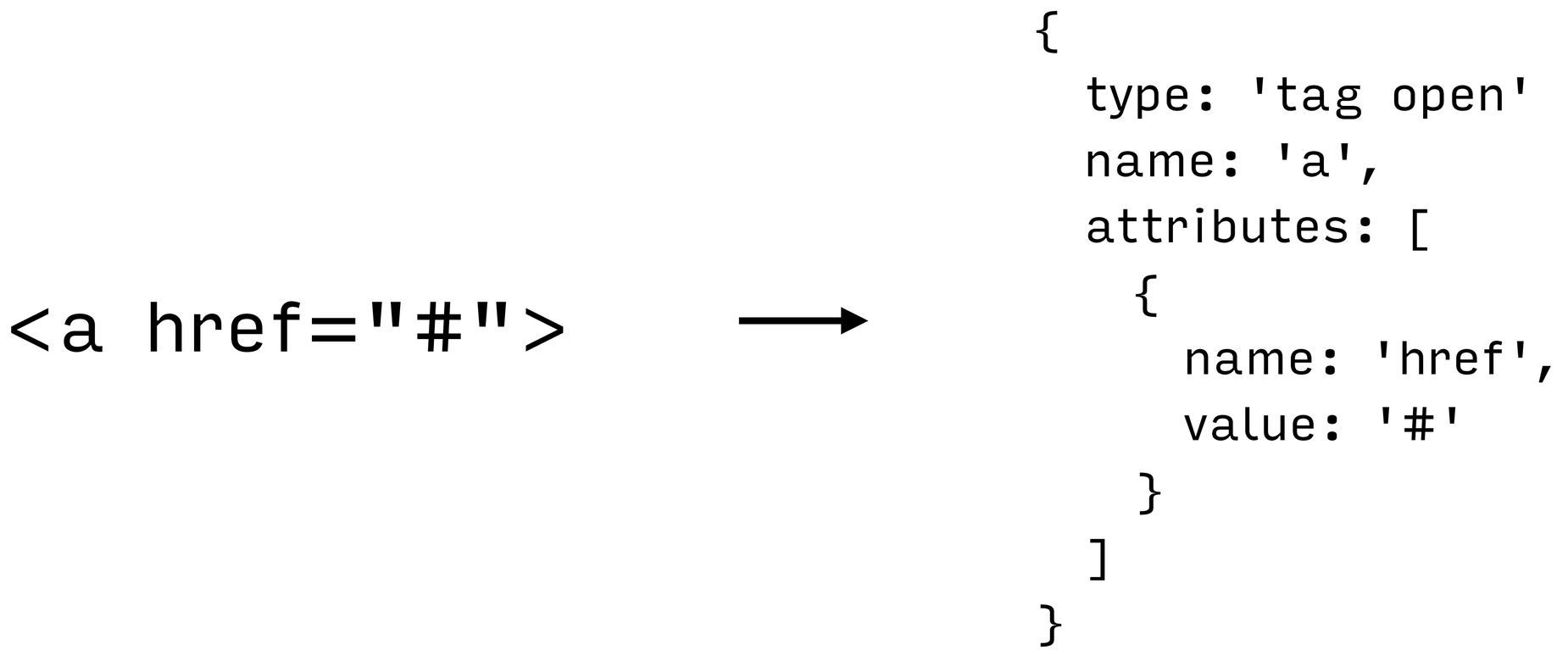

<a href="#">
attribute value (double-quoted) state


<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.

<a<a<a<a<a>

this is a tag!

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

<uʍop-əpısdn>
    ʇɥıs ɪs ɐ ʇɐg
</uʍop-əpısdn>

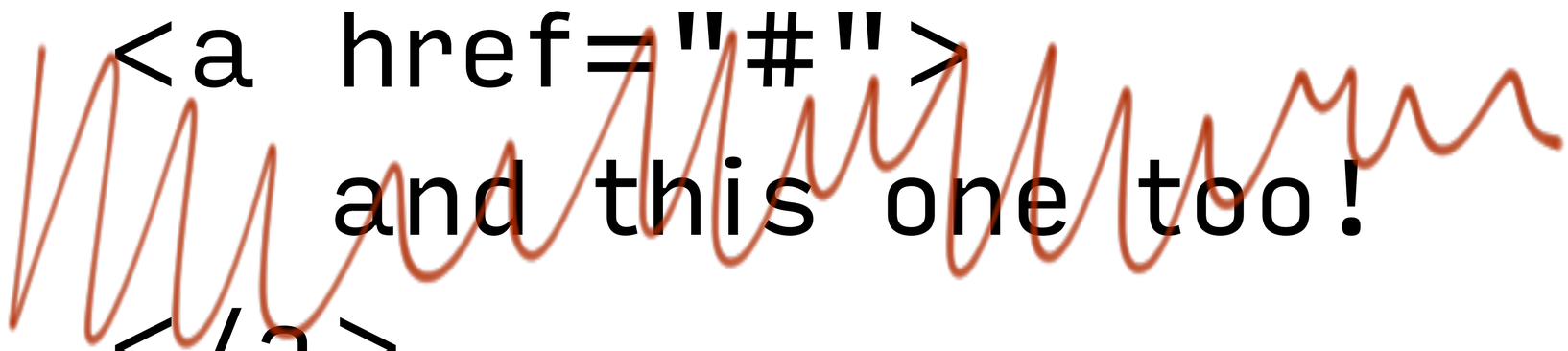

Nuclear Power Plant
Spider Farm

<c(ノ ºロº) ノ ⁀ ┻━━┻>

this really is a tag too!

</c(ノ ºロº) ノ ⁀ ┻━━┻>

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



and this one too!

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.

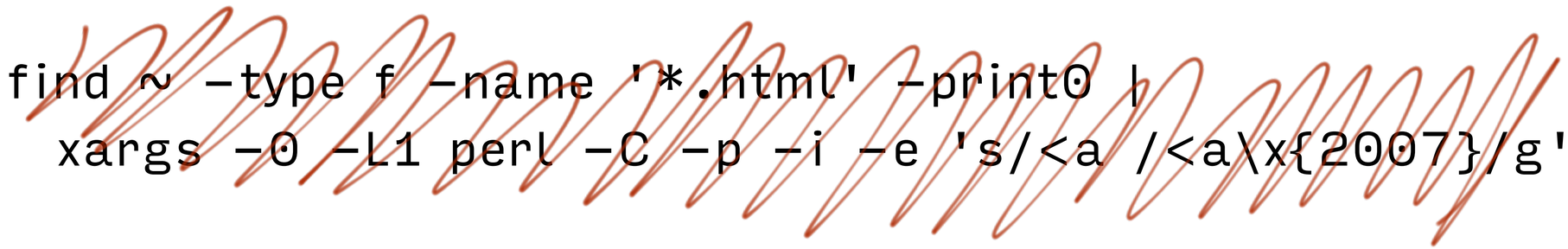
```
<a href="#">
    and this one too!
</a>
```


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


```
<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 /<a\x{2007}/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

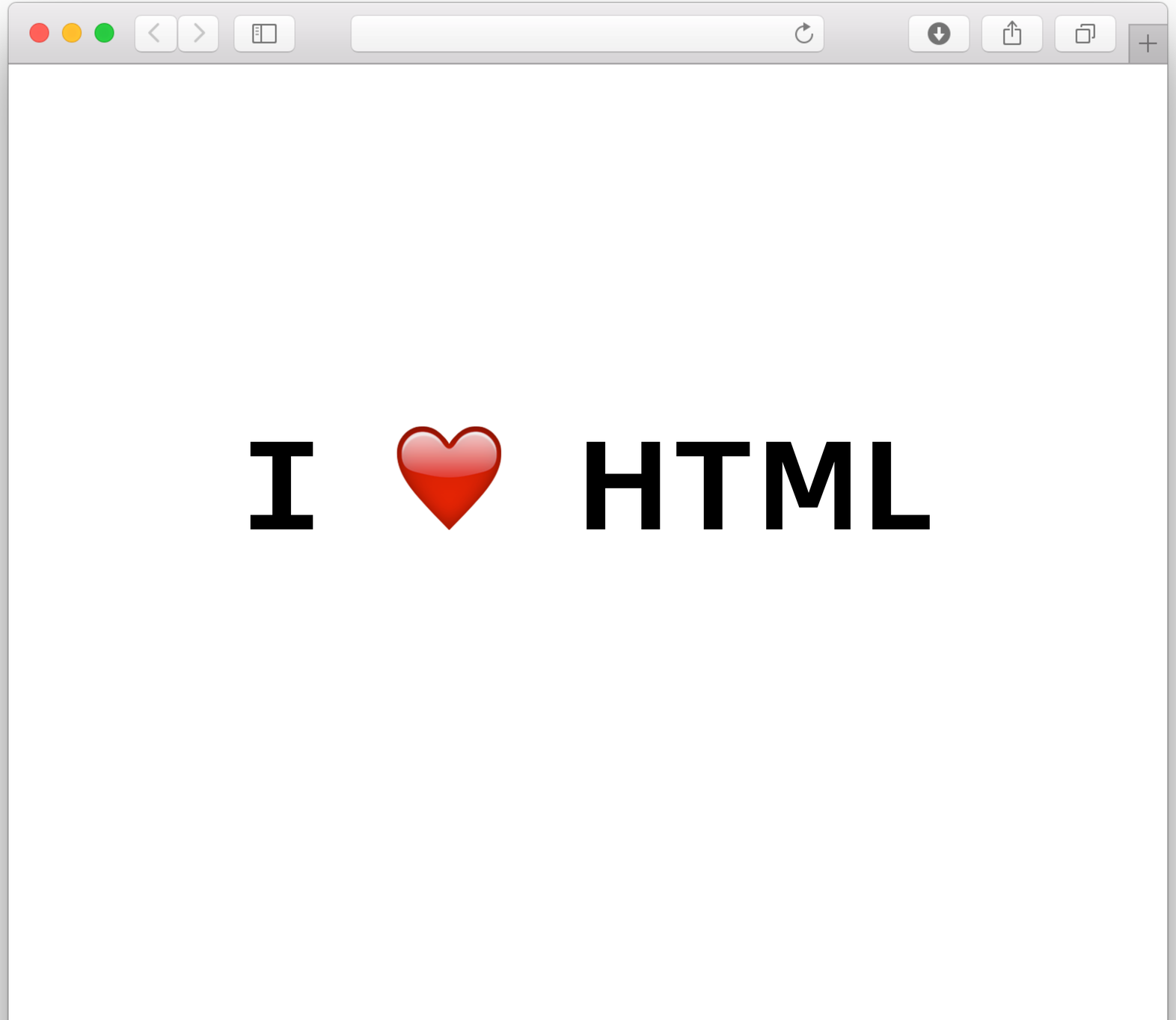
LOLOLOL

<i-?>unicode</i-?>

<i-❤️>unicode</i-❤️>

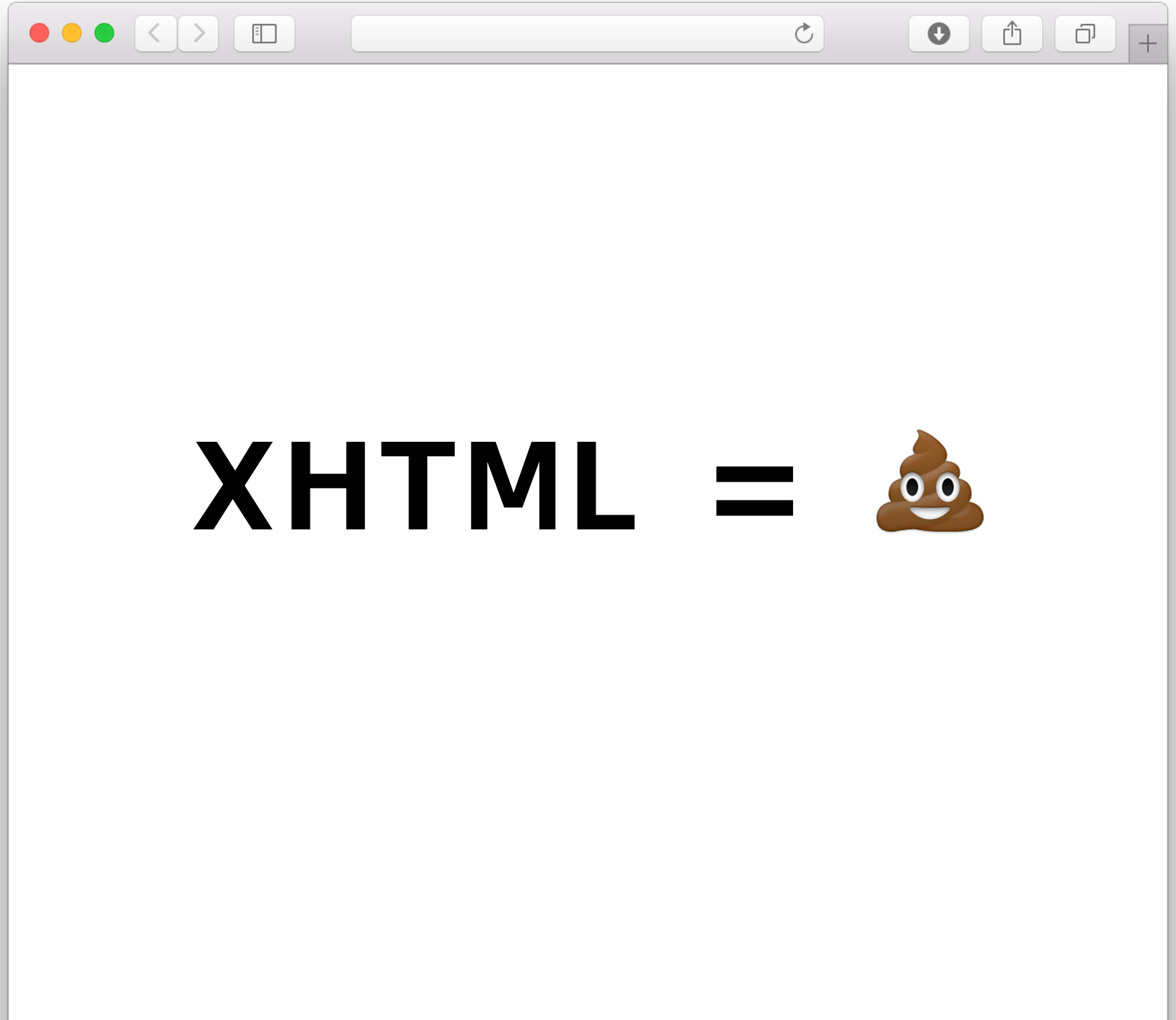
```
<i-❤️>
   HTML
</i-❤️>
```

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

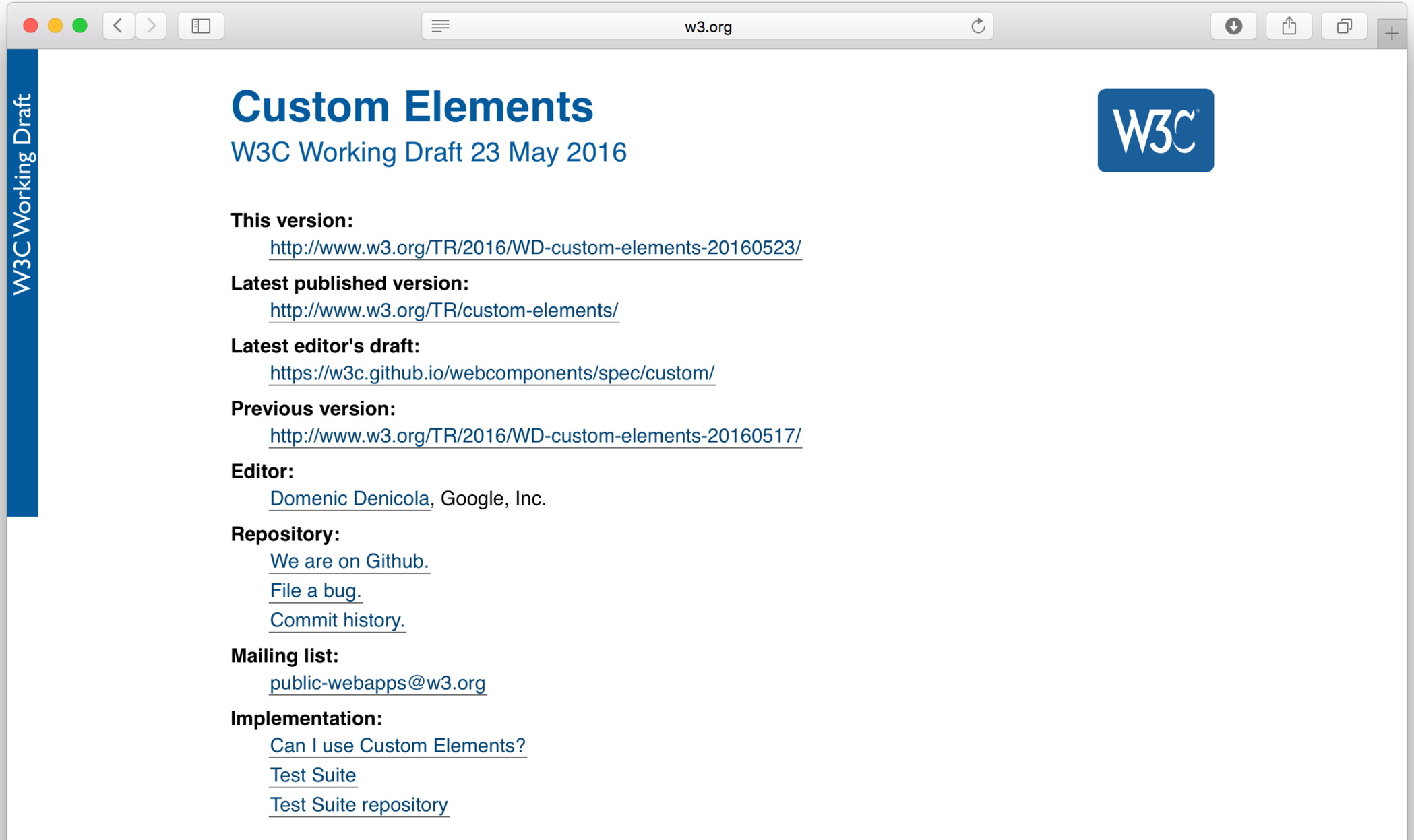


```
<is-💩>
   XHTML
</is-💩>
```

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



<dangerous>


w3.org
W3C Working Draft
Custom Elements
W3C Working Draft 23 May 2016
This version:
http://www.w3.org/TR/2016/WD-custom-elements-20160523/
Latest published version:
http://www.w3.org/TR/custom-elements/
Latest editor's draft:
https://w3c.github.io/webcomponents/spec/custom/
Previous version:
http://www.w3.org/TR/2016/WD-custom-elements-20160517/
Editor:
Domenic Denicola, Google, Inc.
Repository:
We are on Github.
File a bug.
Commit history.
Mailing list:
public-webapps@w3.org
Implementation:
Can I use Custom Elements?
Test Suite
Test Suite repository

custom elements

custom dom interface for your own element

receive events when an element is
created, attached or detached,
and when attributes are changed

A ***valid custom element name*** is a sequence of characters name that meets all of the following requirements:

- name must match the `PotentialCustomElementName` production:

  ```
  PotentialCustomElementName ::=
      [a-z] (PCENChar)* '-' (PCENChar)*

  PCENChar ::=
      "-" | "." | [0-9] | "_" | [a-z] | #xB7 | [#xC0-#xD6] | [#xD8-#xF6] | [#xF8-#x37D] |
      [#x37F-#x1FFF] | [#x200C-#x200D] | [#x203F-#x2040] | [#x2070-#x218F] | [#x2C00-#x2FEF]
      | [#x3001-#xD7FF] | [#xF900-#xFDCF] | [#xFDF0-#xFFFD] | [#x10000-#xEFFFF]
  ```

  This uses the EBNF notation from the *XML* specification. [XML]

- name must not be any of the following:

  - `annotation-xml`
  - `color-profile`
  - `font-face`
  - `font-face-src`
  - `font-face-uri`
  - `font-face-format`
  - `font-face-name`
  - `missing-glyph`

<dangerous>

<not allowed>

<completely-safe>

<this-is-👍>

```
document.createElement('image').tagName
```

A

IMAGE

B

Something else

```
document.createElement('image').tagName
```

A

IMAGE

B

Something else

```
d = document.createElement('div');
d.innerHTML = '<image>';
d.firstChild.tagName
```

A

IMAGE

B

Something else

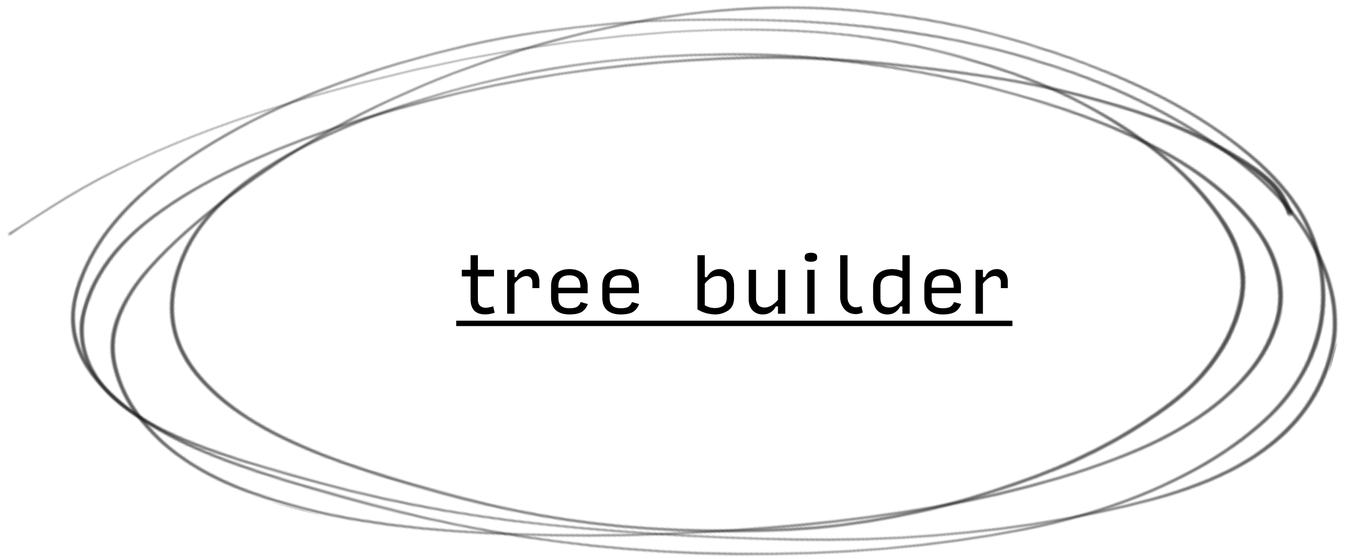
```
d = document.createElement('div');
d.innerHTML = '<image>';
d.firstChild.tagName
```

A

B

IMAGE

Something else

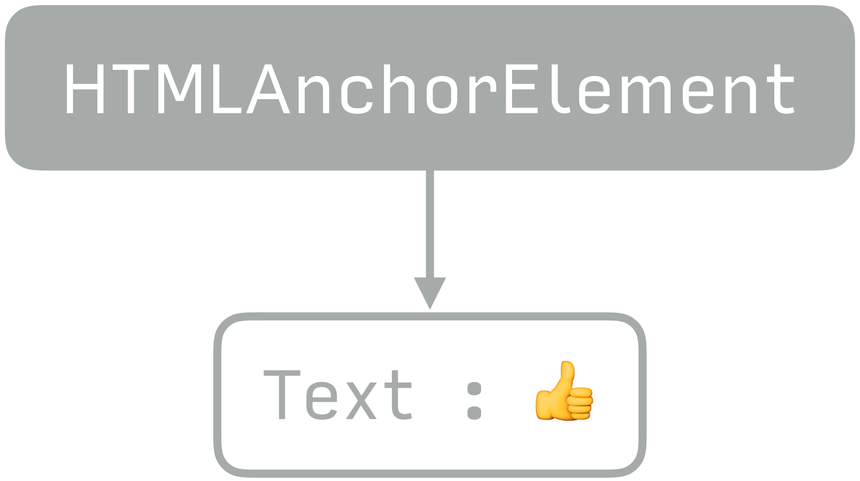

tree builder

👍

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

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

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


HTMLAnchorElement
Text : 👍

creating missing elements

 👍

initial insertion mode

 👍

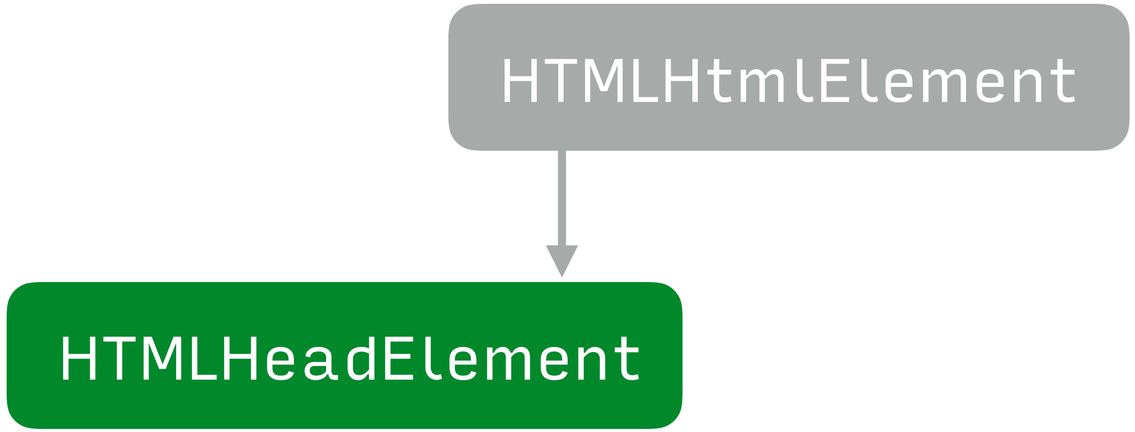
before html insertion mode

 👍

HTMLHtmlElement

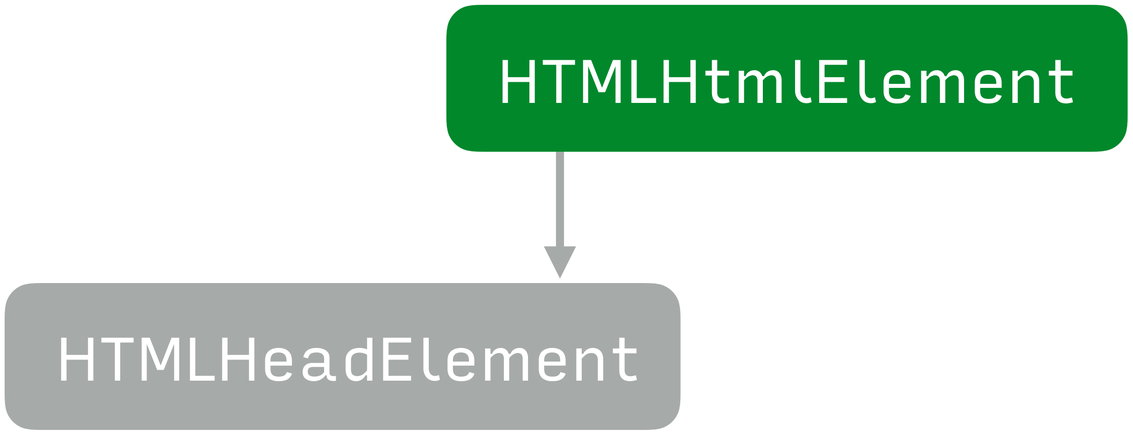
before head insertion mode

 👍



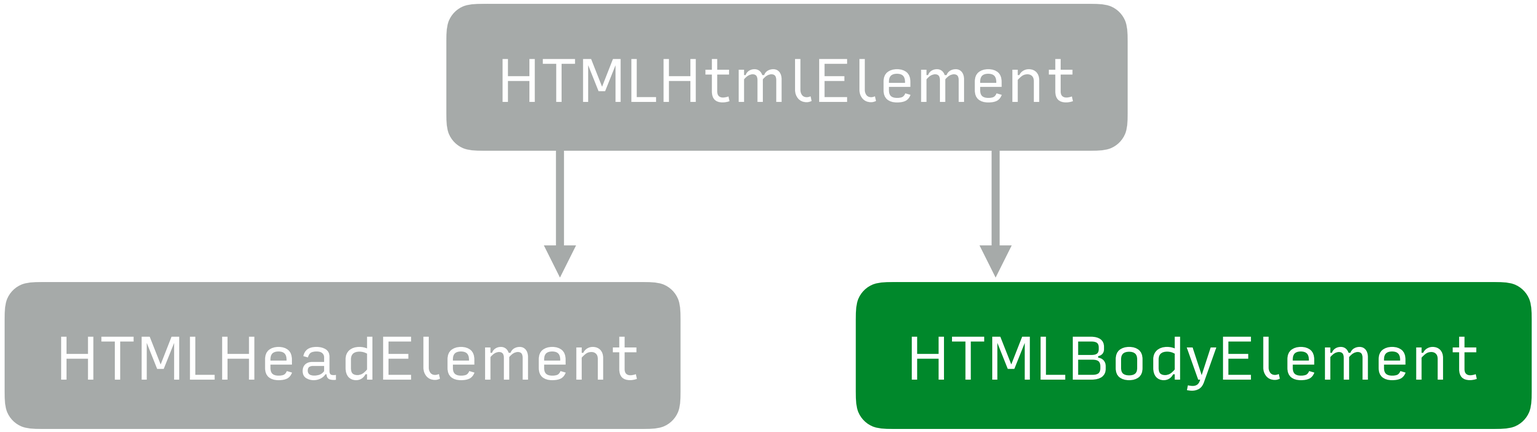
in head insertion mode

 👍

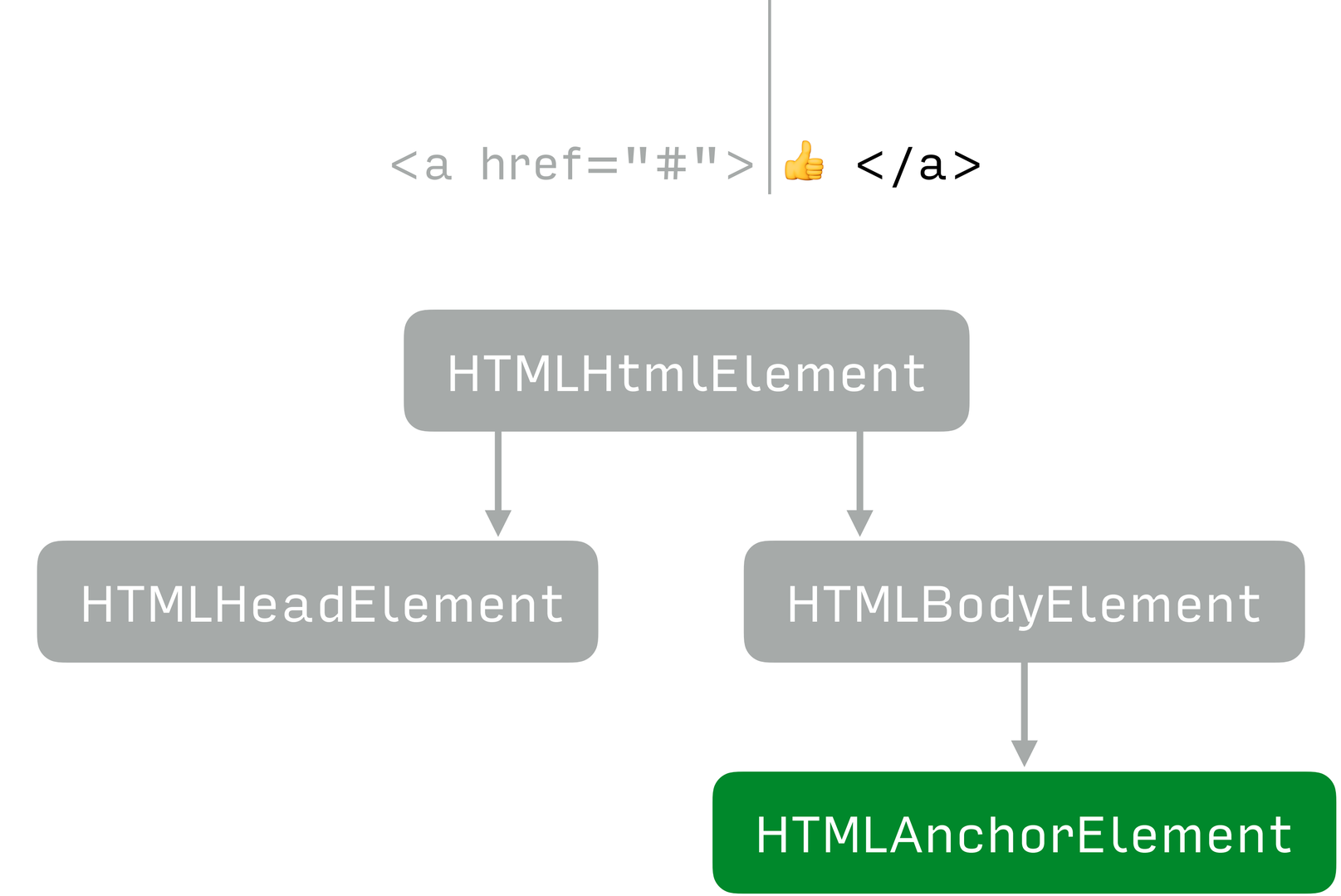


after head insertion mode

 👍

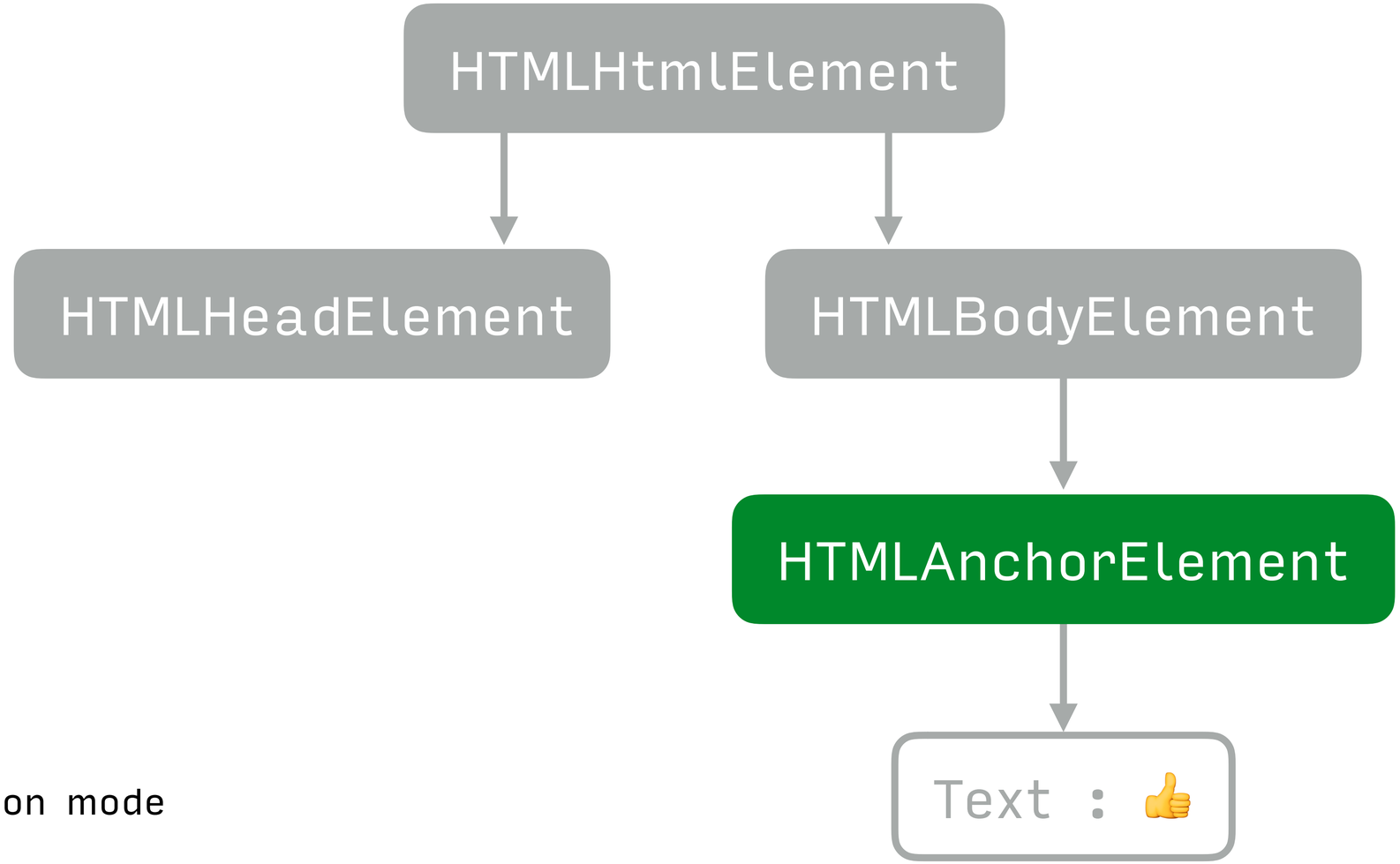


in body insertion mode



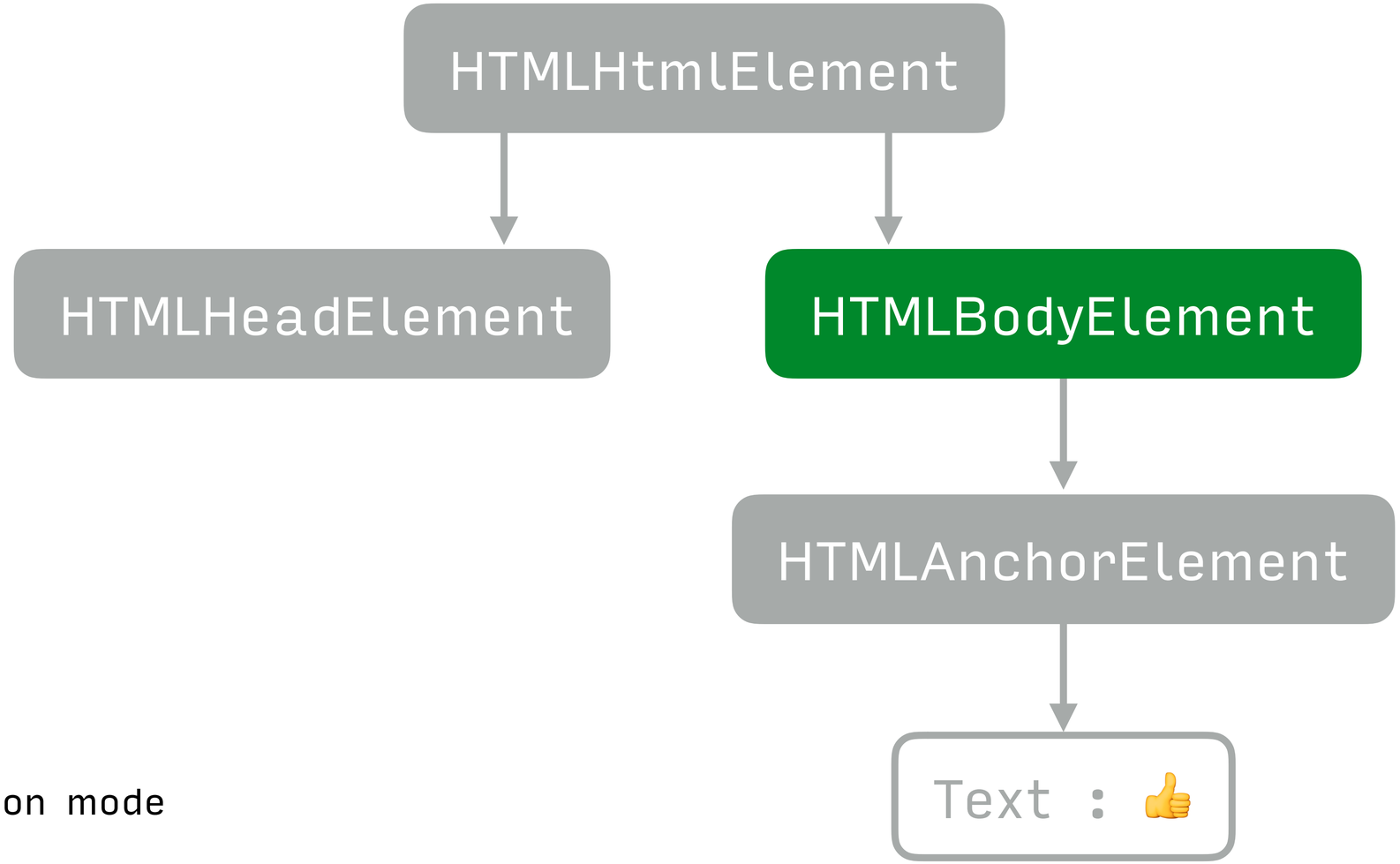
in body insertion mode

👍



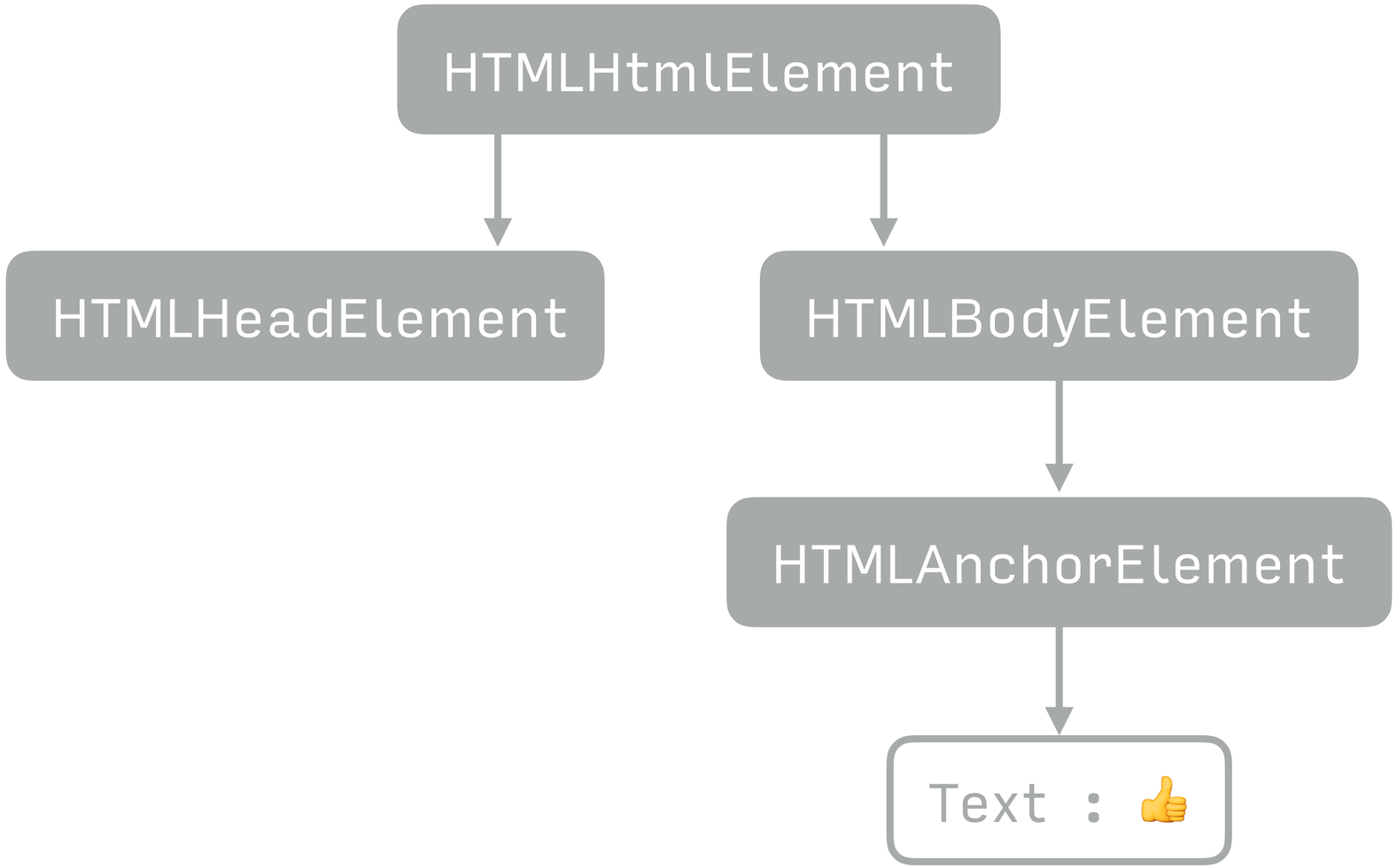
in body insertion mode

 👍



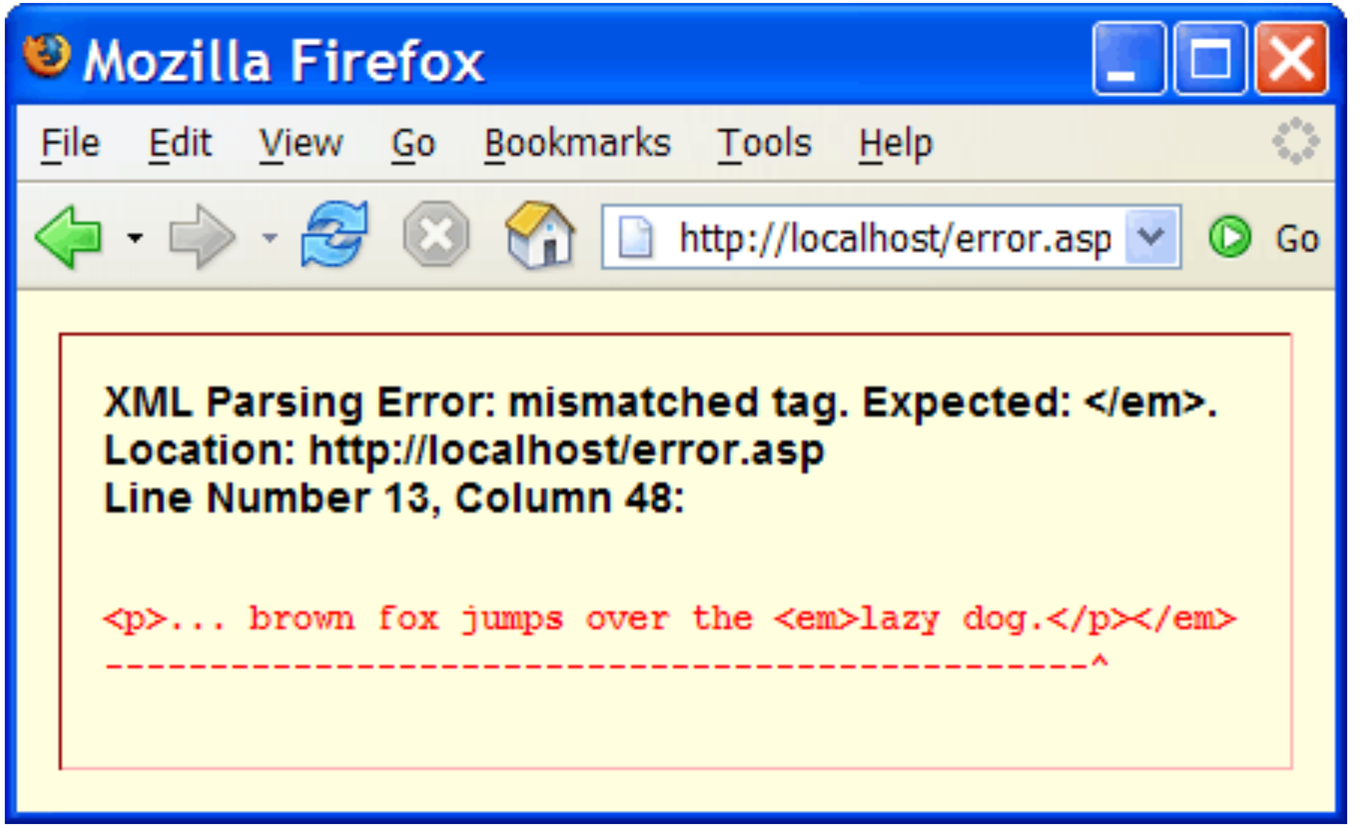
in body insertion mode

 👍


HTMLHtmlElement
HTMLHeadElement
HTMLBodyElement
HTMLAnchorElement
Text : 👍

fixing mistakes

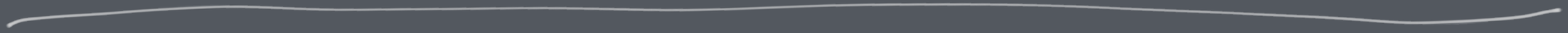
 1 2 3


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/error.asp
Go
XML Parsing Error: mismatched tag. Expected: .
Location: http://localhost/error.asp
Line Number 13, Column 48:
... brown fox jumps over the lazy dog.
--^

 1 2 3

 1 2 3

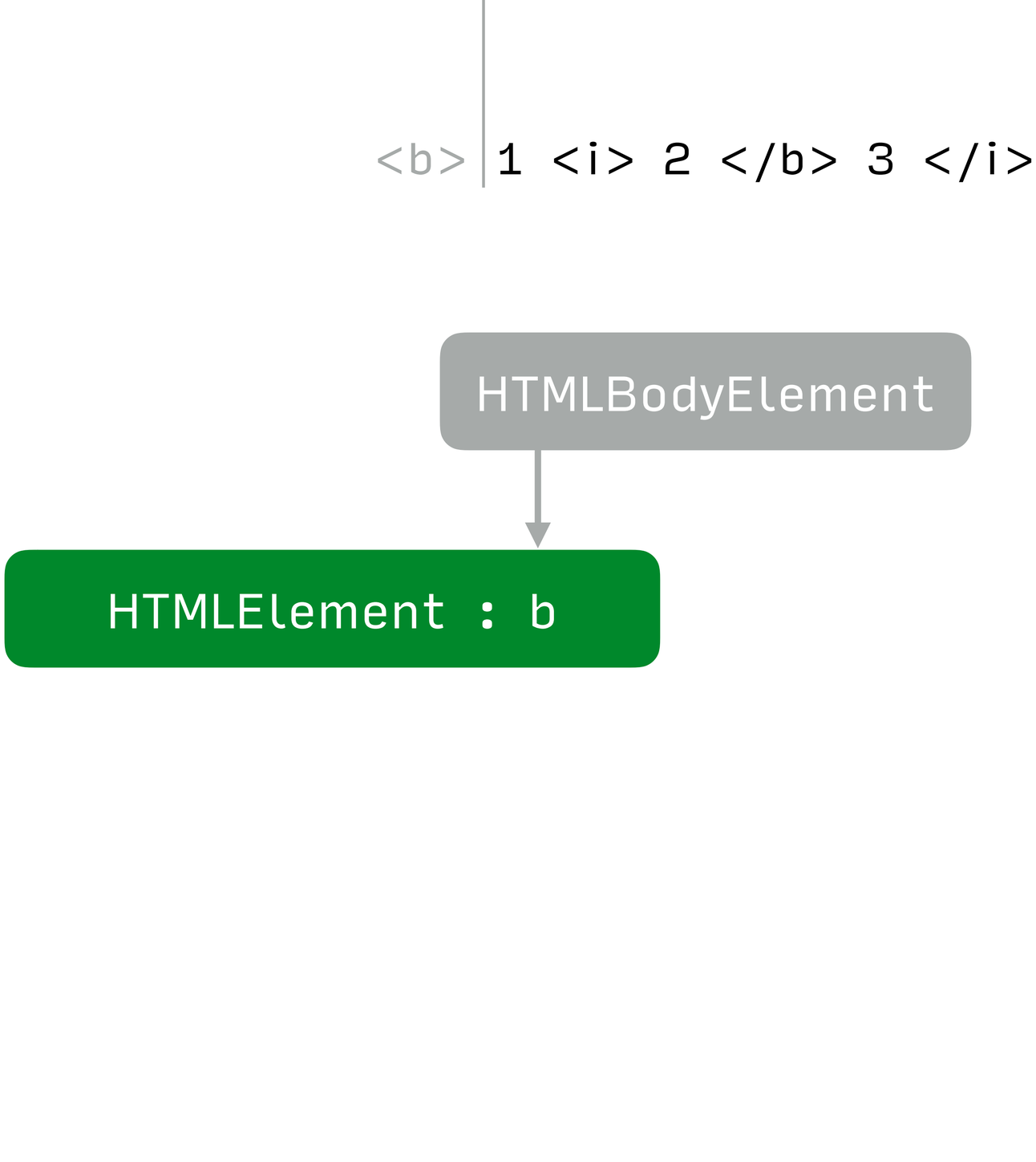
A B C

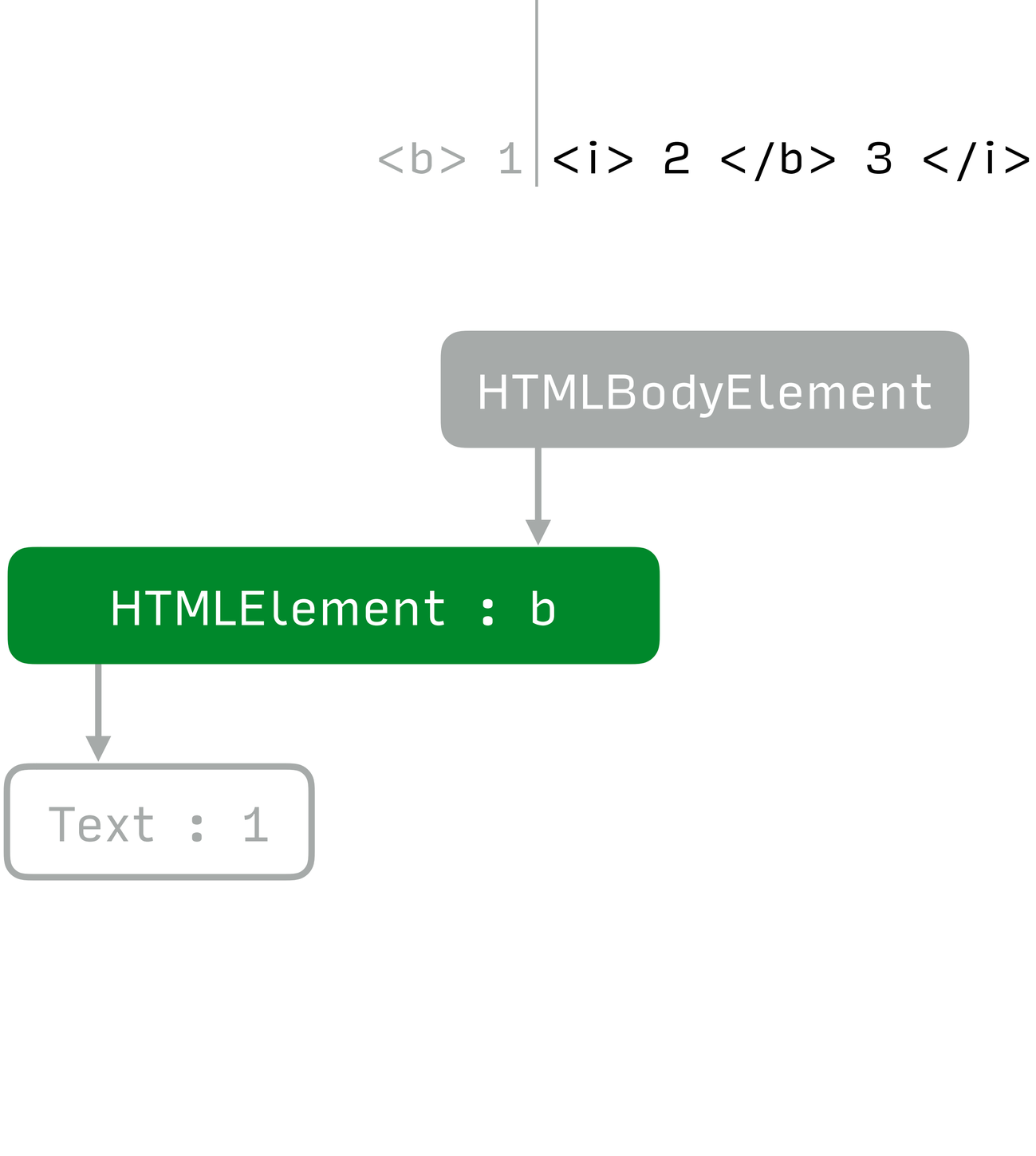


A	B	C
3	*3*	3
Bold Italic	**Italic**	**Regular**

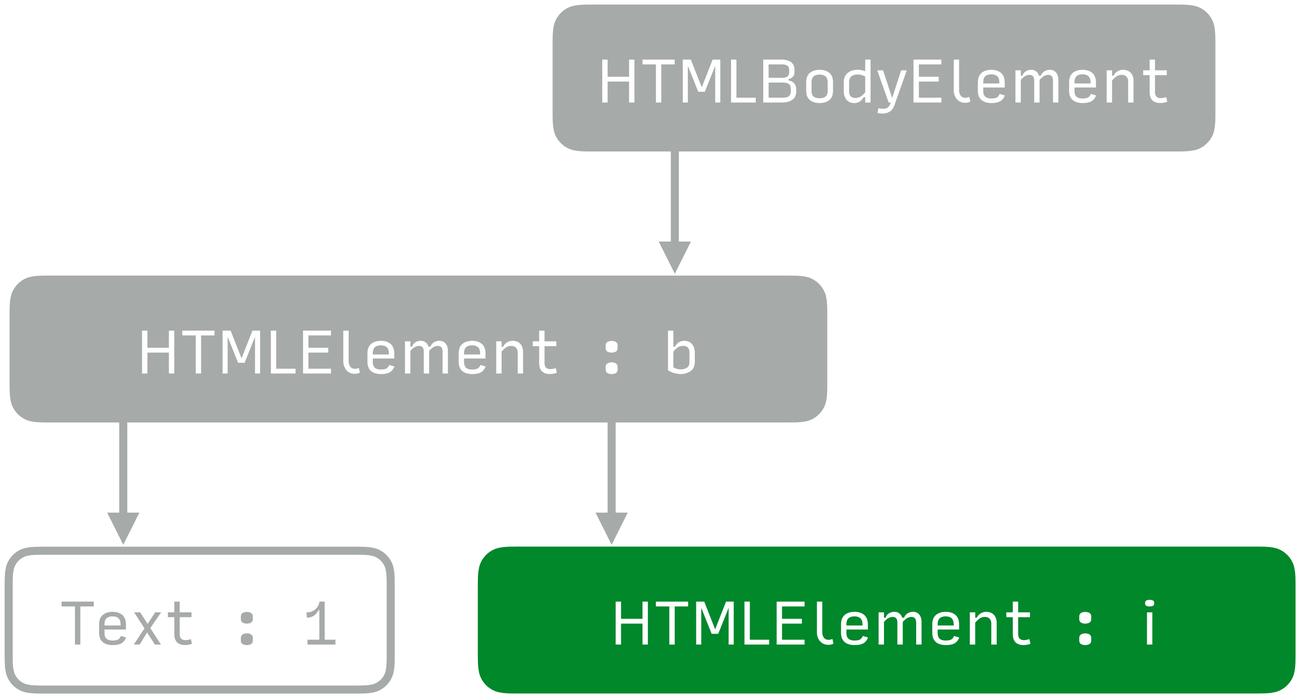
```
<b> 1 <i> 2 </b> 3 </i>
```

HTMLBodyElement

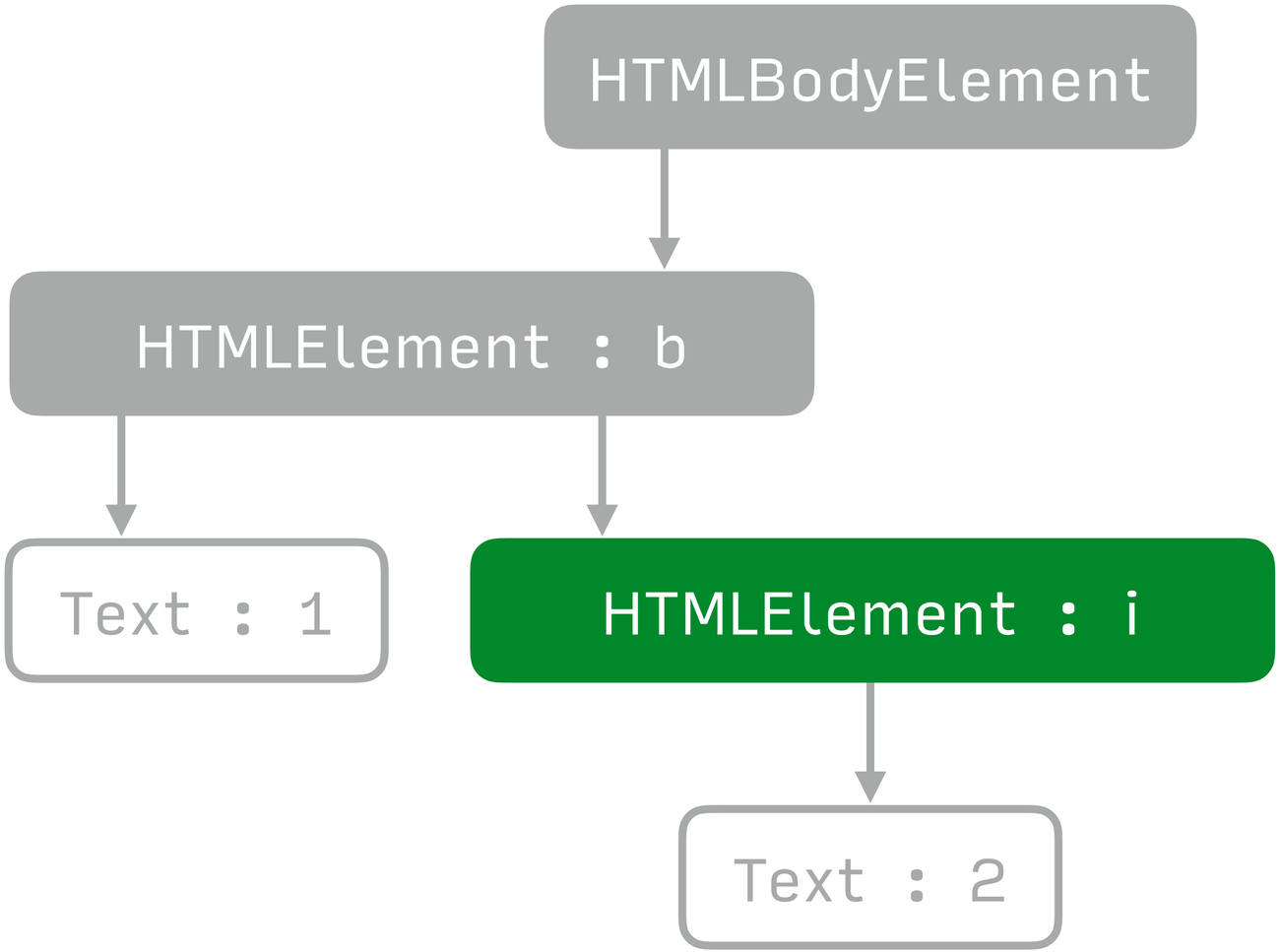

 1 2 3
HTMLBodyElement
HTMLElement : b


 1 2 3
HTMLBodyElement
HTMLElement : b
Text : 1

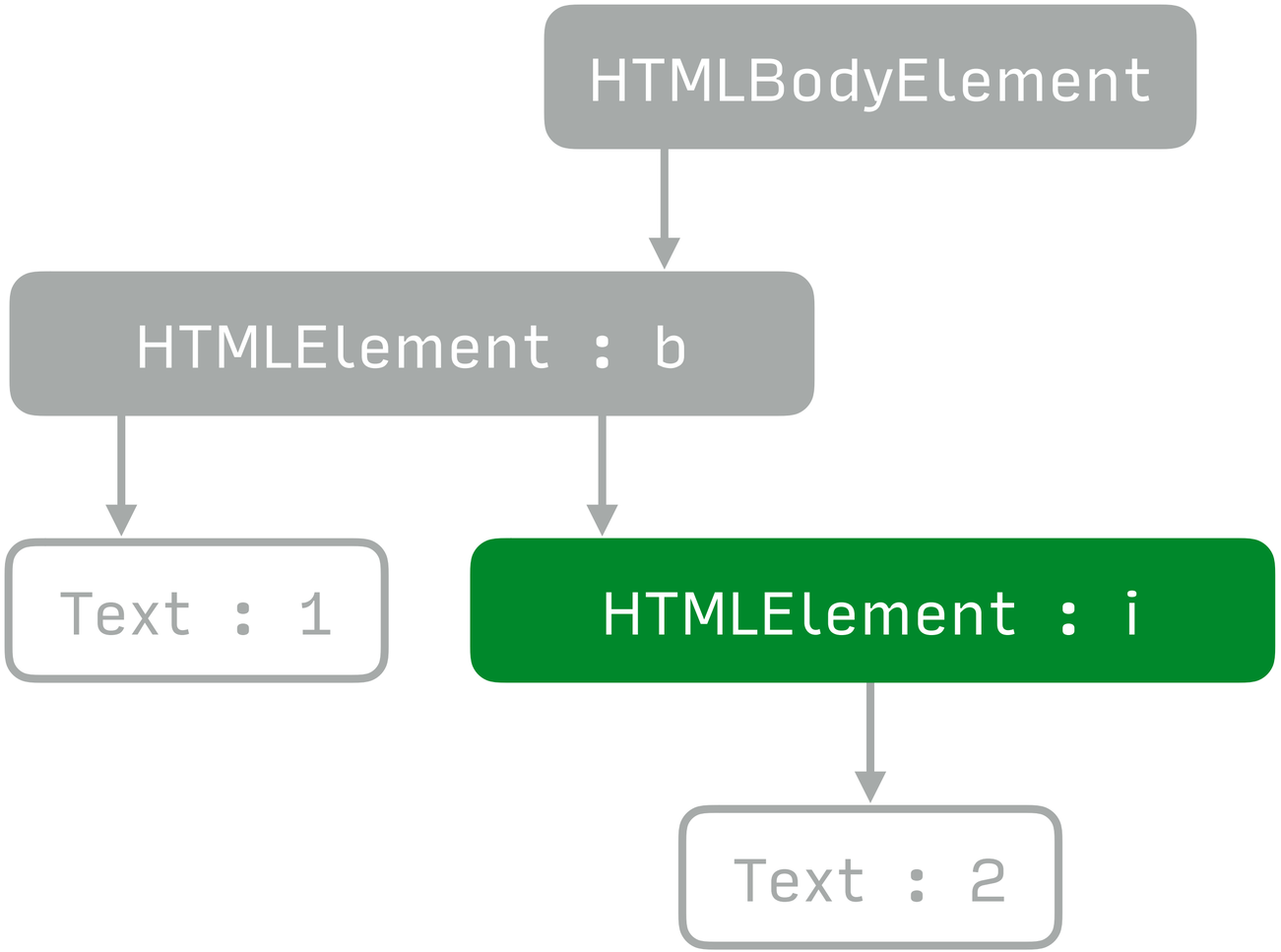
 1 2 3

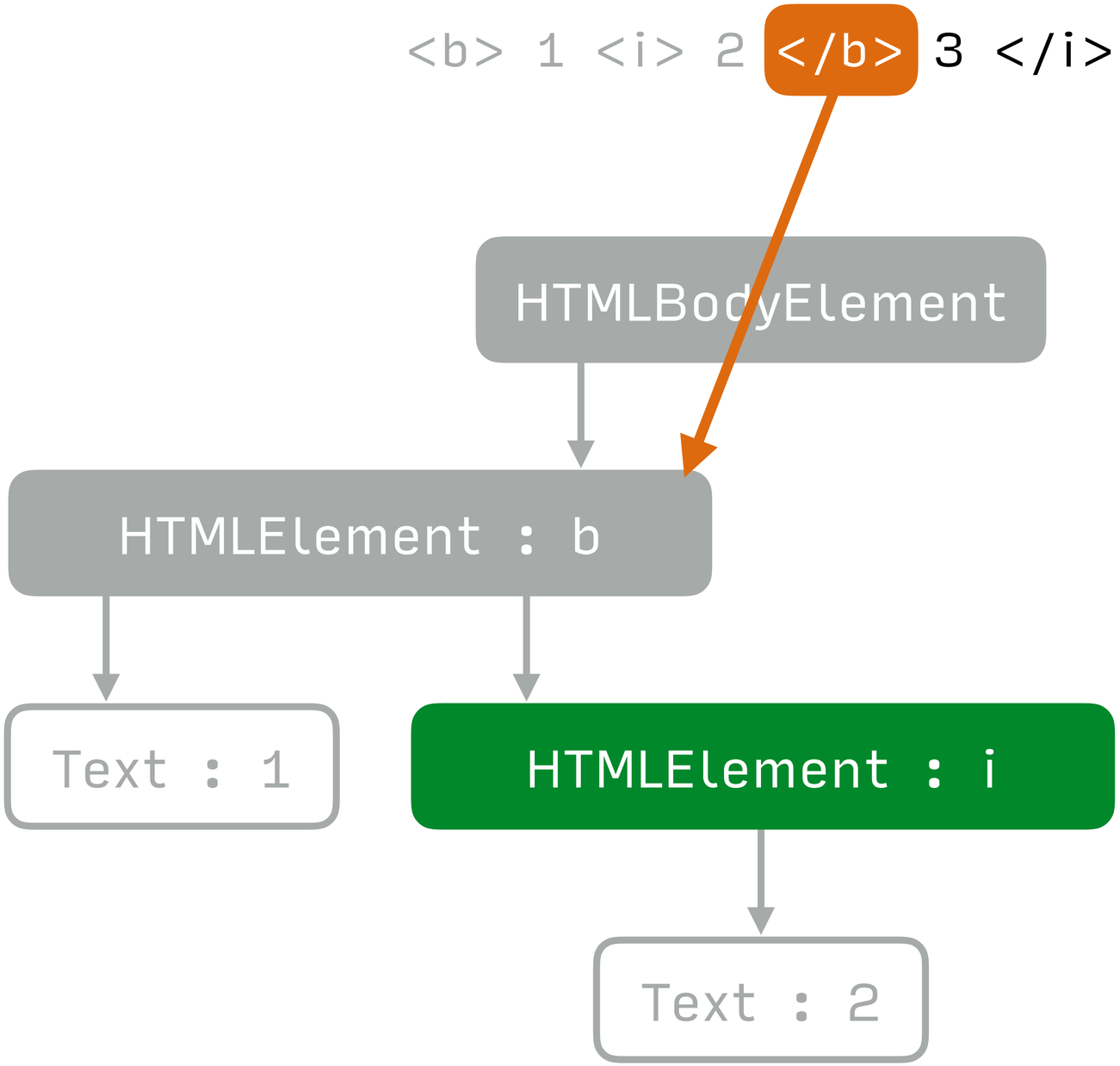


 1 2 3

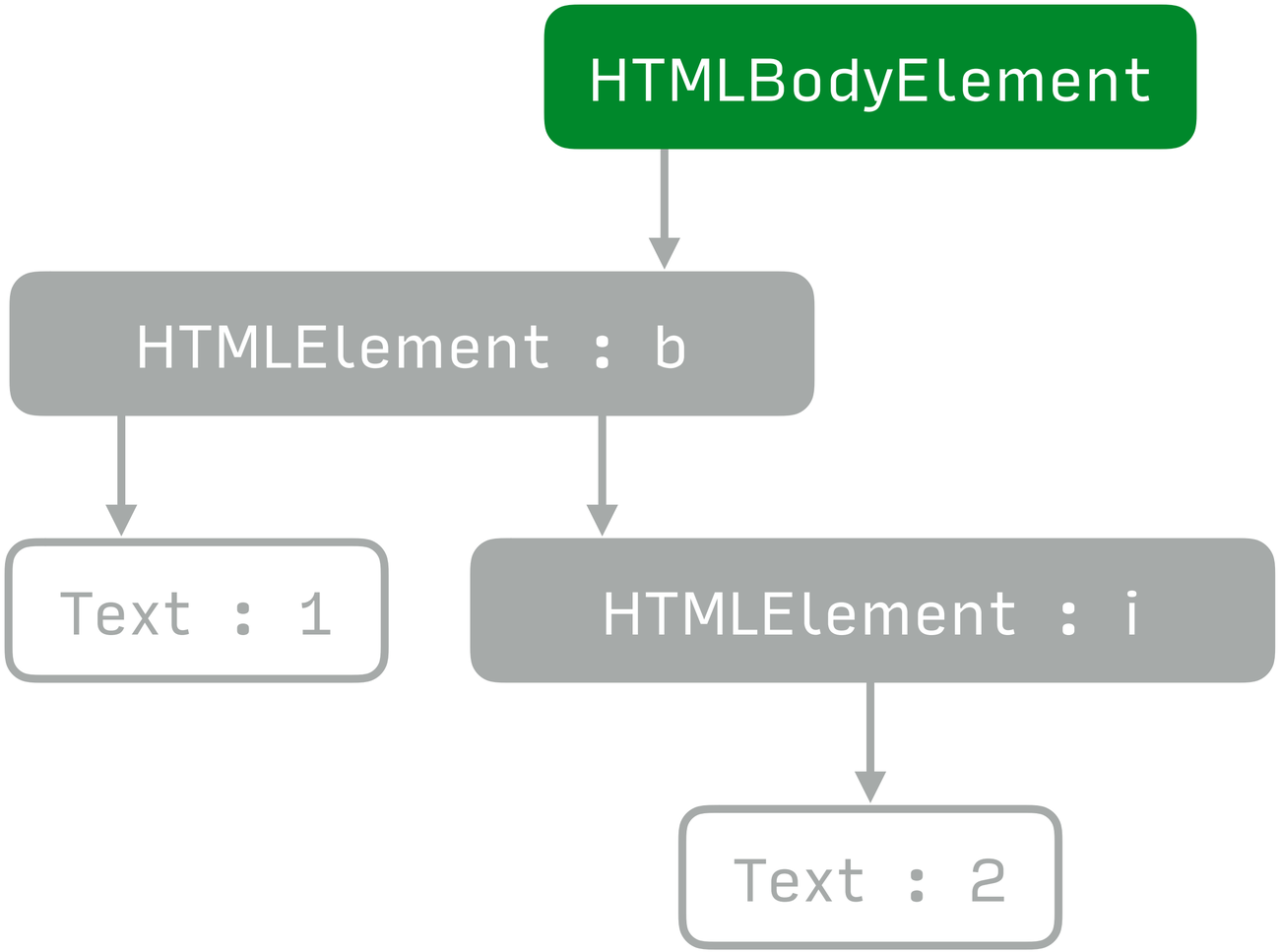


 1 2 3

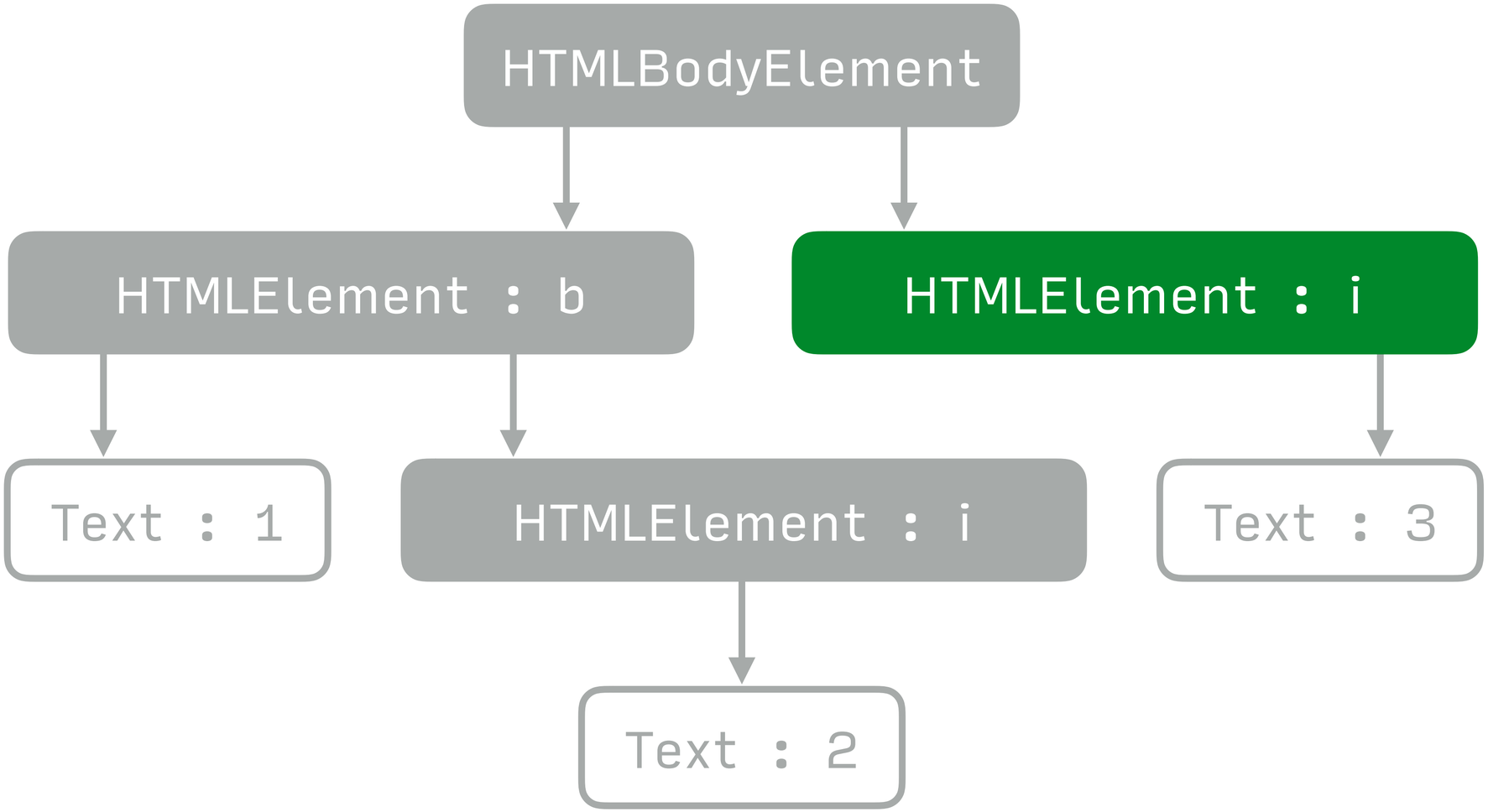



 1 2 3
HTMLBodyElement
HTMLElement : b
Text : 1
HTMLElement : i
Text : 2

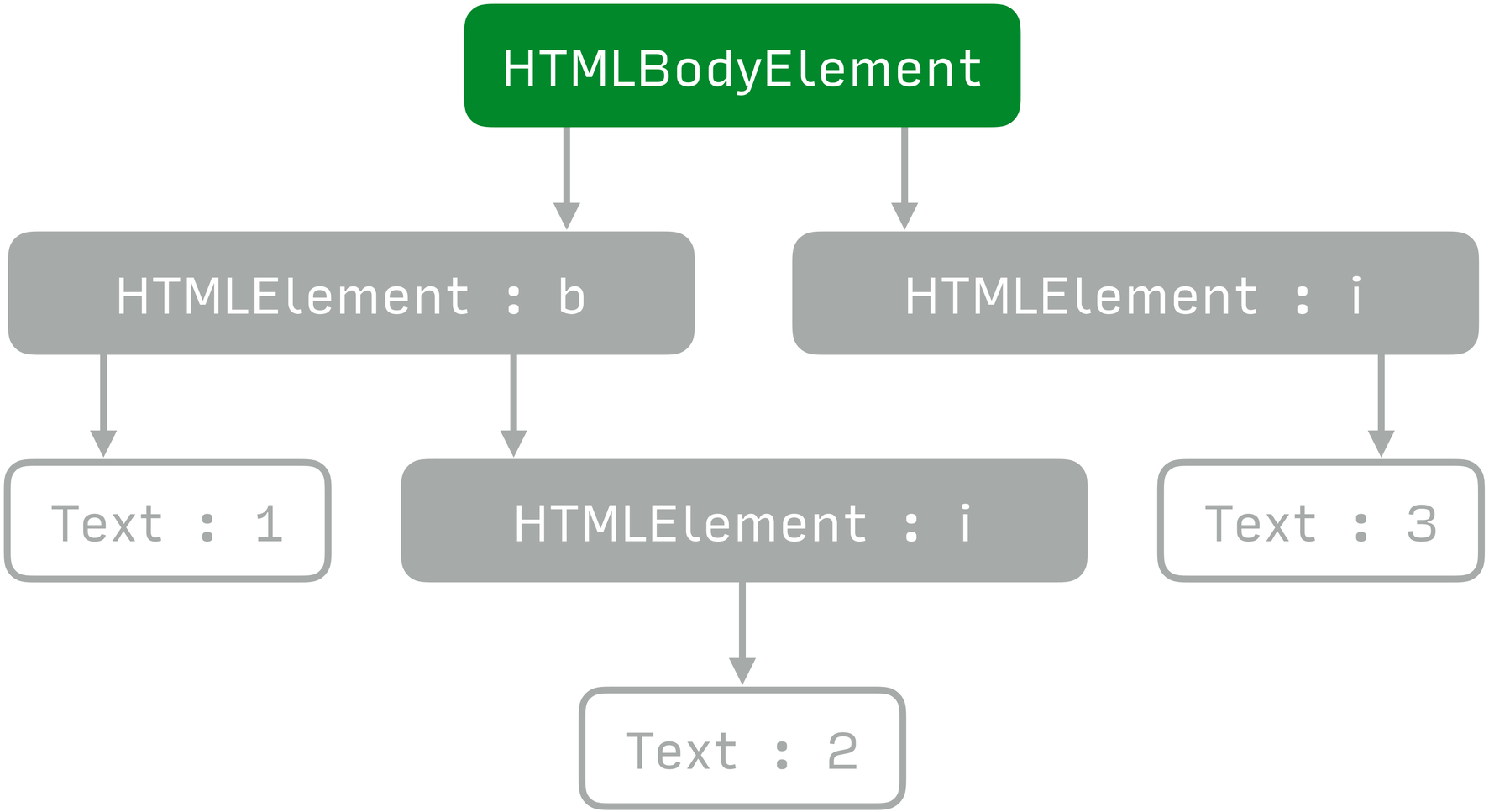
 1 2 3



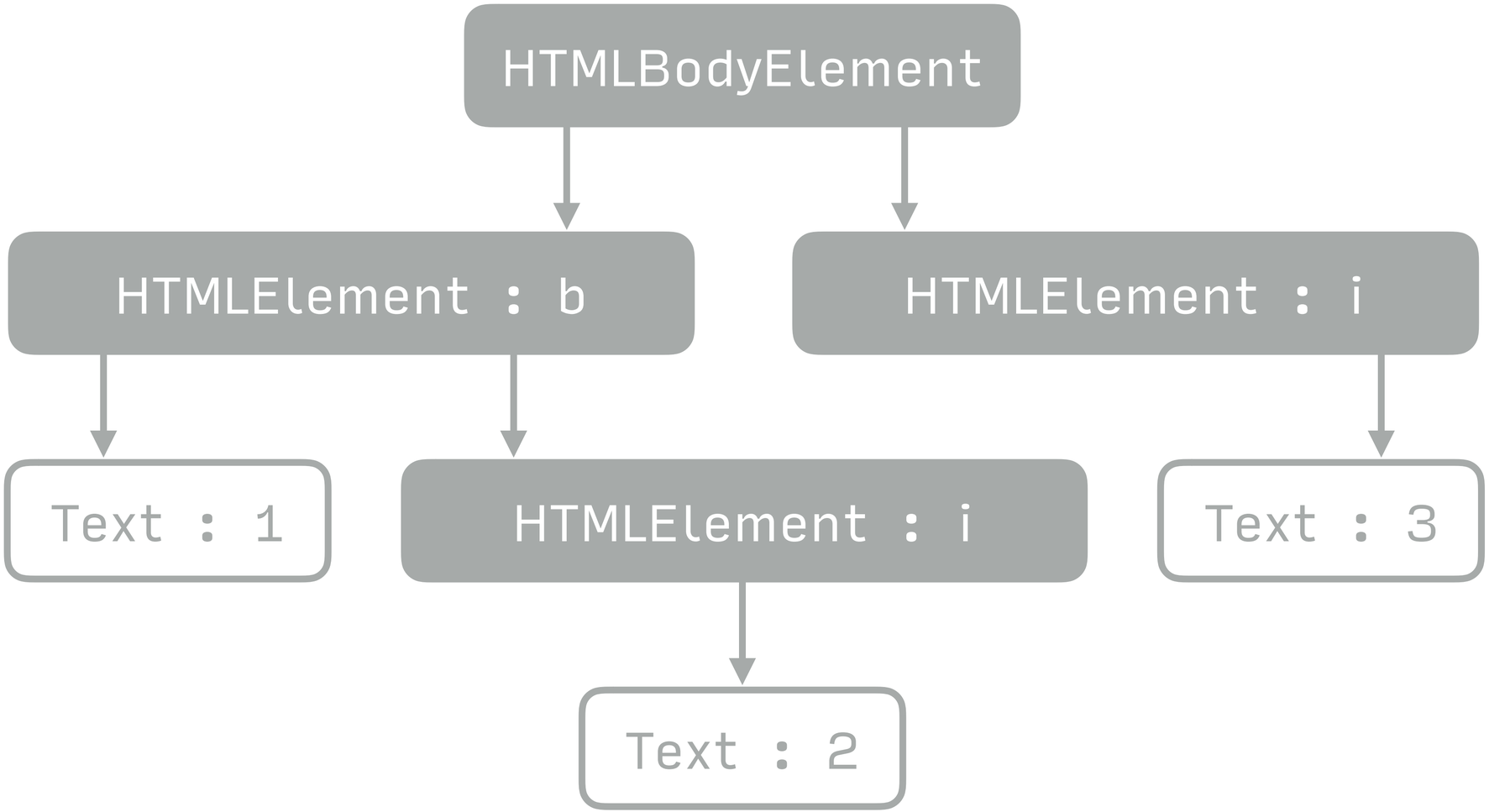
 1 2 3


HTMLBodyElement
HTMLElement : b
HTMLElement : i
Text : 1
HTMLElement : i
Text : 3
Text : 2

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

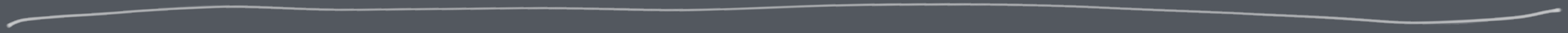

HTMLBodyElement
HTMLElement : b
HTMLElement : i
Text : 1
HTMLElement : i
Text : 3
Text : 2

 1 2 3


HTMLBodyElement
HTMLElement : b
HTMLElement : i
Text : 1
HTMLElement : i
Text : 3
Text : 2

 1 2 3

A B C

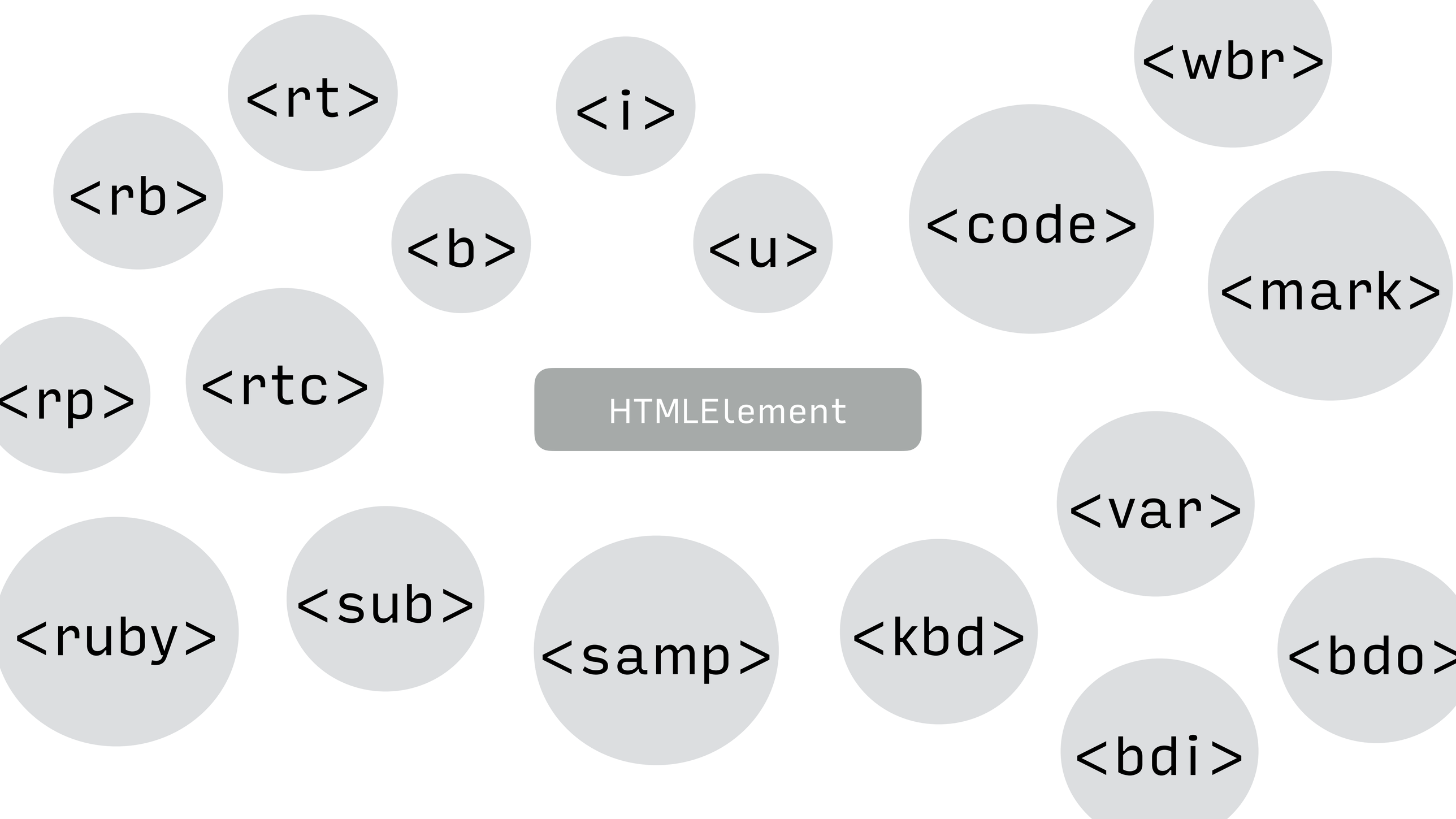


A	B	C
3	*3*	3
Bold Italic	**Italic**	**Regular**

 1 2 3

aside

HTMLElement


<rt>
<i>
<wbr>
<rb>
<b>
<u>
<code>
<mark>
<rp>
<rtc>
HTMLElement
<var>
<ruby>
<sub>
<samp>
<kbd>
<bdo>
<bdi>

```
911
912  /* inline elements */
913
914  u, ins {
915      text-decoration: underline
916  }
917
918  strong, b {
919      font-weight: bold
920  }
921
922  i, cite, em, var, address, dfn {
923      font-style: italic
924  }
925
926  tt, code, kbd, samp {
927      font-family: monospace
```

been popped from the stack.

↪ **An end tag whose tag name is one of: "h1", "h2", "h3", "h4", "h5", "h6"**

If the stack of open elements does not have an element in scope that is an HTML element and whose tag name is one of "h1", "h2", "h3", "h4", "h5", or "h6", then this is a parse error; ignore the token.

Otherwise, run these steps:

1. Generate implied end tags.
2. If the current node is not an HTML element with the same tag name as that of the token, then this is a parse error.
3. Pop elements from the stack of open elements until an HTML element whose tag name is one of "h1", "h2", "h3", "h4", "h5", or "h6" has been popped from the stack.

↪ **An end tag whose tag name is "sarcasm"**

Take a deep breath, then act as described in the "any other end tag" entry below.

↪ **A start tag whose tag name is "a"**

If the list of active formatting elements contains an `a` element between the end of the list and the last marker on the list (or the start of the list if there is no marker on the list), then this is a parse error; run the adoption agency algorithm for the tag name "a", then remove that element from the list of active formatting elements and the stack of open elements if the adoption agency algorithm didn't already remove it (it might not have if the element is not in table scope).

> In the non-conforming stream `<a href="a">a<table><a href="b">b</table>x`, the first `a` element would be closed upon seeing the second one, and the "x" character would be inside a link to "b", not to "a". This is despite the fact that the outer `a` element is not in table scope (meaning that a regular `</a>` end tag at the start of the table wouldn't close the outer `a` element). The result is that the two `a` elements are indirectly nested inside each other — non-conforming markup will often result in non-conforming DOMs when parsed.

Reconstruct the active formatting elements, if any.

Insert an HTML element for the token. Push onto the list of active formatting elements that element.

↪ **A start tag whose tag name is one of: "b", "big", "code", "em", "font", "i", "s", "small", "strike", "strong", "tt", "u"**

been popped from the stack.

↪ **An end tag whose tag name is one of: "h1", "h2", "h3", "h4", "h5", "h6"**

If the stack of open elements does not have an element in scope that is an HTML element and whose tag name is one of "h1", "h2", "h3", "h4", "h5", or "h6", then this is a parse error; ignore the token.

Otherwise, run these steps:

1. Generate implied end tags.
2. If the current node is not an HTML element with the same tag name as that of the token, then this is a parse error.
3. Pop elements from the stack of open elements until an HTML element whose tag name is one of "h1", "h2", "h3", "h4", "h5", or "h6" has been popped from the stack.

↪ **An end tag whose tag name is "sarcasm"**

Take a deep breath, then act as described in the "any other end tag" entry below.

↪ **A start tag whose tag name is "a"**

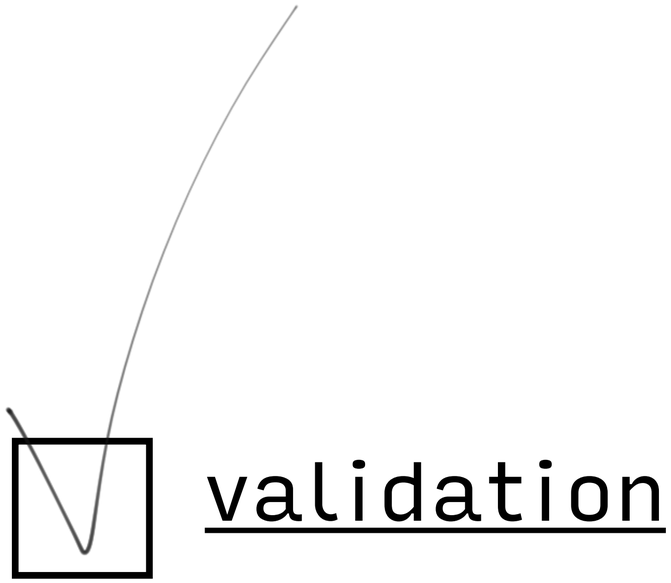
If the list of active formatting elements contains an `a` element between the end of the list and the last marker on the list (or the start of the list if there is no marker on the list), then this is a parse error; run the adoption agency algorithm for the tag name "a", then remove that element from the list of active formatting elements and the stack of open elements if the adoption agency algorithm didn't already remove it (it might not have if the element is not in table scope).

> In the non-conforming stream `<a href="a">a<table><a href="b">b</table>x`, the first `a` element would be closed upon seeing the second one, and the "x" character would be inside a link to "b", not to "a". This is despite the fact that the outer `a` element is not in table scope (meaning that a regular `</a>` end tag at the start of the table wouldn't close the outer `a` element). The result is that the two `a` elements are indirectly nested inside each other — non-conforming markup will often result in non-conforming DOMs when parsed.

Reconstruct the active formatting elements, if any.

Insert an HTML element for the token. Push onto the list of active formatting elements that element.

↪ **A start tag whose tag name is one of: "b", "big", "code", "em", "font", "i", "s", "small", "strike", "strong", "tt", "u"**


validation

validator.w3.org

Markup Validation Service

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

Validate by URI | Validate by File Upload | Validate by Direct Input

Validate by URI

Validate a document online:

Address:

▸ More Options

Check

This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS

validator.w3.org

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!

Result:	3 Errors, 3 warning(s)	
Address :	http://info.cern.ch/hypertext/WWW/TheProject.html	
Encoding :	utf-8	(detect automatically)
Doctype :	HTML 2.0	HTML 2.0
Root Element:	html	

I ♥ VALIDATOR

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

5643

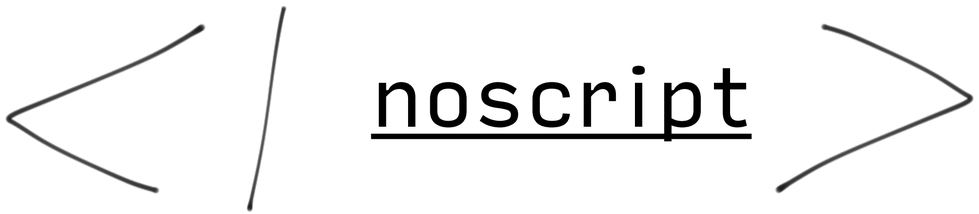
<DD> A summary of the history
of the project.
<DT><A
NAME=37 HREF="Helping.html">How can I help</A> ?
<DD> If you would like
to support the web..
<DT><A
NAME=48 HREF="../README.html">Getting code</A>
<DD> Getting the code by<A
NAME=49 HREF="LineMode/Defaults/Distribution.html">
anonymous FTP</A> , etc.</A>
</DL>
</BODY>

Sorry, this page is not valid AMP HTML

Click here to continue to www.nbcnews.com

Errors

1. Error in line 635, column 52: Invalid URL protocol '%20http:' for attribute 'href' in tag 'a'.


noscript

all browsers support javascript

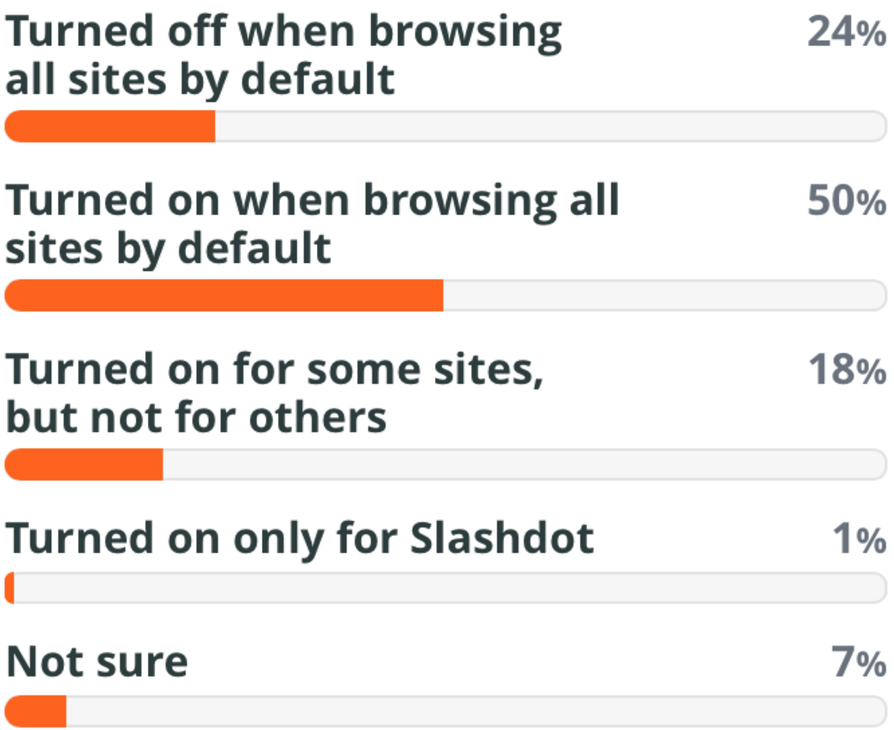
nobody disables javascript anymore

and if you do, stuff will break

deal with it

Regarding javascript on my default desktop web browser I have it:

13,933 Votes

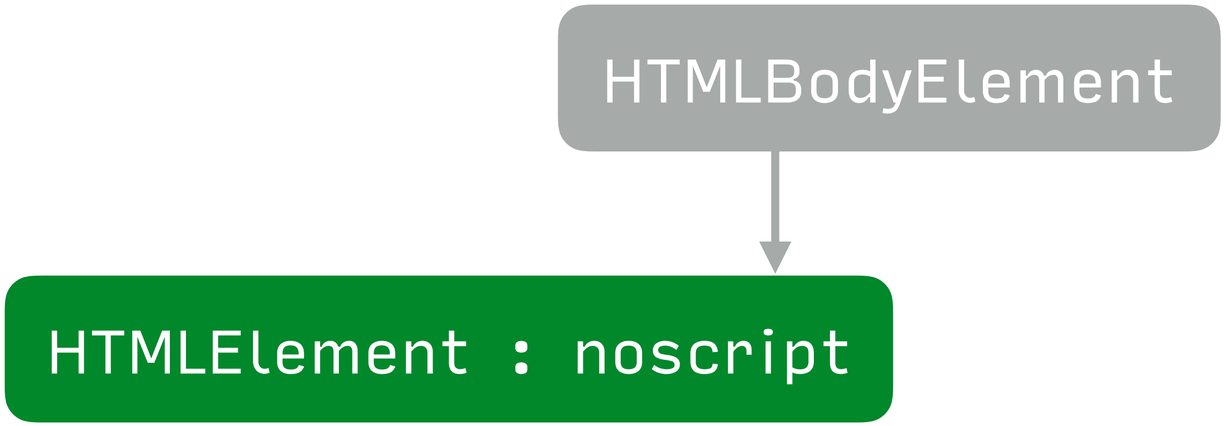


```
<noscript>
    <noscript>
        …
    </noscript>
</noscript>
```

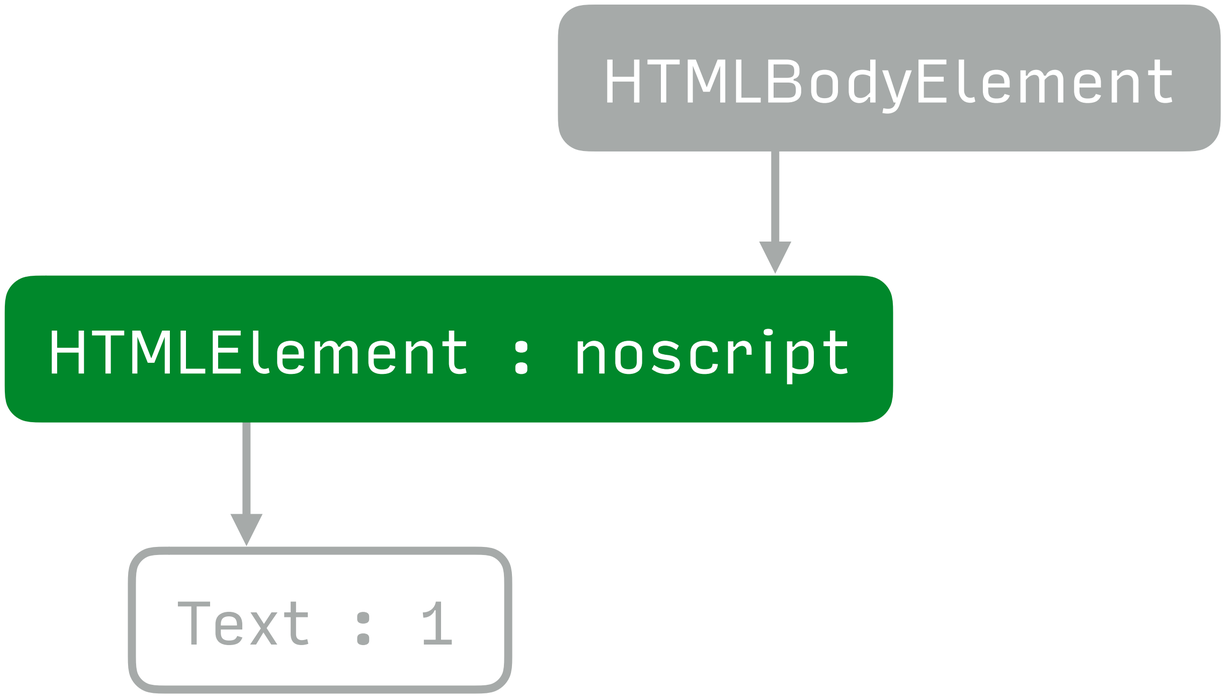
<noscript> 1 <noscript> 2 </noscript> 3 </noscript>

HTMLBodyElement

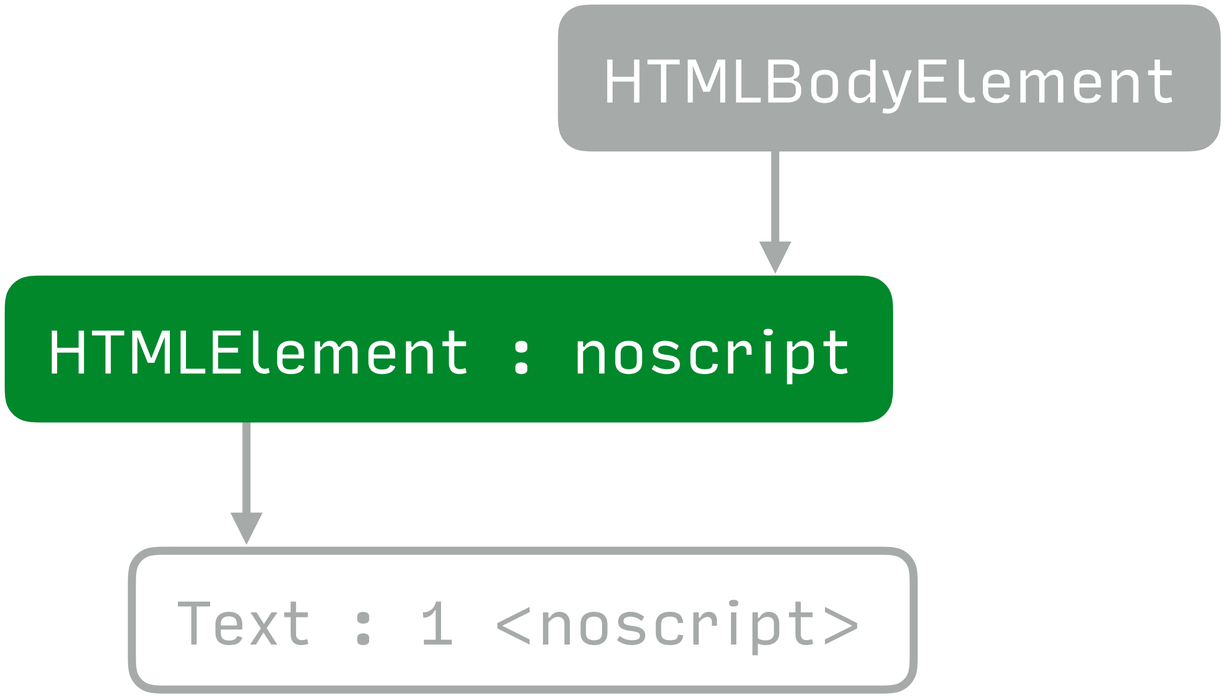
<noscript> | 1 <noscript> 2 </noscript> 3 </noscript>



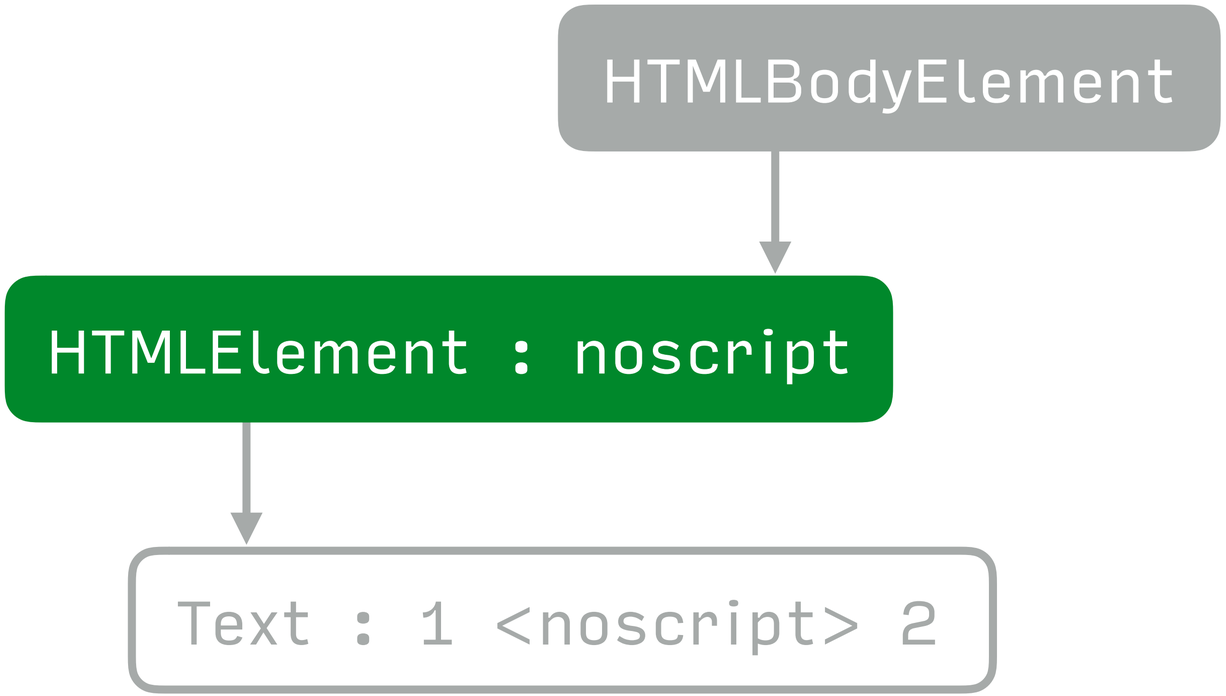
<noscript> 1 | <noscript> 2 </noscript> 3 </noscript>



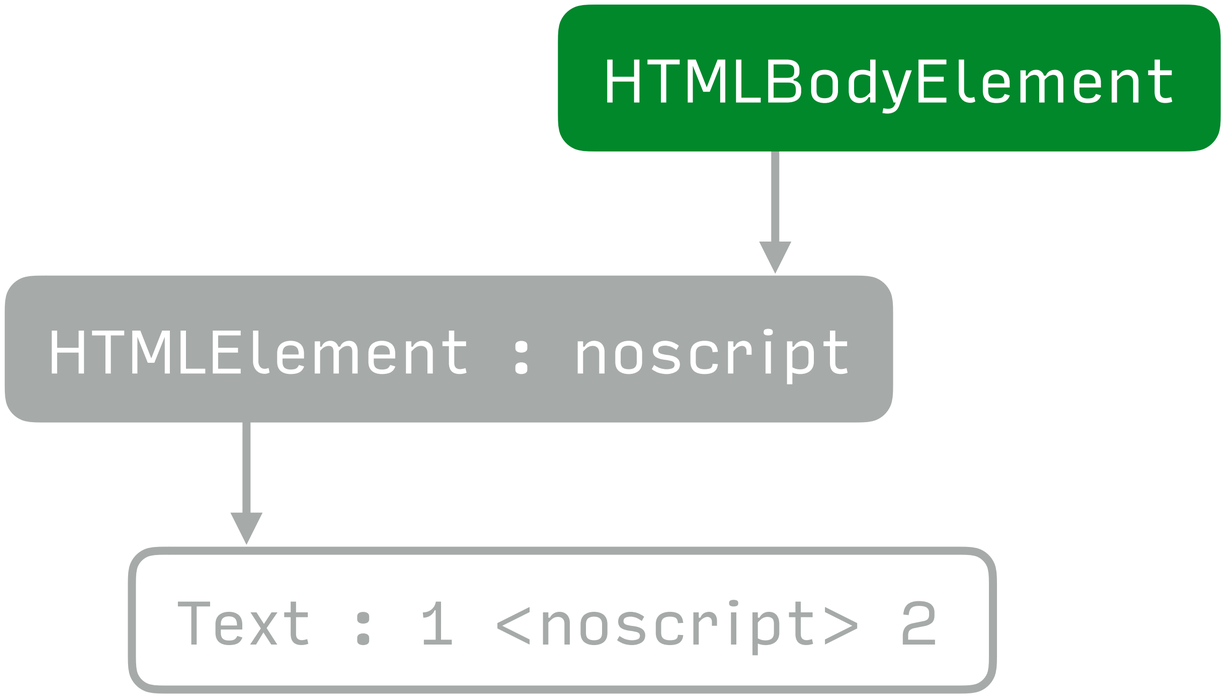
<noscript> 1 <noscript> 2 </noscript> 3 </noscript>



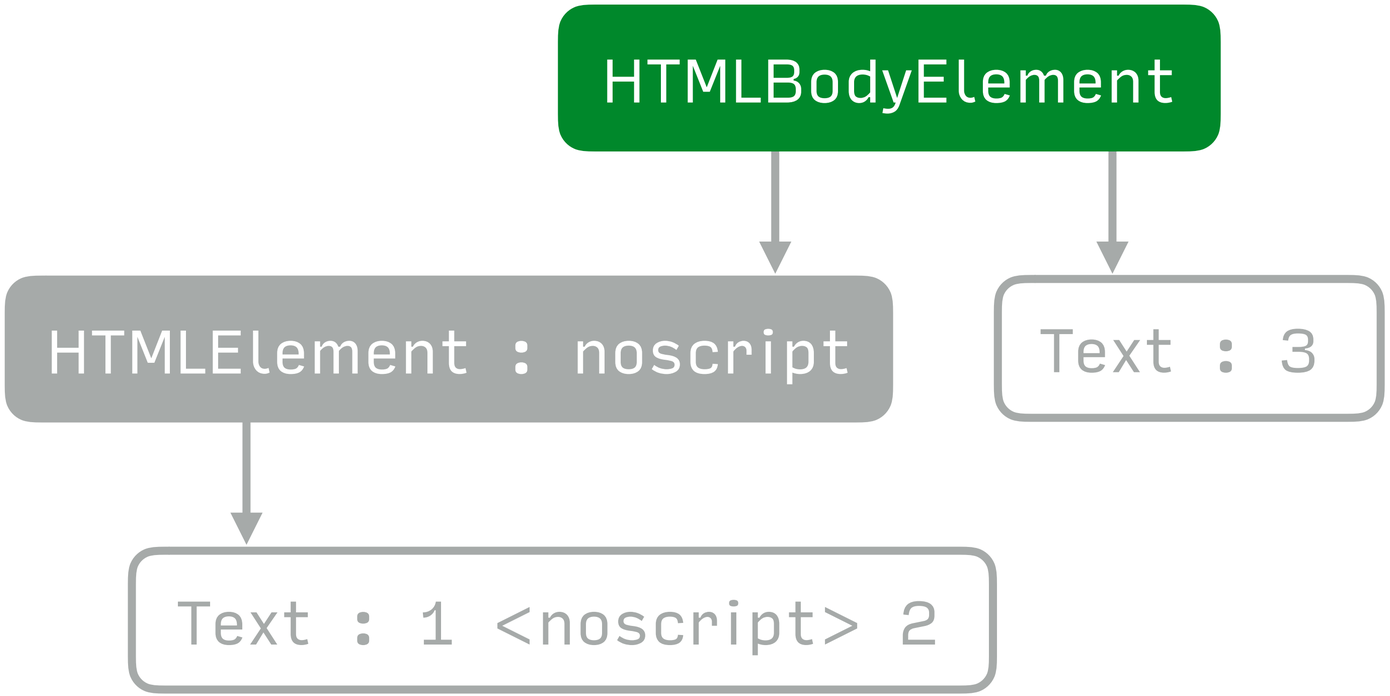
<noscript> 1 <noscript> 2 </noscript> 3 </noscript>



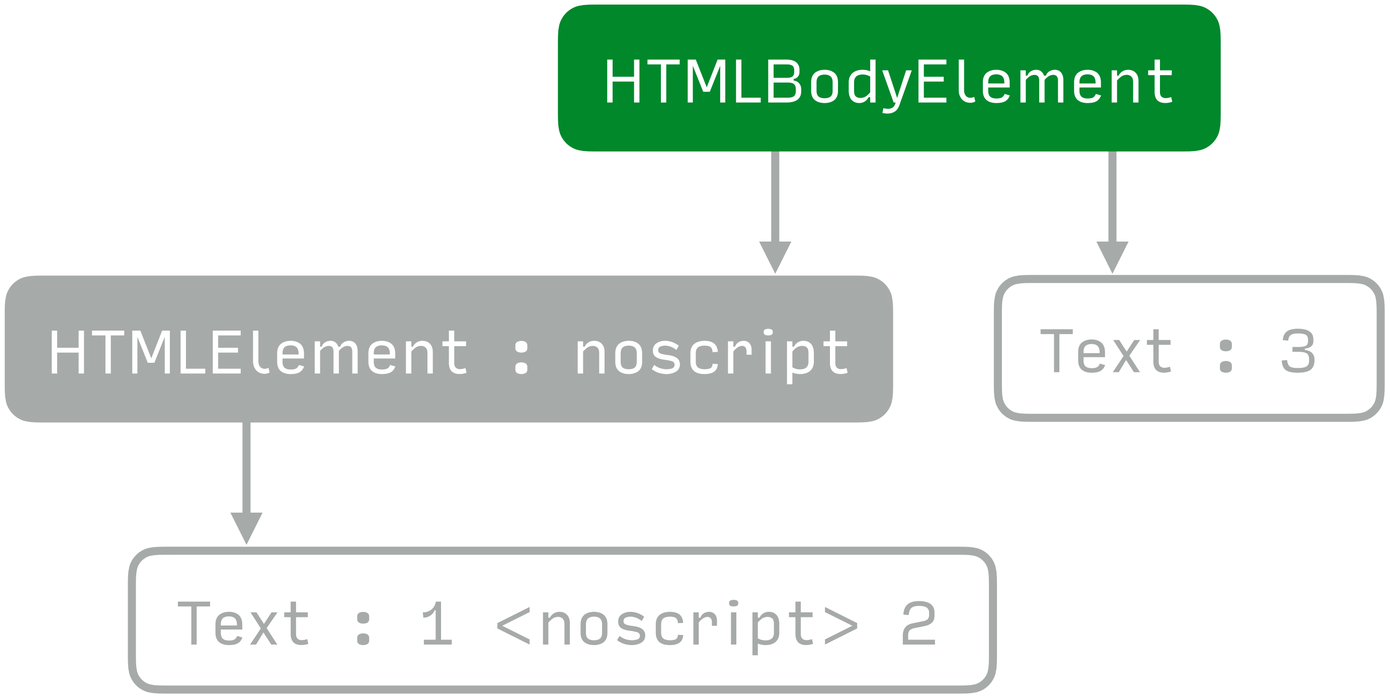
<noscript> 1 <noscript> 2 </noscript> 3 </noscript>



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

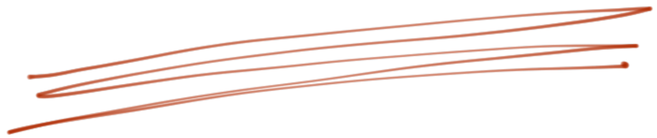


```
<noscript> 1 <noscript> 2 </noscript> 3 </noscript>
```



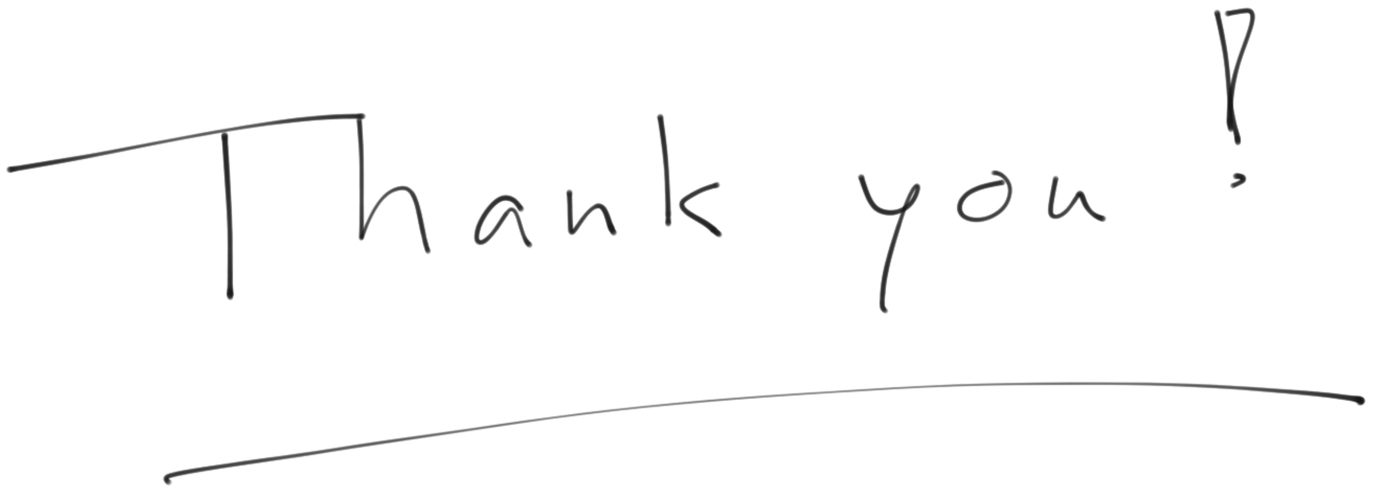
```
<noscript style='…'>
   …
</noscript>
```

no



```
<noscript>
    <script>
        …
    </script>
</noscript>
```

wtf?



@html5test

#HTMLQuiz

https://html5te.st/quiz