

Zajęcia 3 - Wprowadzenie do Javascript

Co to jest Javascript

Javascript jest językiem skryptowym pozwalającym na dołączanie dodatkowej funkcjonalności do stron WWW. Jest ona najczęściej związana z modyfikacją załadowanego w przeglądarce drzewa dokumentu DOM. Dzięki skryptom Javascript strony stają się bardziej responsywne, dynamiczne i bardziej przyjazne dla użytkownika.

Składnia języka Javascript jest wzorowana na C – podstawowe instrukcje sterujące wyglądają podobnie.

Uwaga – nie należy mylić języka Javascript z Javą – międzyplatformowym językiem wysokiego poziomu – to zupełnie inna technologia.

Dołączenie kody Javascript do strony

Kod Javascript można, podobnie jak CSS, podłączyć do strony WWW w postaci pliku zewnętrznego, jak i osadzić go w znaczniku `<script>`.

Istnieje również możliwość umieszczenia kodu bezpośrednio w znaczniku (związane jest to z konkretnym zdarzeniem).

Metoda 1 – plik zewnętrzny

Metoda ta polega na umieszczeniu w sekcji `<head>` strony WWW odnośnika do zewnętrznego pliku Javascript (z rozszerzeniem `.js`):

```
<html>
  <head>
    <script type="text/javascript" src="nazwa_pliku.js"></script>
  </head>
  <body>

    </body>
</html>
```

Metoda 2 – skrypt osadzony

Druga metoda polega na zagnieżdżeniu w sekcji `<head>` znacznika `<script>` z kodem Javascript:

```
<html>
  <head>
    <script>
      // kod Javascript
    </script>
  </head>
  <body>

  </body>
</html>
```

Funkcje w Javascript

Kod Javascript na stronach jest najczęściej zaimplementowany w postaci funkcji, które są wygodnym sposobem na podzielenie funkcjonalności skryptu na mniejsze części.

Definicja przykładowej funkcji znajduje się poniżej. W jej ramach uruchamiana jest wbudowana w przeglądarkę funkcja `window.alert()` pokazująca okienko z odpowiednim komunikatem.

```
function wyswietlAlert() {
  window.alert('Komunikat!')
}
```

Poniżej wersja tej funkcji z parametrem:

```
function wyswietlAlert(komunikat) {
  window.alert('Przekazany komunikat: ' + komunikat)
}
```

W powyższym przykładzie warto zwrócić uwagę na konkatencję łańcuchów znaków (przy pomocy operatora „+”).

W najnowszej specyfikacji Javascript ES6 można wykorzystywać skrócone definicje funkcji, tzw. arrow functions:

```
const wyswietlAlert = komunikat => alert(`Przekazany komunikat: ${komunikat}`)
```

Dodatkowo powyższy przykład wykorzystuje tzn. literal strings, czyli szablony stringów mogące zawierać znak nowej linii oraz umożliwiające włączenie w nie wartości zmiennych przy pomocy konstrukcji `${zmienna}`.

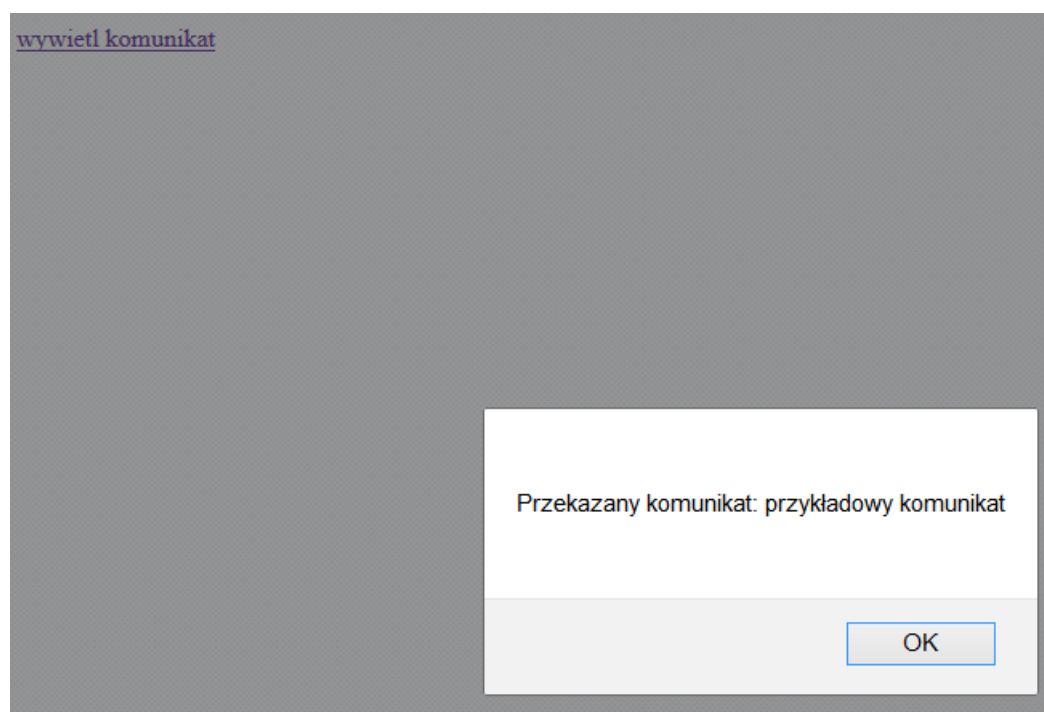
Wywołanie funkcji

Javascript jest językiem opartym na zdarzeniach. Zdarzenia są najczęściej związane z konkretnym elementem strony (np. przyciskiem, odnośnikiem) oraz inicjowane przez użytkownika witryny (np. poprzez kliknięcie w jakiś element).

Poniżej przykład odnośnika, z którym związane jest zdarzenie „click”. Zdarzenie podłączane jest poprzez dodanie atrybutu „onclick”, w środku którego może znaleźć się kod Javascript (np. wywołanie funkcji).

```
<a href="#" onclick="wyswietlAlert('przykładowy komunikat')">wyświetl  
komunikat</a>
```

Po kliknięciu w odnośnik w przeglądarce pokaże się komunikat w nowym okienku:



Dynamiczne podłączenie zdarzeń

Powyższy przykład prezentuje klasyczną składnię, która jednak nie jest polecana, gdyż wymusza łączenie kodu HTML z Javascript. Lepszym rozwiązaniem jest wykorzystanie zdarzenia onLoad obiektu window i w jego ramach programowe podłączenie obsługi zdarzeń:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script>
    window.onload = () => {
      document.querySelector('#btn-alert').onclick = () => {
        alert('Komunikat!')
      }
    }
  </script>
</head>
<body>
  <button id="btn-alert">Pokaż alert</button>
</body>
</html>
```

Zmiana drzewa dokumentu DOM

Funkcjonalność Javascript bardzo często wiąże się z modyfikacją DOM, czyli drzewa dokumentu załadowanego w oknie przeglądarki. W poniższym przykładzie zmieniany jest kolor tła całej strony (poprzez obiekt style):

```
function zmienKolorTla() {
  document.body.style.backgroundColor = "red"
}
```

Javascript a formularze

Javascript jest również często wykorzystywany do rozbudowy funkcjonalności formularzy. Przykładami może tu być przeliczanie wartości, w zależności od wyboru użytkownika, bądź też walidacja formularza (sprawdzenie przed wysłaniem).

Walidacja formularza

Poniżej pokazany jest przykład funkcji walidującej pole formularza. Wykonywane są dwa sprawdzenia; pierwsze dotyczy wypełnienia pola, drugie wartości w nie wprowadzonej (musi być to liczba).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function waliduj() {
        const wartosc = document.getElementById('rok').value

        if (wartosc == "") {
          alert("Pole rok musi być wypełnione.")
          return false
        } else if (isNaN(wartosc)) {
          alert("Pole rok musi być liczbą.")
          return false
        }

        return true
      }

      window.onload = () => {
        document.querySelector('#form').onsubmit = waliduj
      }
    </script>
  </head>
  <body>
    <form method="get" action="" id="form">
      Rok: <input type="text" name="rok" id="rok" />
      <input type="submit" name="wyslij" value="Wyślij" />
    </form>
  </body>
</html>
```

W powyższym przykładzie warto również zwrócić uwagę na moment, kiedy wywoływana jest funkcja `waliduj()` (zdarzenie „`onsubmit`”). Jeśli funkcja podłączona do tego zdarzenia zwróci wartość `false`, formularz nie zostanie wysłany.

Przeliczanie wartości

Przykład 1

W poniższym przykładzie przeliczana jest cena samochodu podana w PLN na cenę w EURO.

Formularz dodawania ogłoszenia	
Marka	<input type="text"/>
Model	<input type="text"/>
Rok	<input type="text"/>
Silnik	<input checked="" type="radio"/> Benzyna <input type="radio"/> Diesel
Wyposażenie	<input type="checkbox"/> Klimatyzacja <input type="checkbox"/> ABS <input type="checkbox"/> Wspomaganie
Stan ogólny	<input type="text" value="Dobry"/>
Cena PLN	<input type="text" value="1000"/>
Cena EURO	<input type="text" value="250"/> <input type="button" value="Nalicz"/>
<input type="button" value="Potwierdź"/> <input type="button" value="Reset"/>	

```
function nalicz_1() {  
    const cena = document.getElementById('cena_pln').value*1 / 4  
  
    document.getElementById('cena_euro').value = cena  
}
```

Przykład 2

W kolejnym przykładzie przeliczana jest cena z PLN na cenę w jednej z dwóch walut (do wyboru).

Formularz dodawania ogłoszenia	
Marka	<input type="text"/>
Model	<input type="text"/>
Rok	<input type="text"/>
Silnik	<input checked="" type="radio"/> Benzyna <input type="radio"/> Diesel
Wyposażenie	<input type="checkbox"/> Klimatyzacja <input type="checkbox"/> ABS <input type="checkbox"/> Wspomaganie
Stan ogólny	<input type="text" value="Dobry"/>
Cena PLN	<input type="text" value="1000"/>
Cena waluta	<input type="text" value="200"/> <input type="radio"/> EURO <input checked="" type="radio"/> USD
<input type="button" value="Nalicz"/>	
<input type="button" value="Potwierdź"/>	<input type="button" value="Reset"/>

```
function nalicz_2() {  
    const kurs_euro = 4.29  
    const kurs_usd = 3.79  
    let wybrana_waluta = 1  
  
    const wybrane = document.querySelector('input[name=waluta]:checked')  
    if (wybrane) {  
        if (wybrane.value == 'EUR') {  
            wybrana_waluta = kurs_euro  
        } else if (wybrane.value == 'USD') {  
            wybrana_waluta = kurs_usd  
        }  
    }  
  
    document.getElementById('cena_waluta').value =  
document.getElementById('cena_pln').value*1 / wybrana_waluta  
}
```

Przykład 3

Ostatni przykład demonstruje sposób działania konstrukcji `switch` na przykładzie sprawdzenia wartości wybranej w liście rozwijalnej (element `<select>`).

Formularz dodawania ogłoszenia	
Marka	<input type="text"/>
Model	<input type="text"/>
Rok	<input type="text"/>
Silnik	<input checked="" type="radio"/> Benzyna <input type="radio"/> Diesel
Wyposażenie	<input type="checkbox"/> Klimatyzacja <input type="checkbox"/> ABS <input type="checkbox"/> Wspomaganie
Stan ogólny	<input type="text" value="Idealny"/>
Cena PLN	<input type="text" value="1000"/>
Cena stan	<input type="text" value="1200"/> <input type="button" value="Nalicz"/>
<input type="button" value="Potwierdź"/> <input type="button" value="Reset"/>	

```
function nalicz_3() {  
    var stan = 0 //stara składnia  
  
    switch (document.getElementById('stan').value) {  
        case "1": stan = 0.8; break;  
        case "2": stan = 0.9; break;  
        case "3": stan = 1.0; break;  
        case "4": stan = 1.1; break;  
        case "5": stan = 1.2; break;  
    }  
  
    document.getElementById('cena_stan').value =  
document.getElementById('cena_pln').value * stan  
}
```


Przydatne linki

- <http://www.w3schools.com/jsref/default.asp> - opis dostępnych parametrów DOM
- <https://kangax.github.io/compat-table/es6/> - tabela kompatybilności składni ES6 z poszczególnymi przeglądarkami
- <https://babeljs.io> – kompilator Javascript ES6 do ES5

Zadania

1. Proszę o uruchomienie naliczania wartości w przykładzie 3 bez konieczności naciskania przycisku „Nalicz” (należy wykorzystać zdarzenie „onchange”)
2. Proszę o zbudowanie prostego kalkulatora:
 - a. Kalkulator będzie dopuszczał dwie składowe działania (w postaci dwóch pól tekstowych)
 - b. Pomiędzy polami powinna znaleźć się lista rozwijalna zawierająca dopuszczalne działania (+ - * / %)
 - c. Wynik działania powinien być wpisywany do trzeciego pola tekstowego, które powinno mieć parametr „tylko do odczytu”
 - d. Wynik działania powinien być liczony automatycznie po wprowadzeniu wartości w pola tekstowe, wybraniu działania, bądź wciśnięciu przycisku „Licz”
 - e. Przed wykonaniem obliczeń należy sprawdzić, czy użytkownik dobrze wypełnił pola (pole wypełnione liczbą). W razie błędu należy to zasygnalizować użytkownikowi.
 - f. Pod kalkulatorem proszę o wyświetlenie historii wykonywanych operacji w formacie: **2 + 2 = 4**

<input type="text"/>	<input type="text" value="+"/> ▼	<input type="text"/>	=	<input type="text"/>	<input type="button" value="Licz"/>
----------------------	----------------------------------	----------------------	---	----------------------	-------------------------------------