JavaScript Day - 12

DOM Manipulation

innerHTML: the contents

The innerHTML property allows to get the HTML inside the element as a string.

We can also modify it. So it's one of the most powerful ways to change the page.

```
<body>
  <p>A paragraph</p>
  <div>A div</div>

  <script>
    alert( document.body.innerHTML ); // read the current contents
    document.body.innerHTML = 'The new BODY!'; // replace it
  </script>

</body>
```

outerHTML: full HTML of the element

The outerHTML property contains the full HTML of the element. That's like innerHTML plus the element itself.

```
<div id="elem">Hello <b>World</b></div>

<script>
  alert(elem.outerHTML); // <div id="elem">Hello <b>World</b></div>
</script>
```

textContent: pure text

The textContent provides access to the text inside the element: only text, minus all <tags>.

```
1  <div id="news">
2    <h1>Headline!</h1>
3    <p>Martians attack people!</p>
4  </div>
5
6  <script>
7    // Headline! Martians attack people!
8    alert(news.textContent);
9  </script>
```

Writing to textContent is much more useful, because it allows to write text the “safe way”.

Let’s say we have an arbitrary string, for instance entered by a user, and want to show it.

- With innerHTML we’ll have it inserted “as HTML”, with all HTML tags.
- With textContent we’ll have it inserted “as text”, all symbols are treated literally.

The “hidden” property

The “hidden” attribute and the DOM property specifies whether the element is visible or not.

We can use it in HTML or assign it using JavaScript, like this:

```
<div>Both divs below are hidden</div>

<div hidden>With the attribute "hidden"</div>

<div id="elem">JavaScript assigned the property "hidden"</div>

<script>
  elem.hidden = true;
</script>
```

Technically, hidden works the same as style="display:none". But it’s shorter to write.

More Properties

DOM elements also have additional properties, in particular those that depend on the class:

1. Input Element - value
2. Anchor Element - href
3. All elements - id

and much more...

HTML attributes

In HTML, tags may have attributes. When the browser parses the HTML to create DOM objects for tags, it recognizes standard attributes and creates DOM properties from them.

So when an element has id or another standard attribute, the corresponding property gets created. But that doesn't happen if the attribute is non-standard.

HTML attributes

```
1  <body id="test" something="non-standard">
2    <script>
3      alert(document.body.id); // test
4      // non-standard attribute does not yield a property
5      alert(document.body.something); // undefined
6    </script>
7  </body>
```

Please note that a standard attribute for one element can be unknown for another one. For instance, "type" is standard for <input>, but not for <body>

HTML attributes

So, if an attribute is non-standard, there won't be a DOM-property for it. Is there a way to access such attributes?

Sure. All attributes are accessible by using the following methods:

- elem.hasAttribute(name) – checks for existence.
- elem.getAttribute(name) – gets the value.
- elem.setAttribute(name, value) – sets the value.
- elem.removeAttribute(name) – removes the attribute.