

HTML. Unit 7. Advanced text formatting.

[1. Introduction](#)

[2. Abbreviations](#)

[2.1. Proposed exercise: Abbreviations example](#)

[2.2. Proposed exercise: Abbreviations for chatting](#)

[3. Citations](#)

[3.1. The <q> element](#)

[3.1.1. Proposed exercise: Inline quotations](#)

[3.2. The <blockquote> element](#)

[3.2.1. Proposed exercise: Famous quotes](#)

[3.3. The <q> + <blockquote> + <cite> elements](#)

[3.3.1. Proposed exercise: Full quotations example](#)

[4. Representing computer code](#)

[4.1. The <code> element](#)

[4.1.1. Proposed exercise: Inline code](#)

[4.2. The <pre> + <code> elements](#)

[4.2.1. Proposed exercise: Block code](#)

[4.3. The <var> element](#)

[4.3.1. Proposed exercise: Equations and variables](#)

[4.4. The <kbd> element](#)

[4.4.1. Proposed exercise: Keyboard shortcuts](#)

[4.5. The <samp> element](#)

[4.5.1. Proposed exercise: Sample output](#)

[4.6. A full example](#)

[4.6.1. Proposed exercise: Code, shortcuts and output](#)

[4.6.2. Proposed exercise: Linux commands](#)

[5. Marking up contact details](#)

[5.1. Proposed exercise: Contact information](#)

[6. Figures](#)

[6.1. Figures with images](#)

[6.1.1. Proposed exercise: Images with captions](#)

[6.2. Figures with poems](#)

[6.2.1. Proposed exercise: Poems](#)

[6.3. Figures with code](#)

[6.3.1. Proposed exercise: Block code](#)

[6.4. Figures with quotations](#)

[6.4.1. Proposed exercise: Famous quotes](#)

1. Introduction

There are many elements in HTML for formatting text, which we didn't get to in previously. The elements described in this unit are less known, but still useful to know about. Here you'll learn about marking up abbreviations, citations, quotations, computer code, mathematical equations and contact information.

2. Abbreviations

The **HTML Abbreviation element** (`<abbr>`) represents an abbreviation or acronym. The optional `title` attribute can provide an expansion or human-readable description for the abbreviation. This text is often presented by browsers as a tooltip when the mouse cursor is hovered over the element. If present, `title` must contain the full description and nothing else. Let's look at some examples where we may use abbreviations or acronyms:

We use HTML to structure our web documents.
You can use CSS to style your HTML.
NASA sure does some exciting work.
Ashok's joke made me LOL big time.

And now let's see the HTML code we should write to get that result:

```
1. <p>We use <abbr title="Hypertext Markup Language">HTML</abbr> to structure our  
   web documents.</p>  
2.  
3. <p>You can use <abbr title="Cascading Style Sheets">CSS</abbr> to style your  
   <abbr title="HyperText Markup Language">HTML</abbr>.</p>  
4.  
5. <p><abbr title="National Aeronautics and Space Administration">NASA</abbr> sure  
   does some exciting work.</p>  
6.  
7. <p>Ashok's joke made me <abbr title="Laugh Out Loud">LOL</abbr> big time.</p>
```

The purpose of this element is purely for the convenience of the author and all browsers display it inline by default, though its default styling varies from one browser to another (this can be adjusted by adding some CSS code):

- Some browsers, like Internet Explorer, do not style it differently than a `` element.
- Opera, Firefox, and some others add a dotted underline to the content of the element.
- A few browsers not only add a dotted underline, but also put it in small caps.

2.1. Proposed exercise: Abbreviations example

Create a web page with the code of the previous example, add a couple of paragraphs with some abbreviations, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

2.2. Proposed exercise: Abbreviations for chatting

The world of email, texting and instant messaging has given rise to a whole series of acronyms or abbreviations that allow texters to complete their messages more quickly. In this exercise you must write a couple of unordered lists. The first one will contain the list of the abbreviations listed below (and some other you may know). The second list will contain at least twenty sentences showing how those abbreviations may be used for chatting.

— Also use `<h1>` and `<h2>` headers to precede the lists and show some descriptive text.

For example, the first list may show the abbreviations like this:

- AFK – Away From Keyboard
- BBIAB – Be Back In A Bit
- BBL – Be Back Later
- BBS – Be Back Soon
- BRB – Be Right Back
- BFF – Best Friends Forever
- BTW – By The Way
- COB – Close Of Business
- DIY – Do It Yourself
- DM – Direct Message

- ETA – Estimated Time Of Arrival
- FISH – First In, Still Here
- IDK – I Don't Know
- IMO – In My Opinion
- IRL – In Real Life
- LMK – Let Me Know
- LML – Laughing Mad Loud
- LOL – Laughing Out Loud
- NYOB – None of Your Business
- OFC – Of Course
- OMG – Oh My God
- PANS – Pretty Awesome New Stuff
- POS – Parents Over Shoulder
- ROFL – Rolling On the Floor Laughing
- SMH – Shaking My Head
- TTYL – Talk To You Later
- YOLO – You Only Live Once
- WTH – What The Heck

The second list you have to show must contain at least twenty sentences where those abbreviations are used. For example:

- This is my BFF, we have known each other since kindergarten.
- BTW, the boss needs to see you in her office in five minutes.
- She bought an old house that needs work because she loves DIY.
- If you need more information just DM me.
- What is the ETA on that report you are writing?
- LOL. That is the funniest message I have ever received.
- Did you see the game last night? I was shocked. I can't believe the Lakers lost to the Phoenix Suns.
- SMH.
- ...

You may follow this source code as an example telling you how to markup both lists:

```
1. <h1>Some abbreviations I use to chat</h1>
```

```
2. <h2>The abbreviations</h2>
3. <ul>
4.   <li>AFK - Away From Keyboard</li>
5.   <li>BBIAB - Be Back In A Bit</li>
6.   <li>BBL - Be Back Later</li>
7.   <li>BBS - Be Back Soon</li>
8.   <li>...</li>
9. </ul>
10. <h2>Some examples</h2>
11. <ul>
12.   <li>This is my <abbr title="Best Friends Forever">BFF</abbr>, we have known
    each other since kindergarten.</li>
13.   <li><abbr title="By The Way">BTW</abbr>, the boss needs to see you in her
    office in five minutes.</li>
14.   <li>She bought an old house that needs work because she loves <abbr
    title="Do It Yourself">DIY</abbr>.</li>
15.   <li>If you need more information just <abbr title="Direct
    Message">DM</abbr> me.</li>
16.   <li>What is the <abbr title="Estimated Time of Arrival">ETA</abbr> on that
    report you are writing?</li>
17.   <li><abbr title="Laughing Out Loud">LOL</abbr>. That is the funniest
    message I have ever received.</li>
18.   <li>Did you see the game last night? I was shocked. I can't believe the
    Lakers lost to the Phoenix Suns. <abbr title="Shaking My Head">SMH</abbr>.</li>
19.   <li>...</li>
20. </ul>
```

3. Citations

3.1. The <q> element

The HTML `<q>` element indicates that the enclosed text is a short inline quotation. Most modern browsers implement this by surrounding the text in quotation marks. You may also use the attribute `cite` to provide a URL that designates the source document or message for the information quoted, although this URL is not shown by the browser by default.

Let's see a couple of examples by having a look at the result and the corresponding source code:

When Dave asks HAL to open the pod bay door, HAL answers: «I'm sorry, Dave. I'm afraid I can't do that».

According to Mozilla's website, «Firefox 1.0 was released in 2004 and became a big success».

```
1. <p>
2.     When Dave asks HAL to open the pod bay door, HAL answers: <q
   cite="https://www.imdb.com/title/tt0062622/quotes/qt0396921">I'm sorry, Dave.
   I'm afraid I can't do that</q>.
3. </p>
4.
5. <p>
6.     According to Mozilla's website,
7.     <q cite="https://www.mozilla.org/en-US/about/history/details/">Firefox 1.0
   was released in 2004 and became a big success</q>.
8. </p>
```

Important: The `<q>` element is intended for short quotations that don't require paragraph breaks. For long quotations use the `<blockquote>` element.

3.1.1. Proposed exercise: Inline quotations

Create a web page with the code of the previous example, add one more paragraph with any quotation you like and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

— You can get a quote for example from your preferred film, as in
«<https://www.starwars.com/news/15-star-wars-quotes-to-use-in-everyday-life>».

3.2. The `<blockquote>` element

The HTML `<blockquote>` element (or *HTML Block Quotation Element*) indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation. A URL for the source of the quotation may be given using the `cite` attribute, while a text representation of the source can be given using the `<cite>` element. For example:

Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do. If you haven't found it yet, keep looking. Don't settle. As with all matters of the heart, you'll know when you find it.

— [Steve Jobs](https://www.brainyquote.com/quotes/steve_jobs_416859)

1. `<blockquote cite="https://www.brainyquote.com/quotes/steve_jobs_416859">`
2. Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do. If you haven't found it yet, keep looking. Don't settle. As with all matters of the heart, you'll know when you find it.
3. `<cite>Steve Jobs</cite>`
4. `</blockquote>`

3.2.1. Proposed exercise: Famous quotes

Create a web page pointing to at least twenty of the most famous quotes. You may find some of them at «<https://www.brainyquote.com>», «<https://www.azquotes.com>/», or you can use any quotes you know, or from any other site.

— You can use `<blockquote>` to wrap the full quote, and `<cite>` inside each quote to wrap the author's name, as in the previous and following examples:

The greatest glory in living lies not in never falling, but in rising every time we fall.

— [Nelson Mandela](#)

The way to get started is to quit talking and begin doing.

— [Walt Disney](#)

Your time is limited, so don't waste it living someone else's life. Don't be trapped by dogma – which is living with the results of other people's thinking. Don't let the

noise of others' opinions drown out your own inner voice. And most important, have the courage to follow your heart and intuition.

— [Steve Jobs](#)

If life were predictable it would cease to be life, and be without flavor.

— [Eleanor Roosevelt](#)

If you look at what you have in life, you'll always have more. If you look at what you don't have in life, you'll never have enough.

— [Oprah Winfrey](#)

...

3.3. The <q> + <blockquote> + <cite> elements

The **HTML Citation element** (`<cite>`) is used to describe a reference to a cited creative work, and should include the title of that work. The reference may be in an abbreviated form according to context-appropriate conventions related to citation metadata. Let's see an example where this element can be useful:

Hello and welcome to my motivation page. As [Confucius' quotes site](#) says:

It does not matter how slowly you go as long as you do not stop.

I also love the concept of positive thinking, and the need to keep your thoughts positive (as mentioned in [Positive quotes.](#))

And now let's have a look at the source code we should write to get that result:

1. `<h1>Motivation page</h1>`
2. `<p>Hello and welcome to my motivation page. As <a`


```

3. href="http://www.brainyquote.com/quotes/authors/c/confucius.html"><cite>Confucius
quotes site</cite></a> says:</p>
4. <blockquote cite="http://www.brainyquote.com/quotes/authors/c/confucius.html">
5.     <p>It does not matter how slowly you go as long as you do not stop.</p>
6. </blockquote>
7.
8. <p>I also love the concept of positive thinking, and the need to <q
cite="https://www.azquotes.com/quotes/topics/positive.html">keep your thoughts
positive</q> (as mentioned in <a
href="https://www.azquotes.com/quotes/topics/positive.html"><cite>Positive
quotes</cite></a>.)</p>

```

3.3.1. Proposed exercise: Full quotations example

Create a web page with the code of the previous example, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

4. Representing computer code

There are a number of elements available for marking up computer code using HTML:

- `<code>` : For marking up generic pieces of computer code.
- `<pre>` : For retaining whitespace (generally code blocks). If you use indentation or excess whitespace inside your text, browsers will ignore it and you will not see it on your rendered page. As explained in a previous unit, if you wrap the text in `<pre></pre>` tags however, your whitespace will be rendered identically to how you see it in your text editor.
- `<var>` : For specifically marking up variable names.
- `<kbd>` : For marking up keyboard (and other types of) input entered into the computer.
- `<samp>` : For marking up the output of a computer program.

4.1. The `<code>` element

The HTML `<code>` element displays its contents styled in a fashion intended to indicate that the text is a short fragment of computer code. By default, the content text is displayed using the default monospace font. For example:

The `push()` method adds one or more elements to the end of an array and returns the new length of the array.

The function `selectAll()` highlights all the text in the input field so the user can, for example, copy or delete the text.

```
1.  <p>
2.      The <code>push()</code> method adds one or more elements to the end of an
   array and returns the new length of the array.
3.  </p>
4.
5.  <p>
6.      The function <code>selectAll()</code> highlights all the text in the input
   field so the user can, for example, copy or delete the text.
7.  </p>
```

4.1.1. Proposed exercise: Inline code

Create a web page with the code of the previous example, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

4.2. The `<pre>` + `<code>` elements

The `<code>` element by itself only represents a single phrase of code or line of code. To represent multiple lines of code, we can wrap the `<code>` element within a `<pre>` element. For example:

```
if (a > b) {           // Example of code in
  console.log('Hello!'); // the JavaScript language
}
```

```
1. <pre><code>
2.   if (a > b) {           // Example of code in
3.     console.log('Hello!'); // the JavaScript language
4.   }
5. </code></pre>
```

4.2.1. Proposed exercise: Block code

Create a web page with the code of the previous example, add a new block with any piece of code of any language, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

4.3. The <var> element

The HTML **Variable element** (<var>) represents the name of a variable in a mathematical expression or a programming context. It's typically presented using an italic version of the current typeface, although that behavior is browser-dependent. For example:

A simple equation: $x = y + 2$

The volume of a box is $l \times w \times h$, where l represents the length, w the width and h the height of the box.

The variables *minSpeed* and *maxSpeed* control the minimum and maximum speed of the apparatus in revolutions per minute (RPM).

```
1. <p>
2.   A simple equation: <var>x</var> = <var>y</var> + 2
```

```
3. </p>
4.
5. <p>
6.     The volume of a box is <var>l</var> × <var>w</var> × <var>h</var>, where
   <var>l</var> represents the length, <var>w</var> the width and <var>h</var> the
   height of the box.
7. </p>
8.
9. <p>
10.    The variables <var>minSpeed</var> and <var>maxSpeed</var> control the
    minimum and maximum speed of the apparatus in revolutions per minute (RPM).
11. </p>
```

4.3.1. Proposed exercise: Equations and variables

Create a web page with the code of the previous example, add a new paragraph with any equation you like and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

4.4. The <kbd> element

The **HTML Keyboard Input element** (`<kbd>`) represents a span of inline text denoting textual user input from a keyboard, voice input, or any other text entry device. By convention, the browser renders the contents of a `<kbd>` element using its default monospace font. For example:

Please press `Ctrl + Shift + R` to re-render an MDN page.

Use the command `help mycommand` to view documentation for the command «mycommand».

You can create a new document using the keyboard shortcut `Ctrl + N`.

```
1. <p>
2.     Please press <kbd>Ctrl</kbd> + <kbd>Shift</kbd> + <kbd>R</kbd> to re-render
   an MDN page.
3. </p>
```

```
4.
5.  <p>
6.      Use the command <kbd>help mycommand</kbd> to view documentation for the
      command "mycommand".
7.  </p>
8.
9.  <p>
10.     You can create a new document using the keyboard shortcut <kbd>Ctrl</kbd> +
      <kbd>N</kbd>.
11. </p>
```

4.4.1. Proposed exercise: Keyboard shortcuts

Create a web page with the code of the previous example, add a couple of paragraphs with any shortcuts you know, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

4.5. The <samp> element

The **HTML Sample Element** (`<samp>`) is used to enclose inline text which represents sample (or quoted) output from a computer program. Its contents are typically rendered using the browser's default monospaced font (such as Courier or Lucida Console). For example:

I was trying to boot my computer, but I got this hilarious message:

```
Keyboard not found
Press F1 to continue
```

...

When the process is complete, the utility will output the text Scan complete. Found *N* results. You can then proceed to the next step.

```
1.  <p>I was trying to boot my computer, but I got this hilarious message:</p>
```

```
2. <p><samp>Keyboard not found <br>Press F1 to continue</samp></p>
3.
4. ...
5.
6. <p>When the process is complete, the utility will output the text <samp>Scan
   complete. Found <em>N</em> results.</samp> You can then proceed to the next
   step.</p>
```

4.5.1. Proposed exercise: Sample output

Create a web page with the code of the previous example, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

4.6. A full example

Let's have a look at the following example which uses all these elements (find below the corresponding source code):

```
var para = document.querySelector('p');

para.onclick = function() {
  alert('Owww, stop poking me!');
}
```

You shouldn't use presentational elements like `` and `<center>`.

In the above JavaScript example, *para* represents a paragraph element.

Select all the text with `Ctrl/Cmd + A`.

```
$ ping mozilla.org
PING mozilla.org (63.245.215.20): 56 data bytes
```

```
64 bytes from 63.245.215.20: icmp_seq=0 ttl=40 time=158.233 ms
```

```

1.  <pre><code>var para = document.querySelector('p');
2.
3.  para.onclick = function() {
4.      alert('Owww, stop poking me!');
5.  }</code></pre>
6.
7.  <p>You shouldn't use presentational elements like <code>&lt;font&gt;</code> and
   <code>&lt;center&gt;</code>.</p>
8.
9.  <p>In the above JavaScript example, <var>para</var> represents a paragraph
   element.</p>
10.
11. <p>Select all the text with <kbd>Ctrl</kbd>/<kbd>Cmd</kbd> + <kbd>A</kbd>.</p>
12.
13. <pre>$ <kbd>ping mozilla.org</kbd>
14. <samp>PING mozilla.org (63.245.215.20): 56 data bytes
15. 64 bytes from 63.245.215.20: icmp_seq=0 ttl=40 time=158.233 ms</samp></pre>

```

4.6.1. Proposed exercise: Code, shortcuts and output

Create a web page with the code of the previous example, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

4.6.2. Proposed exercise: Linux commands

Create a web page to display a table with some of the most important linux commands, and their usage description, more or less like the table below.

— You have to use the `<code>` tag for the commands in the left column, and the `<kbd>` tag for the examples of user's input. Also notice that the table has headers and caption.

Linux commands

Commands	Description
<code>passwd</code>	Changes your user password: 1. Type your old password 2. Insert new password 3. Confirm new password
<code>~</code>	User's home directory (short-cut for: <code>/home/username</code>)
<code>ls</code>	Lists folders and files in current directory
<code>mkdir</code>	Create new directory into the current one: <code>mkdir newdir</code>
<code>cd</code>	Changes directory: <code>cd test</code> (go to a directory named 'test') <code>cd ..</code> (go to parent directory) <code>cd ~</code> (go to home directory)
<code>rm</code>	Removes specified file or directory: <code>rm filename</code> (remove single <i>filename</i>) <code>rm *.txt</code> (remove ALL .txt files into current directory) <code>rm -r dirname</code> (remove directory and its files) Please be careful when you use the <code>-f</code> option!
<code>rmdir</code>	Removes specified EMPTY directory <code>rmdir dirname</code>
<code>pwd</code>	Prints current absolute path
<code>man</code>	Shows specified command's manual page: <code>man ls</code> (shows <code>ls</code> help)
<code>vi x.sh</code>	VI is a <i>text editor</i> . If <code>x.sh</code> does not exists, <code>vi</code> creates a new file called <code>x.sh</code> and opens it; otherwise, it just opens the existing file.

<code>less textfile</code>	<code>less</code> is a <i>text pager</i> . Opens (read only) <i>textfile</i> file. You can use up and down arrows to move across the text, shares many commands with VI.
<code>chmod</code>	<p>Changes POSIX permissions of a file or directory. Allows to protect files from unwanted access:</p> <p>r : read permission w : write permission x : exec permission</p> <p><code>chmod +x file.sh</code> (allows execution) <code>chmod -w file.sh</code> (denies write)</p>
<code>chown</code>	<p>Changes owner of a file or directory:</p> <p><code>chown username file.sh</code></p>
<code>top</code>	Shows current executing processes
<code>cat</code>	Prints the content of a file on screen
<code>grep</code>	<p>Filters specified text file, and shows the lines that contains the pattern:</p> <p><code>grep pattern file.sh</code></p> <p>You can use also pipe the output of another command:</p> <p><code>cat file.sh grep home</code> <code>cat file.sh grep "home page"</code></p>

5. Marking up contact details

HTML has an element for marking up contact details (`<address>`) This simply wraps around your contact details, for example:

1. `<address>`
2. `<p>Chris Mills, Manchester, The Grim North, UK</p>`

```
3. </address>
```

It could also include more complex markup, and other forms of contact information, for example:

```
1. <address>
2.   <p>
3.     Chris Mills<br />
4.     Manchester<br />
5.     The Grim North<br />
6.     UK
7.   </p>
8.
9.   <ul>
10.    <li>Tel: 01234 567 890</li>
11.    <li>Email: me@grim-north.co.uk</li>
12.  </ul>
13. </address>
```

Note that something like this would also be ok, if the linked page contained the contact information:

```
1. <address>
2.   You can contact author at <a
3. href="http://www.somedomain.com/contact">www.somedomain.com</a>.<br />
4.   If you see any bugs, please <a
5. href="mailto:webmaster@somedomain.com">contact webmaster</a>.<br />
6.   You may also want to visit us:<br />
7.   Mozilla Foundation<br />
8.   331 E Evelyn Ave<br />
9.   Mountain View, CA 94041<br />
10.  USA
11. </address>
```

5.1. Proposed exercise: Contact information

Create a web page with the code of the previous examples and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

6. Figures

The HTML `<figure>` (Figure With Optional Caption) element represents self-contained content, potentially with an optional caption, which is specified using the (`<figcaption>`) element. The figure, its caption, and its contents are referenced as a single unit.

6.1. Figures with images



Dog asking for some food

```
1.  <!-- Just an image -->
2.  <figure>
3.      
5.  </figure>
6.
7.  <!-- Image with a caption -->
8.  <figure>
9.      
11.      <figcaption>MDN Logo</figcaption>
12.  </figure>
13.
14. <!-- Another image with caption -->
15. <figure>
16.     
18.     <figcaption>Dog asking for some food</figcaption>
19. </figure>
```

6.1.1. Proposed exercise: Images with captions

Create a web page with the code of the previous example, add a couple of figures with pictures, and choose a suitable caption for each one. Finally check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

6.2. Figures with poems

Bid me discourse, I will enchant thine ear,
Or like a fairy trip upon the green,
Or, like a nymph, with long dishevell'd hair,
Dance on the sands, and yet no footing seen:
Love is a spirit all compact of fire,
Not gross to sink, but light, and will aspire.

Venus and Adonis, by William Shakespeare

```
1.  <figure>
2.      <p>
3.          Bid me discourse, I will enchant thine ear,<br />
4.          Or like a fairy trip upon the green,<br />
5.          Or, like a nymph, with long dishevell'd hair,<br />
6.          Dance on the sands, and yet no footing seen:<br />
7.          Love is a spirit all compact of fire,<br />
8.          Not gross to sink, but light, and will aspire.
9.      </p>
10.  <figcaption>
11.      <cite>Venus and Adonis</cite>, by William Shakespeare
12.  </figcaption>
13. </figure>
```

6.2.1. Proposed exercise: Poems

Create a web page with the code of the previous example, add a new figure with another poem you like, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

6.3. Figures with code

Get browser details using navigator :

```
function NavigatorExample() {
    var txt;
    txt = "Browser CodeName: " + navigator.appCodeName + "; ";
    txt+= "Browser Name: " + navigator.appName + "; ";
    txt+= "Browser Version: " + navigator.appVersion + "; ";
    txt+= "Cookies Enabled: " + navigator.cookieEnabled + "; ";
    txt+= "Platform: " + navigator.platform + "; ";
    txt+= "User-agent header: " + navigator.userAgent + "; ";
    console.log("NavigatorExample", txt);
}
```

```
1. <figure>
2. <figcaption>Get browser details using <code>navigator</code></figcaption>
3. <pre><code>function NavigatorExample() {
4.     var txt;
5.     txt = "Browser CodeName: " + navigator.appCodeName + "; ";
6.     txt+= "Browser Name: " + navigator.appName + "; ";
7.     txt+= "Browser Version: " + navigator.appVersion + "; ";
8.     txt+= "Cookies Enabled: " + navigator.cookieEnabled + "; ";
9.     txt+= "Platform: " + navigator.platform + "; ";
10.    txt+= "User-agent header: " + navigator.userAgent + "; ";
11.    console.log("NavigatorExample", txt);
12. }</code></pre>
13. </figure>
```

6.3.1. Proposed exercise: Block code

Create a web page with the code of the previous example, and check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

6.4. Figures with quotations

Edsger Dijkstra:

If debugging is the process of removing software bugs, then programming must be the process of putting them in.

```
1. <figure>
2.   <figcaption><cite>Edsger Dijkstra:</cite></figcaption>
3.   <blockquote>
4.       If debugging is the process of removing software bugs,
5.       then programming must be the process of putting them in.
6.   </blockquote>
7. </figure>
```

6.4.1. Proposed exercise: Famous quotes

Create a web page with the code of the previous example, and add some famous quotes using the same format, to show at least ten quotes. Finally check the results in your browser. Do not forget to include the rest of the basic HTML tags, and validate your code.

— You can use the same quotes you used in a previous exercise, or you can get some new ones from «<https://www.brainyquote.com/>«, «<https://www.azquotes.com/>«, or any other site you may know.

