

CSS

CSS FUNDAMENTALS

Pseudo-element



IN A ROCKET

Learn front-end development at *rocket speed*

“**Pseudo-elements** create abstractions about the document tree beyond those specified by the document language. They may also provide authors a way to refer to content that does not exist in the source document.”



SOURCE: [Selectors Level 3](#) by W3C.

Pseudo-elements

::first-letter

::first-line

::before

::after

PSEUDO-ELEMENTS / FIRST-LETTER

Represents the first letter of an element, if it is not preceded by any other content on its line.

Syntax `selector::first-letter {style properties}`

`p::first-letter {color: green}`

With this code the first letter of all paragraphs is shown in green.

HTML

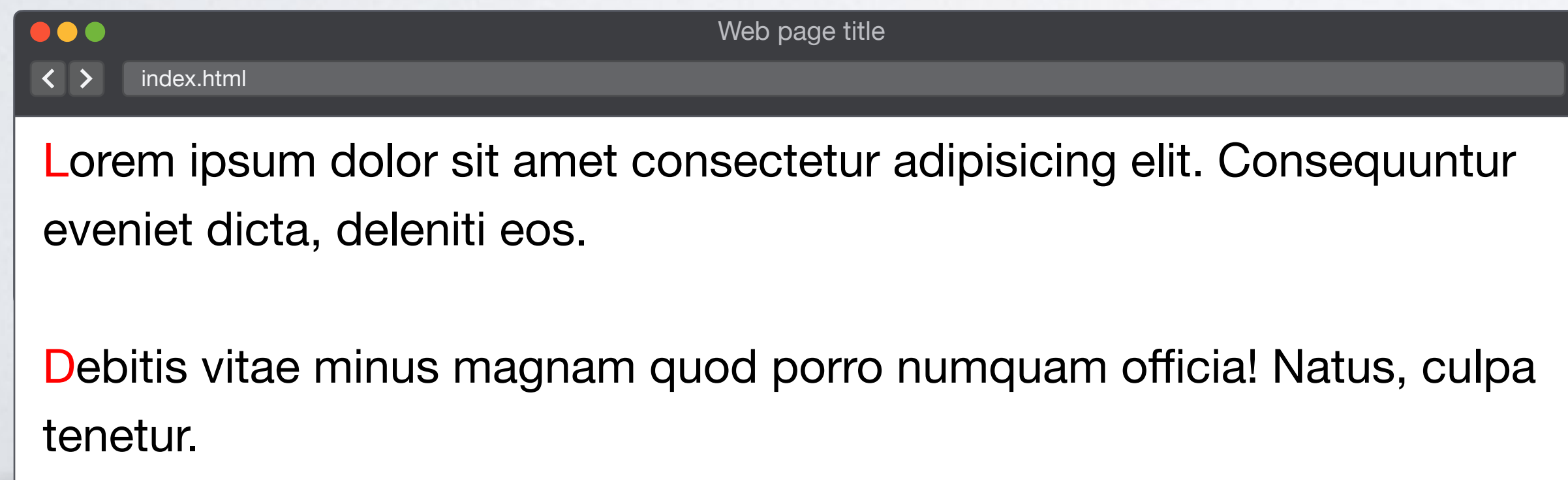
```
<body>  
<p>Lorem ipsum dolor sit amet consectetur  
adipisicing elit. Consequuntur eveniet dicta,  
deleniti eos.</p>
```

```
<p>Debitis vitae minus magnam quod porro  
numquam officia! Natus, culpa tenetur.</p>  
</body>
```

CSS

```
p::first-letter { color: red; }
```

Browser



Pseudo-elements

::first-letter

::first-line

::before

::after

PSEUDO-ELEMENTS / FIRST-LINE

Describes the contents of the first formatted line of an element.

Syntax `selector::first-line {style properties}`

`p::first-line {color: green}`

With this code the first line of all paragraphs is shown in green.

HTML

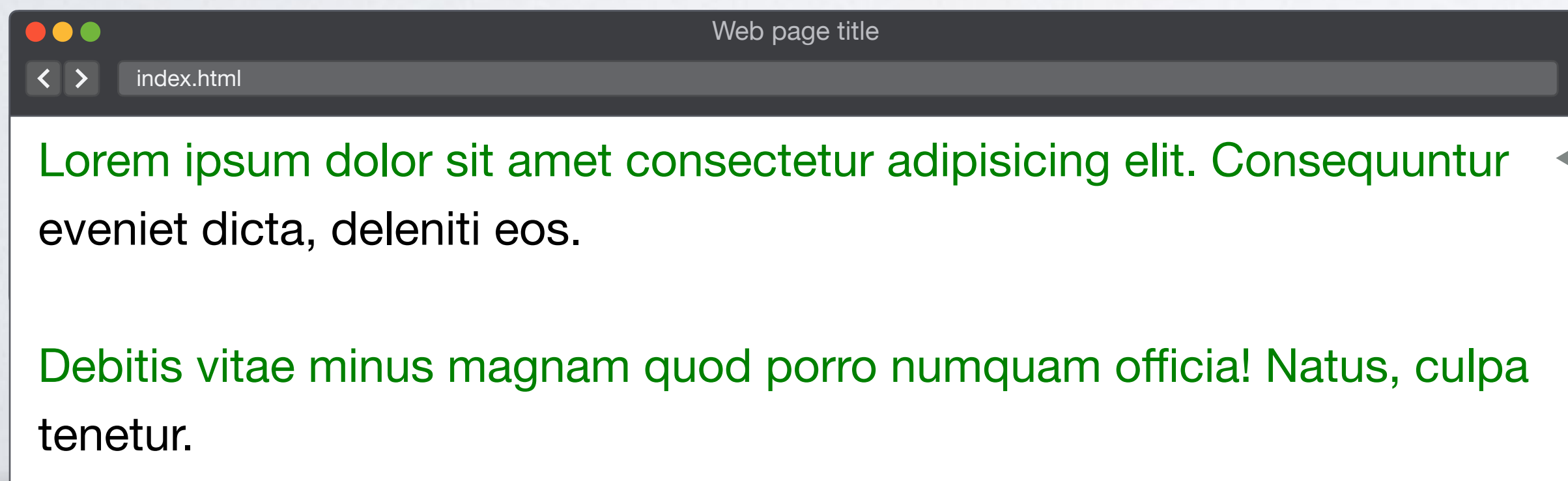
```
<body>  
<p>Lorem ipsum dolor sit amet consectetur  
adipisicing elit. Consequuntur eveniet dicta,  
deleniti eos.</p>
```

```
<p>Debitis vitae minus magnam quod porro  
numquam officia! Natus, culpa tenetur.</p>  
</body>
```

CSS

```
p::first-line { color: green; }
```

Browser



Updated by the browser checking viewport's width

Pseudo-elements

::first-letter

::first-line

::before

::after

PSEUDO-ELEMENTS / BEFORE

Describes generated content before an element's content.

Syntax `selector::before {style properties}`

`p::before {content: "★"}`

With this code all paragraphs are preceded by "★".

HTML

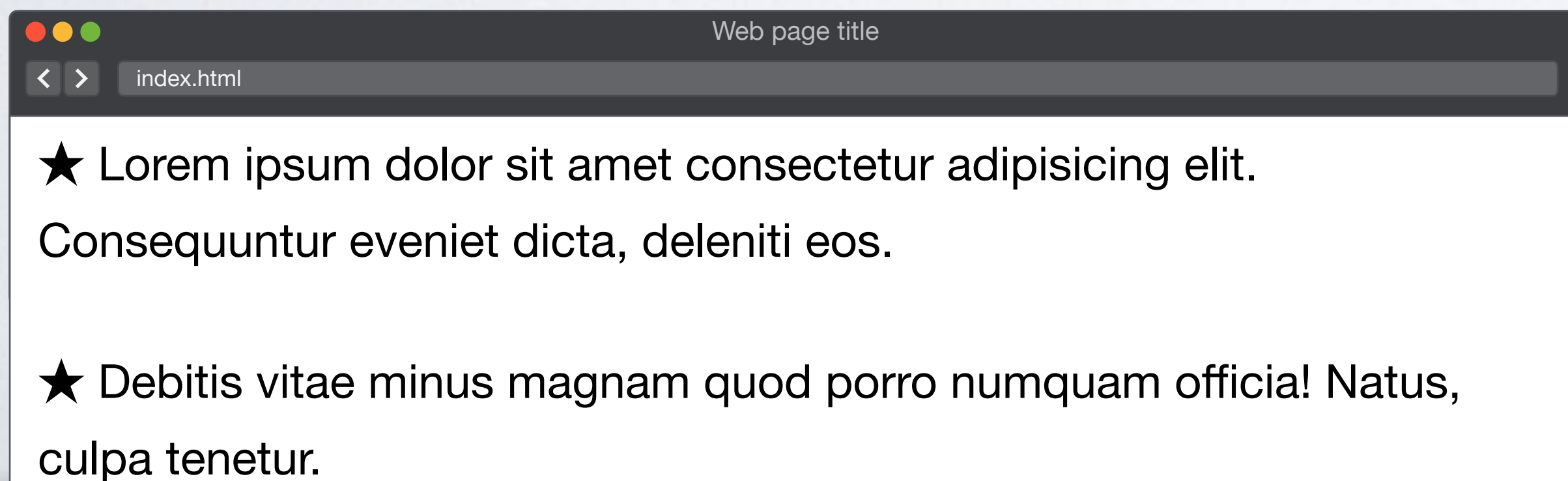
```
<body>  
<p>Lorem ipsum dolor sit amet consectetur  
adipisicing elit. Consequuntur eveniet dicta,  
deleniti eos.</p>
```

```
<p>Debitis vitae minus magnam quod porro  
numquam officia! Natus, culpa tenetur.</p>  
</body>
```

CSS

```
p::before { content: "★"; }
```

Browser



PSEUDO-ELEMENTS / BEFORE

HTML

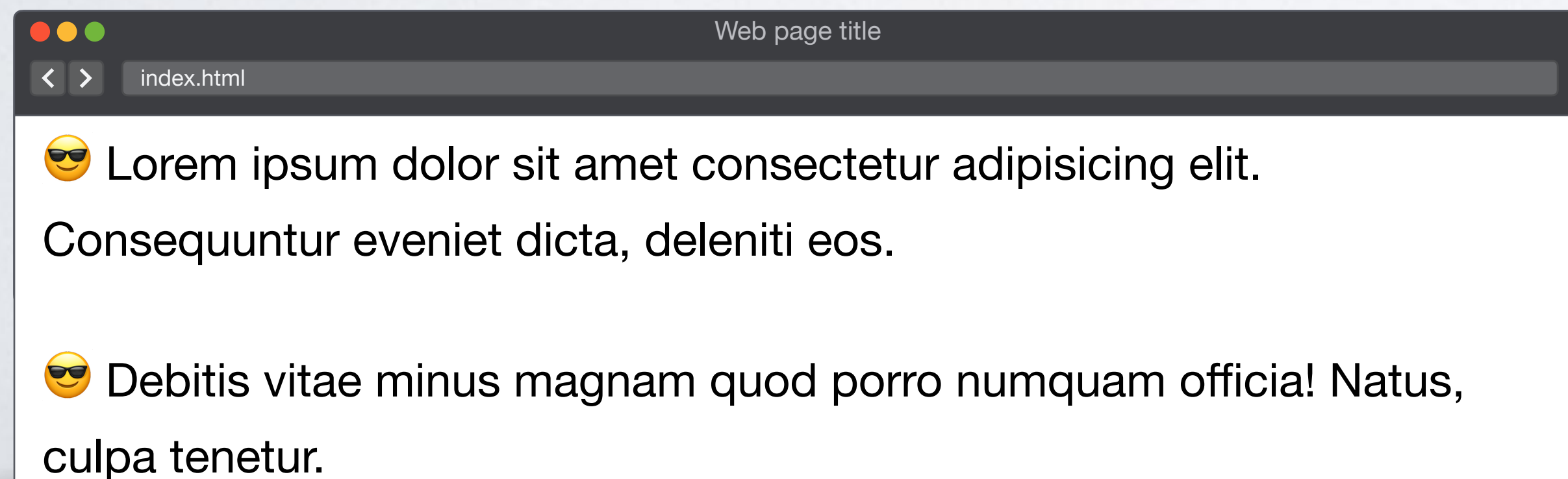
```
<body>  
<p>Lorem ipsum dolor sit amet consectetur  
adipisicing elit. Consequuntur eveniet dicta,  
deleniti eos.</p>
```

```
<p>Debitis vitae minus magnam quod porro  
numquam officia! Natus, culpa tenetur.</p>  
</body>
```

CSS

```
p::before { content: "😎"; }
```

Browser



PSEUDO-ELEMENTS / BEFORE

HTML

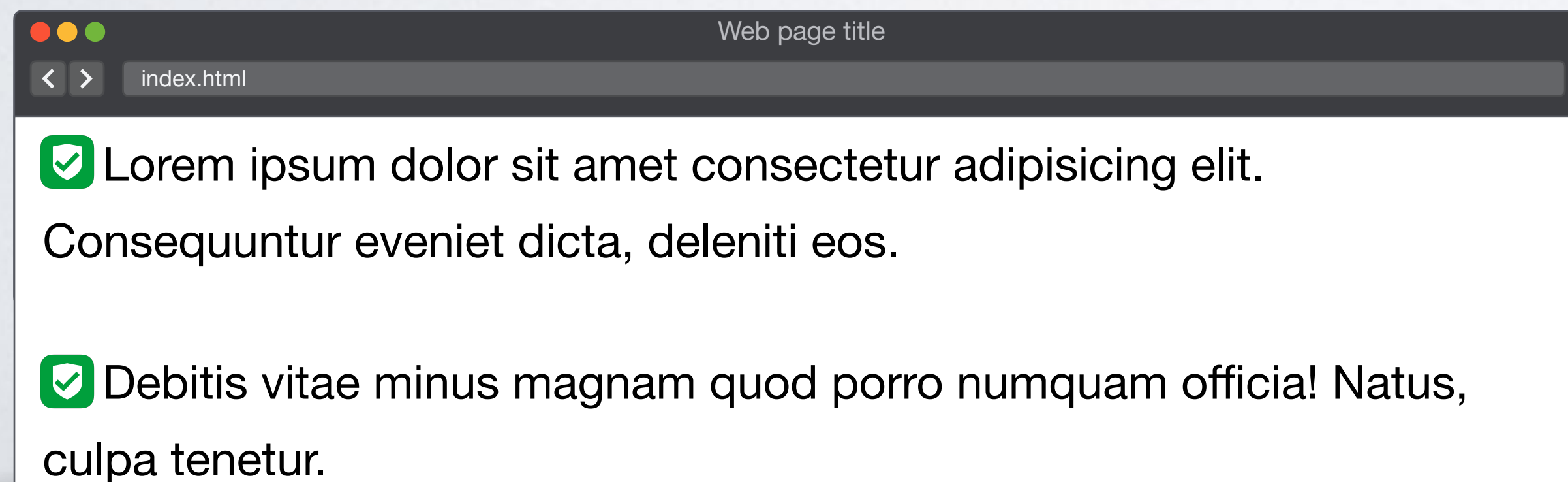
```
<body>  
<p>Lorem ipsum dolor sit amet consectetur  
adipisicing elit. Consequuntur eveniet dicta,  
deleniti eos.</p>
```

```
<p>Debitis vitae minus magnam quod porro  
numquam officia! Natus, culpa tenetur.</p>  
</body>
```

CSS

```
p::before { content: url(image.png); }
```

Browser



Pseudo-elements

::first-letter

::first-line

::before

::after

PSEUDO-ELEMENTS / AFTER

Describes generated content after an element's content.

Syntax `selector::after {style properties}`

`p::after {content: "✓"}`

With this code all paragraphs are finished with ✓.

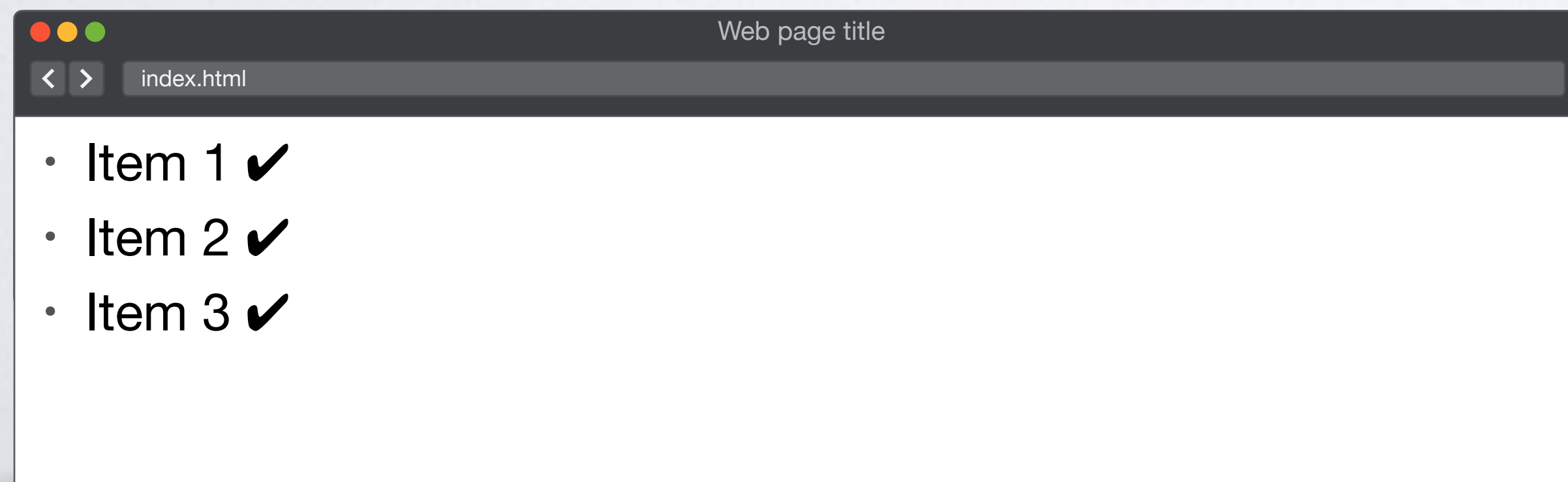
HTML

```
<body>
<ul>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
</ul>
</body>
```

CSS

```
li::after { content: "✓"; }
```

Browser



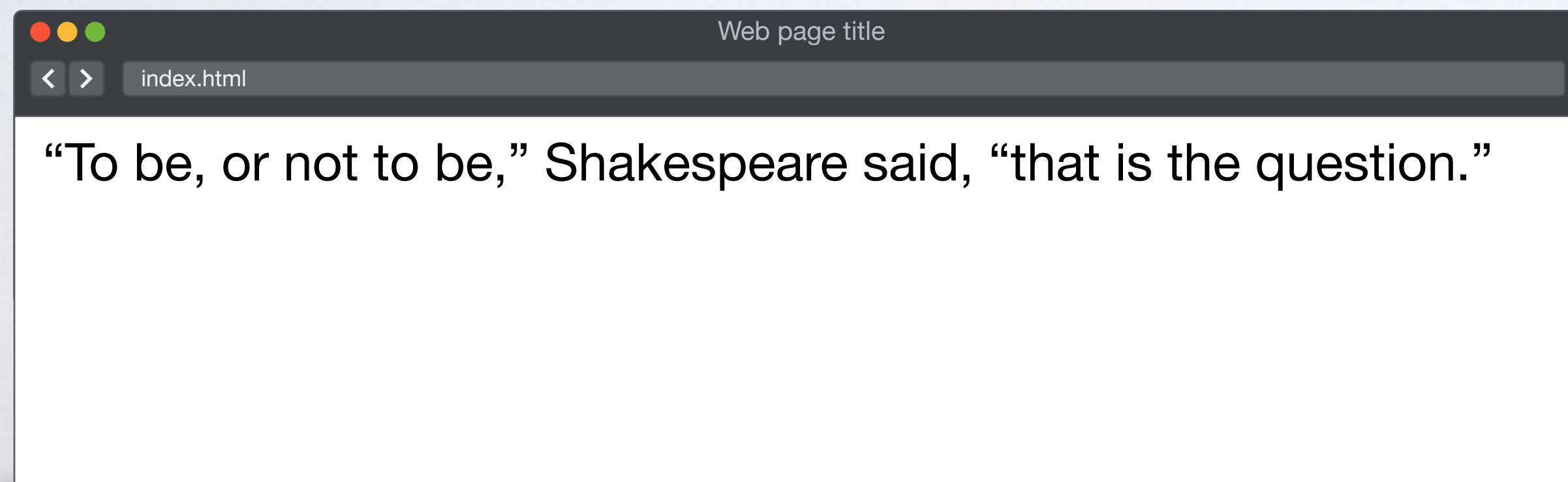
HTML

```
<body>  
<q>To be, or not to be,</q> Shakespeare  
said, <q>that is the question.</q>  
</body>
```

CSS

```
q::before { content: "“"; }  
  
q::after { content: "”"; }
```

Browser



W3C Recommendation

TABLE OF CONTENTS

1. Introduction

1.1. Dependencies

1.2. Terminology

1.3. Changes from CSS2

2. Selectors

3. Case sensitivity

4. Selector syntax

5. Groups of selectors

6. Simple selectors

6.1. Type selector

6.1.1. Type selectors and namespaces

6.2. Universal selector

6.2.1. Universal selector and namespaces

6.3. Attribute selectors

6.3.1. Attribute presence and value selectors

6.3.2. Substring matching attribute selectors

6.3.3. Attribute selectors and namespaces

6.3.4. Default attribute values in DTDs

6.4. Class selectors

6.5. ID selectors

6.6. Pseudo-classes

6.6.1. Dynamic pseudo-classes

6.6.1.1. The link pseudo-classes: :link and :visited


6.6.1.2. The user action pseudo-classes :hover, :active, and :focus

6.6.2. The target pseudo-class :target

6.6.3. The language pseudo-class :lang

Selectors Level 3

W3C Recommendation 06 November 2018



This version:
<https://www.w3.org/TR/2018/REC-selectors-3-20181106/>

Latest version:
<https://www.w3.org/TR/selectors-3/>

Previous version:
<https://www.w3.org/TR/2018/PR-selectors-3-20180911/>

Latest version of Selectors:
<https://www.w3.org/TR/selectors/>

Editor's Draft
<https://drafts.csswg.org/selectors-3/>

Feedback:
File an [issue on GitHub](#)

Editors:
[Tantek Çelik](#) (Invited Expert)
[Elika J. Etemad](#) (Invited Expert)
Daniel Glazman (Disruptive Innovations SARL)
[Ian Hickson](#) (Google)
Peter Linss (former editor, [Netscape/AOL](#))
John Williams (former editor, [Quark, Inc.](#))

Please check the [errata](#) for any errors or issues reported since publication.

Copyright © 2018 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

Selectors are patterns that match against elements in a tree, and as such form one of several technologies that

SOURCE: [Selectors Level 3 by W3C](#).

CSS

CSS FUNDAMENTALS

Pseudo-element



IN A ROCKET

Learn front-end development at *rocket speed*