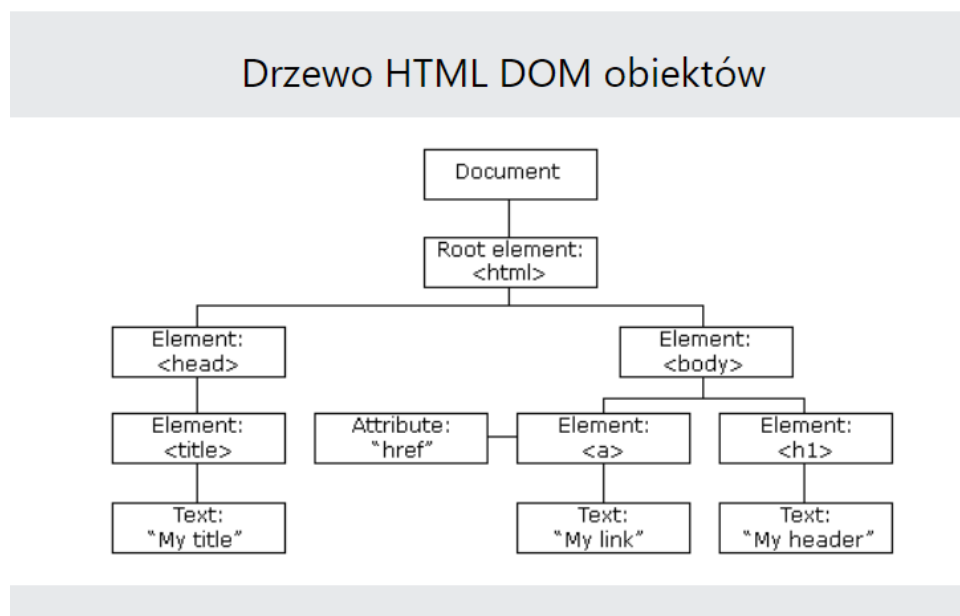


## HTML DOM (obiektowy model dokumentu)

Po załadowaniu strony internetowej przeglądarka tworzy model **objektu dokumentu strony**.

Model **HTML DOM** jest zbudowany jako drzewo **obiektów** :



Dzięki modelowi obiekтовemu JavaScript uzyskuje całą moc potrzebną do tworzenia dynamicznego HTML:

- JavaScript może zmienić wszystkie elementy HTML na stronie
- JavaScript może zmienić wszystkie atrybuty HTML na stronie
- JavaScript może zmienić wszystkie style CSS na stronie
- JavaScript może usuwać istniejące elementy i atrybuty HTML
- JavaScript może dodawać nowe elementy i atrybuty HTML
- JavaScript może reagować na wszystkie istniejące zdarzenia HTML na stronie
- JavaScript może tworzyć nowe zdarzenia HTML na stronie

## Co to jest DOM?

DOM jest standardem W3C (World Wide Web Consortium).

DOM definiuje standard dostępu do dokumentów:

*„Document Object Model (DOM) W3C to platforma i neutralny językowo interfejs, który umożliwia programom i skryptom dynamiczny dostęp i aktualizację treści, struktury i stylu dokumentu”.*

Standard W3C DOM jest podzielony na 3 różne części:

- Core DOM - standardowy model dla wszystkich typów dokumentów
- XML DOM - standardowy model dokumentów XML
- HTML DOM - standardowy model dokumentów HTML

## Co to jest DOM HTML?

HTML DOM to standardowy model **obiektowy** i **interfejs programistyczny** dla HTML. określa:

- Elementy HTML jako **obiekty**
- Właściwości wszystkich elementów **HTML**
- Metody dostępu do wszystkich elementów **HTML**
- Zdarzenia **dla** wszystkich elementów HTML

Innymi słowy: **DOM HTML to standard pobierania, zmiany, dodawania lub usuwania elementów HTML.**

## Metody DOM HTML

Metody HTML DOM to **akcje**, które możesz wykonać (na elementach HTML).

Właściwości HTML DOM to **wartości** (elementów HTML), które możesz ustawić lub zmienić.

## Interfejs programowania DOM

W DOM wszystkie elementy HTML są zdefiniowane jako **obiekty** .

Interfejs programistyczny to właściwości i metody każdego obiektu.

Właściwość to wartość , którą można uzyskać lub ustawić (jak zmiana zawartości elementu HTML).

Metoda to czynność , którą możesz wykonać (na przykład dodanie lub usunięcie elementu HTML).

## Przykład

Poniższy przykład zmienia zawartość (the **innerHTML**) elementu `<p>` za pomocą `id="demo"`:

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

W powyższym przykładzie `getElementById` jest to **metoda** , podczas gdy `innerHTML` jest **właściwością** .

Spróbuj sam:

[https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_dom\\_method](https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_method)

Właściwość `innerHTML`

Najłatwiejszym sposobem uzyskania zawartości elementu jest użycie właściwości `innerHTML`.

Właściwość `innerHTML` jest przydatna do pobierania lub zastępowania zawartości elementów HTML.

Właściwość `innerHTML` może służyć do pobierania lub zmiany dowolnego elementu HTML, w tym `<html>` i `<body>`.

## Dokument JavaScript HTML DOM

### Obiekt dokumentu HTML DOM

Obiekt `document` reprezentuje twoją stronę internetową.

Jeśli chcesz uzyskać dostęp do dowolnego elementu na stronie HTML, zawsze zaczynaj od uzyskania dostępu do obiektu dokumentu.

Poniżej znajduje się kilka przykładów wykorzystania obiektu `document` do uzyskiwania dostępu do kodu HTML i manipulowania nim.

## Znajdowanie i zmiana elementów HTML

### Znajdowanie elementów HTML

Method	Description
<code>document.getElementById(<i>id</i>)</code>	Find an element by element id
<code>document.getElementsByTagName(<i>name</i>)</code>	Find elements by tag name
<code>document.getElementsByClassName(<i>name</i>)</code>	Find elements by class name

### Zmiana elementów HTML

Property	Description
<code>element.innerHTML = new html content</code>	Change the inner HTML of an element
<code>element.attribute = new value</code>	Change the attribute value of an HTML element
<code>element.style.property = new style</code>	Change the style of an HTML element
Method	Description
<code>element.setAttribute(<i>attribute</i>, <i>value</i>)</code>	Change the attribute value of an HTML element

## Dodawanie i usuwanie elementów

### Dodawanie i usuwanie elementów

Method	Description
<code>document.createElement(<i>element</i>)</code>	Create an HTML element
<code>document.removeChild(<i>element</i>)</code>	Remove an HTML element
<code>document.appendChild(<i>element</i>)</code>	Add an HTML element
<code>document.replaceChild(<i>new</i>, <i>old</i>)</code>	Replace an HTML element
<code>document.write(<i>text</i>)</code>	Write into the HTML output stream

## Obsługa zdarzeń

### Dodawanie obsługi zdarzeń

Method	Description
<code>document.getElementById(<i>id</i>).onclick = function(){<i>code</i>}</code>	Adding event handler code to an onclick event

## Znajdowanie obiektów HTML

Pierwszy HTML DOM Level 1 (1998) zdefiniował 11 obiektów HTML, kolekcji obiektów i właściwości. Są one nadal ważne w HTML5.

Później, w HTML DOM Level 3, dodano więcej obiektów, kolekcji i właściwości.

Property	Description	DOM
document.anchors	Returns all <a> elements that have a name attribute	1
document.applets	Deprecated	1
document.baseURI	Returns the absolute base URI of the document	3
document.body	Returns the <body> element	1
document.cookie	Returns the document's cookie	1
document.doctype	Returns the document's doctype	3
document.documentElement	Returns the <html> element	3
document.documentMode	Returns the mode used by the browser	3
document.documentURI	Returns the URI of the document	3
document.domain	Returns the domain name of the document server	1
document.domConfig	Obsolete.	3
document.embeds	Returns all <embed> elements	3
document.forms	Returns all <form> elements	1
document.head	Returns the <head> element	3
document.images	Returns all <img> elements	1
document.implementation	Returns the DOM implementation	3
document.inputEncoding	Returns the document's encoding (character set)	3
document.lastModified	Returns the date and time the document was updated	3
document.links	Returns all <area> and <a> elements that have a href attribute	1
document.readyState	Returns the (loading) status of the document	3
document.referrer	Returns the URI of the referrer (the linking document)	1
document.scripts	Returns all <script> elements	3
document.strictErrorChecking	Returns if error checking is enforced	3
document.title	Returns the <title> element	1
document.URL	Returns the complete URL of the document	1

## Ćwiczenia

**Zad1.** Zmodyfikuj styl tekstu akapitu za pomocą kodu javascript. ([Użyj pliku zad1.html](#))

**Zad2.** Napisz funkcję JavaScript, aby uzyskać wartości Imię i Nazwisko w następującym formularzu i wyświetli je w konsoli. ([Użyj pliku zad2.html](#))

**Zad3.** Napisz program w JavaScript ustawiający kolor tła akapitu. ([Użyj pliku zad1.html](#))

**Zad4.** Oto przykładowy plik html z przyciskiem wysyłania. Napisz funkcję JavaScript, aby uzyskać wartość atrybutów href, hreflang, rel, target i type określonego łącza. ([Użyj pliku zad4.html](#))

**Zad.5** Napisz funkcję JavaScript, aby dodawać wiersze do tabeli. ([Użyj pliku zad5.html](#))

**Zad.6** Napisz funkcję JavaScript, która akceptuje wiersz, kolumnę (w celu identyfikacji określonej komórki) oraz ciąg znaków aktualizujący zawartość tej komórki. ([Użyj pliku zad6.html](#))

**Zad.7** Napisz funkcję JavaScript, która tworzy tabelę, akceptuje numery wierszy i kolumn od użytkownika i wprowadza numer wiersz-kolumna jako zawartość (np. wiersz-0 kolumna-0) komórki. ([Użyj pliku zad7.html](#))

**Zad.8** Napisz program JavaScript usuwający elementy z listy rozwijanej. ([Użyj pliku zad8.html](#))

[Wersja przeglądarkowa zadań](#)