Block-level elements

Theory

18 minutes reading

Verify to skip | Start practicing

In this topic, we will take a look at web page formatting. You'll find out what block-level HTML elements are needed for, what properties they possess, and how to use them most effectively in creating the structure of HTML code.

1 required topic

✓ HTML page structure — In project

3 dependent topics

Basic syntax — In project

HTML attributes id and class — In project

DOM — In project

§1. What are HTML elements?

As you know, HTML files can be opened in browsers. After receiving an HTML document, the browser reads the tags in it and uses them to create an HTML page that users see on their monitor screens.

All you see on the page in your browser viewer are HTML elements. And this is where the difference between HTML tags and HTML elements lies: elements are what the user sees on the browser page, while tags are what the developer writes when creating an HTML document.

> It is worth remembering that not all HTML tags are elements. For example, the `DOCTYPE` instruction, necessary for correct interpretation of code by a browser, and tags in the header of an HTML document are not elements.

There are two main types of page elements: **block-level** elements and **inline** elements. Both of them have their own peculiarities.

§2. Block-level Elements

Block-level elements are mostly used to create the structure of web pages or logically divide an HTML document into parts. Here are some examples of block elements:

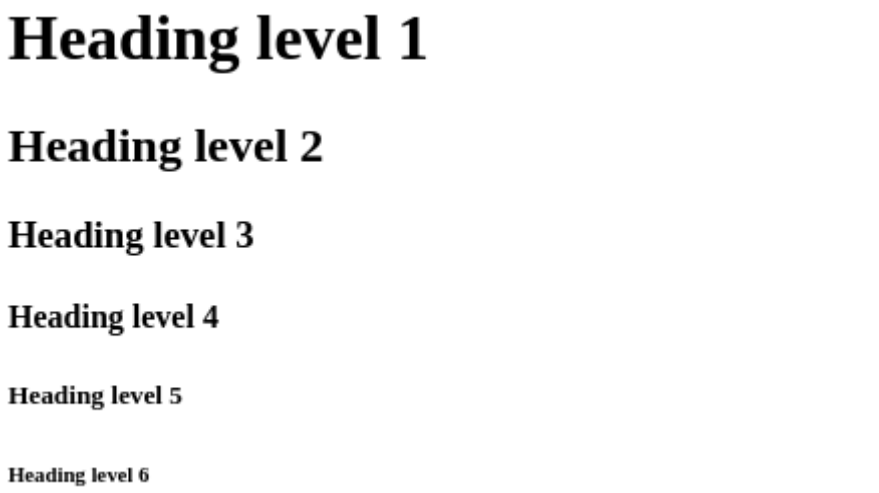
- `<div>` is a universal content divider. It is used to group similar HTML elements:

```
<div>
  <h1>Element h1 is inside the div</h1>
  <p>Element p is also inside the div</p>
</div>
```

- `<h1>-<h6>` are heading tags:

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

The result will be displayed in the browser as follows:



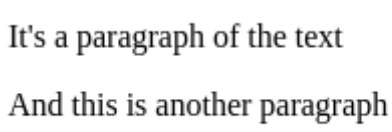
As you can see, these tags make it easy to identify the headings that convey the level of importance of the text.

> It is recommended to use only one `<h1>` tag on a web page – this should be the title showing the main theme of the page.

- `<p>` defines a text paragraph:

```
<p>It's a paragraph of the text</p>
<p>And this is another paragraph</p>
```

This is how it looks in the browser:



The text enclosed in this tag is not highlighted in bold, unlike the case of `<h1>` - `<h6>`.

- `<pre>` defines a block of formatted text:

```
<pre>
.........／>  フ
.........|  _  _ |
.......／` ミ_xノ
......／     |
...../  ヽ   ノ
..../  | | |
..../  | | |
／￣|   | | |
| (￣ヽ__ヽ_)__)
.＼二つ
</pre>
```

By default, any number of spaces occurring in the code in a row are shown as one space on a web page. The `<pre>` tag displays the text the way you want it, with all the spaces inserted between characters:

- `<hr>` creates a horizontal line:

```
<hr>
```

Now let's see how it's displayed in the browser:

Table of contents:

↑ Block-level elements

This is by far not a complete list of block-level elements – there are definitely many more to know.

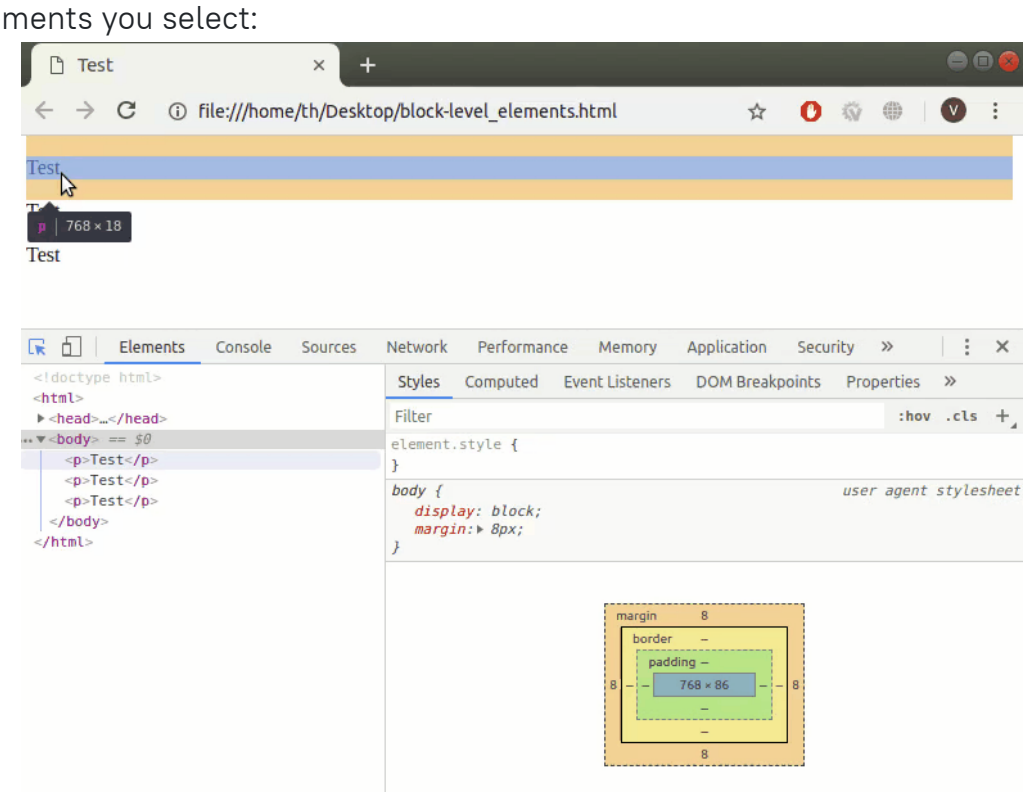
§3. Features of block-level elements

The following features are characteristic of block-level elements:

1. They can contain both inline elements and other block-level elements.
2. Browsers display them with a line break before and after the element. Take a look at the behavior of block-level elements and compare it with that of inline elements to better understand this feature:

3. Block-level elements take up the full width available. That is, if the element is located inside `<body>`, it will occupy the entire width of the web page. If the block-level element is inside another block-level element, it will use the entire width of the parent element.

   To verify it, just press the *Ctrl+Shift+I* or *Cmd+Opt+I* keys on a web page and point the mouse cursor at some object. In the browser, rectangles of different height and width will be highlighted in color. This is the area occupied by the elements you select:



4. The height of a block-level element is calculated automatically by the browser based on the block contents.

Block level elements behave like that because they act like containers that keep all their content within.

§4. Conclusion

We have discussed the basic features of block-level HTML elements. It is noteworthy that the distinction of block-level and inline elements was used up to HTML version 4.0. In HTML 5, these concepts were replaced by a more complex set of content categories: now, each HTML element must follow the rules that

§1. What are HTML elements

§2. Block level Elements

§3. Features of block level elements

§4. Conclusion

Discussion

determine what information can be stored in it. We shall take a closer look at that a bit later. Now, let's review what we've learned so far with a bit of practice, as this knowledge is sure to come in handy sometime.

Report a typo

789 users liked this piece of theory. 12 didn't like it. **What about you?**

Start practicing

Verify to skip

Comments (8) Hints (0) Useful links (0) Show discussion