Co to jest JavaScript?

JavaScript to język skryptowy dla sieci. Jest to język interpretowany, co oznacza, że nie potrzebuje kompilatora do tłumaczenia swojego kodu, tak jak C lub C++. Kod JavaScript działa bezpośrednio w przeglądarce internetowej.

Najnowszą wersją języka jest ECMAScript 2018, który został wydany w czerwcu 2018 roku.

JavaScript współpracuje z HTML i CSS do tworzenia aplikacji lub stron internetowych. JavaScript jest obsługiwany przez większość nowoczesnych przeglądarek internetowych, takich jak Google Chrome, Firefox, Safari, Microsoft Edge, Opera itp. Większość przeglądarek mobilnych na Androida i iPhone'a obsługuje teraz również JavaScript.

JavaScript kontroluje dynamiczne elementy stron internetowych. Działa w przeglądarkach internetowych, a ostatnio również na serwerach internetowych. Interfejsy programowania aplikacji (API) są również obsługiwane przez JavaScript, co zapewnia większą funkcjonalność.

Zrozumienie wszystkich sposobów działania JavaScriptu jest trochę łatwiejsze, gdy zrozumiesz, jak działa programowanie internetowe, więc dowiedzmy się więcej.

Bloki konstrukcyjne aplikacji internetowych

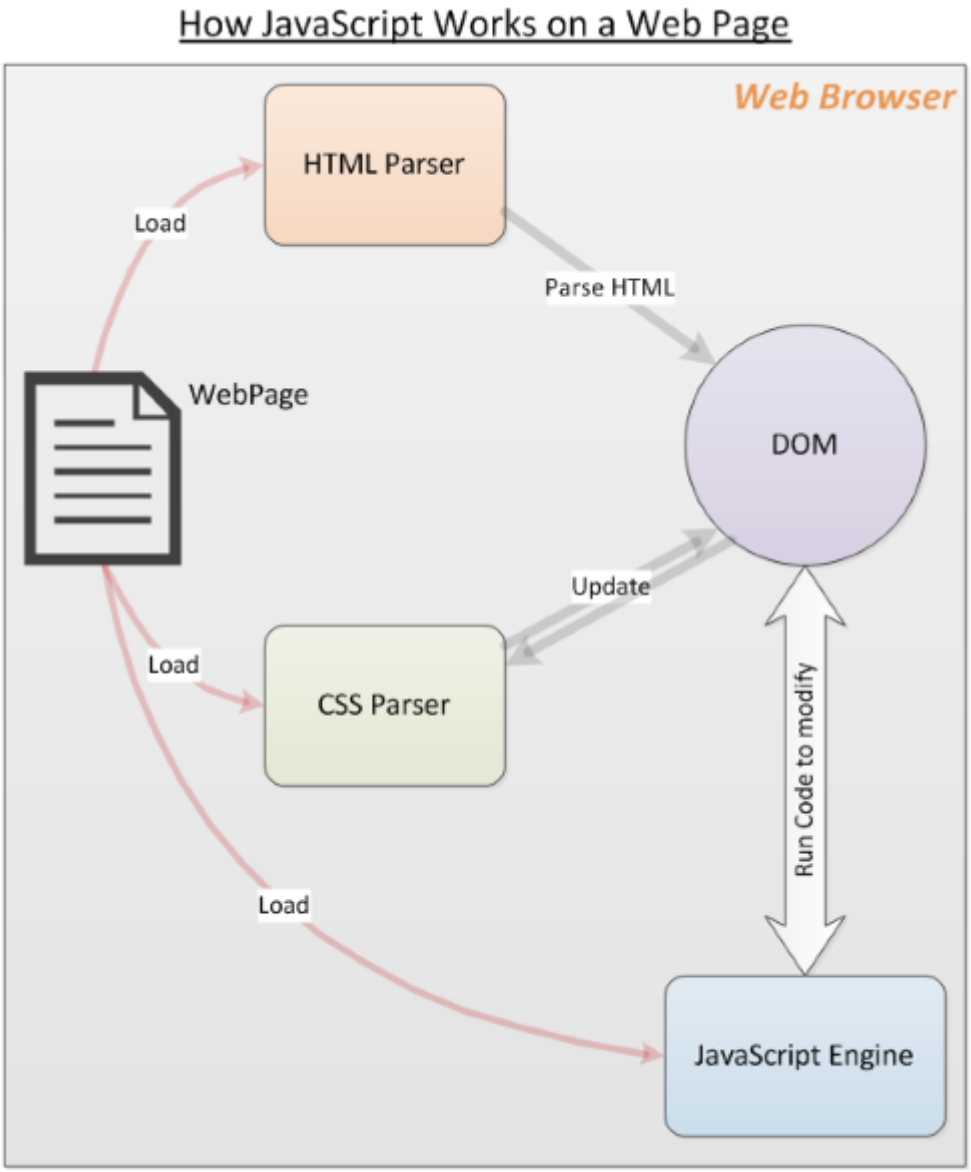
Istnieją trzy komponenty, które tworzą strony internetowe i aplikacje: Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) i JavaScript. Każdy z nich odgrywa rolę w tworzeniu aplikacji internetowej.

- **HTML** to język znaczników, który tworzy szkielet strony internetowej. Wszystkie paragrafy, sekcje, obrazy, nagłówki i tekst są napisane w HTML. Treść pojawia się na stronie w kolejności, w jakiej są napisane w HTML.
- **CSS** kontroluje styl i dodatkowe aspekty układu. CSS służy do tworzenia projektu strony internetowej, tworząc kolory, czcionki, kolumny, obramowania itp. Przenosi stronę od zwykłych elementów tekstowych do kolorowych projektów.
- Trzecim elementem jest **JavaScript**. HTML i CSS tworzą strukturę, ale nic z nią nie robią. JavaScript tworzy dynamiczną aktywność w Twojej aplikacji. Skrypty w JavaScript są tym, co kontroluje funkcje po kliknięciu przycisków, w jaki sposób uwierzytelniane są formularze haseł, jak kontrolowane są media.

Wszystkie trzy części współpracują ze sobą, tworząc pełnowymiarowe aplikacje. Dobrym pomysłem byłoby **dowiedzieć się więcej o HTML i CSS** , jeśli nie czujesz się z nimi do końca zaznajomiony.

Jak działa JavaScript?

Przed napisaniem JavaScript ważne jest, aby wiedzieć, jak działa pod maską. Istnieją dwie ważne części, o których należy się dowiedzieć: Jak działa przeglądarka internetowa i Model obiektowy dokumentu (DOM).



Przeglądarka internetowa ładuje stronę internetową, analizuje kod HTML i na podstawie zawartości tworzy tak zwany model obiektu dokumentu (DOM). DOM przedstawia podgląd strony internetowej na żywo w kodzie JavaScript.

Przeglądarka pobierze wszystko, co jest powiązane z HTML, jak obrazy i pliki CSS. Informacje CSS pochodzą z parsera CSS.

HTML i CSS są łączone przez DOM, aby najpierw utworzyć stronę internetową. Następnie silnik JavaScript przeglądarki ładuje pliki JavaScript i kod wbudowany, ale nie uruchamia kodu od razu. Czeka na zakończenie ładowania HTML i CSS.

Po wykonaniu tej czynności JavaScript jest wykonywany w kolejności, w jakiej został napisany kod. Powoduje to, że DOM jest aktualizowany przez kod JavaScript i renderowany przez przeglądarkę.

Kolejność tutaj jest ważna. Gdyby JavaScript nie czekał na zakończenie HTML i CSS, nie byłby w stanie zmienić elementów DOM.

Osadzanie kodu JavaScript w HTML

Najprostszym sposobem na umieszczenie kodu JS w HTML jest użycie tagów **<script></script>**. Wszystko co znajdzie się pomiędzy tymi znacznikami będzie traktowane jako kod JavaScript.

```
<script>
    alert('Hello World!');
</script>
```

Elementy **<script></script>** możemy umieścić w blokach head oraz body jednak miejsce, gdzie umieścimy nasz kod JavaScript, ma znaczenie.

```
1  <!DOCTYPE html>
2
3  <html>
4  <head>
5      <meta charset="UTF-8">
6      <title>Osadzanie skryptów</title>
7  </head>
8  <body>
9
10     <script>
11         alert('Hello World!');
12     </script>
13 </body>
14 </html>
```

Miejsce osadzenia kodu JavaScript ma znaczenie

Z racji tego, że **kod HTML oraz JavaScript jest parsowany (czytany) od góry do dołu** – linijka po linijce to miejsce, w którym osadzimy kod JavaScript, ma duże znaczenie.

Kiedy nasz kod JavaScript jest umieszczony w sekcji head, może dojść do sytuacji gdzie strona lub aplikacja internetowa, zamiast wyrenderować (wyświetlić) kod HTML zajmuje się wykonaniem skryptu JavaScript, co w efekcie może skutkować **wolnym ładowaniem się strony lub błędami** spowodowanymi brakiem nie wyrenderowanych jeszcze elementów kodu HTML, na których chcemy operować.

```html
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <title>Miejsce umieszczenie kodu na znaczenie</title>

    <script>
        const elementOne = document.querySelector('#one');
        console.log(elementOne); // null
    </script>
</head>
<body>

    <div id="one"></div>
    <div id="two"></div>

    <script>
        const elementTwo = document.querySelector('#two');
        console.log(elementTwo); // element div
    </script>
</body>
</html>
```

W powyższym kodzie zostały umieszczone dwa elementy script - jeden w sekcji head a drugi na końcu sekcji body.

Kod JavaScript w obu tych elementach ma za zadanie **odnaleźć konkretne elementy o identyfikatorach one oraz two**.

Kod osadzony w sekcji head nie jest w stanie wykonać tego zadania i w związku z tym zmienna elementOne zawiera wartość null. Dzieje się tak, ponieważ kod uruchomiony w sekcji head **zostanie wykonany wcześniej niż kod HTML zawierający informacje o elementach one oraz two**.

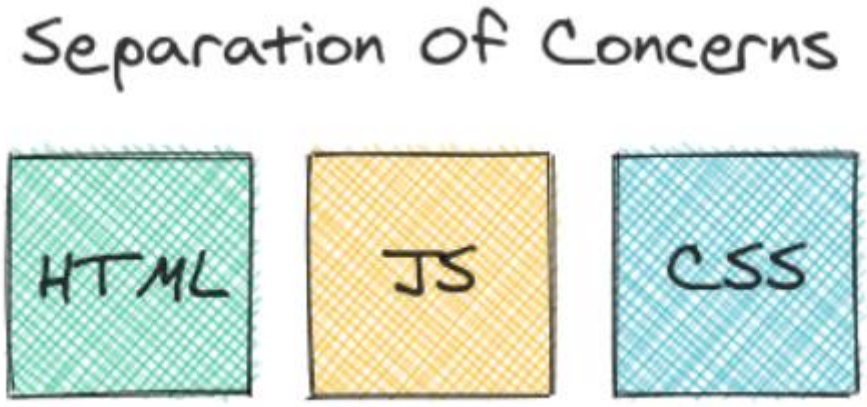
Kod JavaScript umieszczony na końcu sekcji body bez problemu odnalazł swój element gdyż został on wykonany po kodzie HTML odpowiadajacym za elementy one oraz two.

Pamiętaj! Z kodem JS oraz HTML pracuj w myśl zasady **najpierw wyrenderuj kod HTML, a następnie zajmij się interpretacją i wykonaniem kodu JavaScript** i właśnie dlatego najlepszą praktyką jest umieszczenie kodu JavaScript odpowiedzialnego za działanie **strony lub aplikacji** na samym końcu sekcji body.

Uwaga! Czasami umieszczenie kodu JavaScript w sekcji head jest uzasadnione i zazwyczaj ten wyjątek dotyczy zewnętrznych bibliotek takich jak np. jQuery lub integracji z usługami takimi jak Google Analytics.

Separation of Concerns (SoC)

Separation of Concerns (SoC) to zasada projektowa, która namawia do stosowania tzw. podziału odpowiedzialności. W praktyce oznacza to podział programu na mniejsze oddzielone od siebie elementy, które połączone tworzą jedną spójną całość.



W kontekście HTML i JS oznacza to oddzielenie kodu HTML od kodu JavaScript w taki sposób, że kod JavaScript piszemy w oddzielnym pliku z rozszerzeniem .js i dołączamy go do naszego pliku .html. (Dokładnie tak jak w przypadku HTML i CSS).

Aby dołączyć plik z kodem JavaScript do kodu HTML to w elemencie <script></script> musimy dodać atrybut src ze ścieżką do pliku JavaScript który chcemy dołączyć.

Kod JavaScript – app.js

```
const elementOne = document.querySelector('#one');
console.log(elementOne);
const elementTwo = document.querySelector('#two');
console.log(elementTwo);
```

Kod HTML z załadowanym plikiem app.js

```
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <title>Miejsce umieszczenie kodu na znaczenie</title>
</head>
<body>

    <div id="one"></div>
    <div id="two"></div>

    <script src="app.js"></script>
</body>
</html>
```