

Tworzenie Aplikacji Internetowych

Laboratorium 3

JavaScript (JS)

JavaScript jest językiem skryptowym interpretowanym przez przeglądarkę. Umożliwia on m.in. modyfikację elementów na stronie HTML, czy obsługę zdarzeń. Umieszczenie skryptu JS na stronie może być następujące:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="skrypt.js"></script> <!-- plik zewnętrzny ze skryptem (podejście 1) -->
    <script> /* skrypt umieszczony w kodzie strony (podejście 2) */
      // ... treść skryptu
    </script>
  </head>
  <body>
    <!-- zawartość strony HTML -->
    <script src="skrypt.js"></script> <!-- plik zewnętrzny (podejście 3) -->
    <script> /* skrypt umieszczony w kodzie strony (podejście 4) */
      // ... treść skryptu
    </script>
  </body>
</html>
```

Uwaga: skrypty umieszczone w znaczniku **head** nie mogą bezpośrednio przetwarzać zawartości **body** (nie jest ona jeszcze wczytana przez przeglądarkę). Wyjątkiem są **funkcje**, które są wywoływane po załadowaniu strony (np. w przypadku wystąpienia zdarzenia kliknięcia na element strony). Skrypty umieszczone w końcowej sekcji znacznika **body** (podejście 3 i 4) **mają dostęp** do wszystkich umieszczonych wyżej elementów.

Składnia (podstawowa) języka JS jest podobna do składni języka C++, co zaprezentowano w poniższej tabeli:

Deklaracja zmiennej (zasięg globalny lub funkcyjny) – nie wymaga podania typu zmiennej	var zmienna = wartość; // np. var a = 15; var b = "tekst";
Deklaracja zmiennej (zasięg globalny lub określony nawiasami { oraz })	let zmienna = wartość; // np. let a = 15; let b = "tekst";
Deklaracja stałej (lub stałej referencji)	const stała = wartość;
Wyświetlenie typu zmiennej	typeof zmienna;
Deklaracja tablicy (nie wymagania podania typu)	let tab = [1, 2, 3, 4]; // pusta tablica: let tab = [];
Odwołanie się do elementu tablicy	tab[indeks];
Dodanie elementów na koniec tablicy	tab. push (element, [element], ...)
Dodanie elementów na początku tablicy	tab. unshift (element, [element], ...)
Pobranie liczby elementów tablicy	let el = tab.length;
Instrukcje warunkowe	if (warunki) { ... } else { ... }
Pętla for	for (inicjalizacja; warunek; aktualizacja) { ... }
Pętla while	while (warunki) { ... }
Instrukcja switch	switch (zmienna) { case wartość1: ... break; case wartość2: ... break; default: ... }
Funkcje (nie wymaga podania typu wartości zwracanej oraz typów argumentu)	function nazwa(argumenty) { ... return wartość; }
Funkcje strzałkowe (mogą być np. definiowane w locie, np. jako argumenty funkcyjne innych funkcji)	(argumenty) => { ... }

W języku JavaScript można w prosty sposób pobrać obiekty reprezentujące dane elementy strony internetowej. Szczegóły odnośnie odnoszenia się do takich elementów zostaną przedstawione podczas zajęć związanych z DOM oraz programowaniem obiektowym. W celach tego laboratorium wystarczające będzie przedstawienie następujących funkcji i atrybutów:

```
var element = document.getElementById("nazwa"); // pobranie elementu dla znacznika o id="nazwa"
element.innerHTML = "nowa zawartość"; // przypisanie nowej zawartości wewnątrz znacznika
element.innerHTML += "dodana zawartość"; // dodanie zawartości do znacznika
element.value; // pobranie wartości z pola (np. wpisanej zawartości dla znaczników input)
element.setAttribute("nazwa", "wartość"); // dodanie nowego atrybutu do znacznika
element.removeAttribute("nazwa"); // usunięcie atrybutu ze znacznika
element.classList.add("stylCSS"); // dodanie do konkretnego znacznika kolejnego stylu CSS
element.classList.remove("stylCSS"); // usunięcie z konkretnego znacznika danego stylu CSS
```

Przykład skryptu obliczającego silnię i wyświetlającego ją na stronie internetowej:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <p id="sila"></p>
    <p id="silb"></p>
    <script>
      function silnia(l) { // utworzenie funkcji zwracającą wartość silni
        if (l < 0) return -1;
        else if (l == 0) return 0;
        let s = 1;
        for (let i = 2; i <= l; i++) {
          s *= i;
        }
        return s;
      }
      // elementa reprezentuje znacznik p o id="sila"
      let elementa = document.getElementById("sila");
      // wywołanie utworzonej funkcji silnia i przypisanie wyniku do zmiennej
      let wynik1 = silnia(6);
      // umieszczenie wyniku w znaczniku p o id="sila"
      elementa.innerHTML = "Silnia z 6: " + wynik1;
      // poniżej to samo dla silni z liczby 12 i znacznika o id="silb"
      let elementb = document.getElementById("silb");
      let wynik2 = silnia(12);
      elementb.innerHTML = "Silnia z 12: " + wynik2;
    </script>
  </body>
</html>
```

Wynik działania powyższego skryptu:



Silnia z 6: 720

Silnia z 12: 479001600

Zadanie 1

Napisać skrypt, który obliczy sumę oraz znajdzie maksymalną i minimalną wartość z następującej tablicy:

let tab = [5, 7, 21, 11, 9, 19, 13];

Skrypt nie musi, ale może, bazować na funkcjach. Otrzymane wyniki wyświetlić w elementach **p**.

Zdarzenia

Zdarzenia czyli wystąpienia określonych sytuacji. Mogą być wywoływane przez użytkownika lub różne elementy na stronie internetowej (najprostszym zdarzeniem jest obsługa kliknięcia). Najprostszym sposobem jest przypisanie zdarzeń jako atrybutów wybranych znaczników.

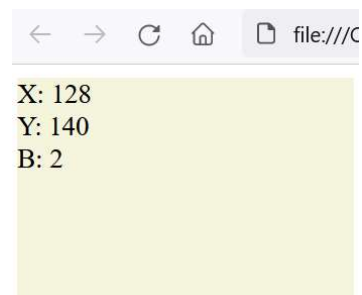
Zdarzenia związane z **myszką** (parametr event umożliwia dostęp do pozycji myszki oraz statusu przycisków):

- onclick – kliknięcie w dany element
- ondblclick – podwójne kliknięcie na element
- onwheel – wykorzystanie kółeczka myszy
- onmouseover – najechanie kursorem na element
- onmouseout – opuszczenie elementu przez kursor
- onmouseenter – wejście kursorem na element
- onmouseleave – opuszczenie elementu przez kursor
- onmousedown – wciśnięcie klawisza
- onmouseup – puszczenie klawisza

Przykład:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      #pole { width: 200px; height: 200px; background: beige; }
    </style>
    <script>
      function pinfo(event) {
        let pel = document.getElementById("pole");
        pel.innerHTML = "X: " + event.clientX + "<br>";
        pel.innerHTML += "Y: " + event.clientY + "<br>";
        pel.innerHTML += "B: " + event.buttons + "<br>";
      }
    </script>
  </head>
  <body>
    <div id="pole" onmousemove="pinfo(event);"></div>
  </body>
</html>
```

Wynik:



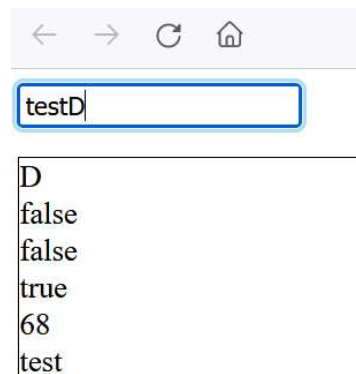
Zdarzenia związane z klawiaturą (parametr event pozwala uzyskać dostęp do informacji o klawiszach):

- onkeypress – wciśnięcie i puszczenie klawisza
- onkeydown – wciśnięcie klawisza
- onkeyup – puszczenie klawisza
- oninput – zdarzenie związane ze zmianą wywoływane przez specjalne typy input
- onchange – zmiana wartości elementu (np. dla znaczników input)
- onfocus – przy aktywacji danego elementu (np. znacznika input)
- onblur – przy utracie aktywacji danego elementu
- onsubmit – wysyłanie formularza

Przykład:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function wypisz(event) {
        let pol = document.getElementById("pole");
        let wyn = document.getElementById("klawisze");
        wyn.innerHTML = event.key + "<br>";
        wyn.innerHTML += event.altKey + "<br>";
        wyn.innerHTML += event.ctrlKey + "<br>";
        wyn.innerHTML += event.shiftKey + "<br>";
        wyn.innerHTML += event.keyCode + "<br>";
        wyn.innerHTML += pol.value + "<br>";
      }
    </script>
  </head>
  <body>
    <input id="pole" onkeypress="wypisz(event);"><br>
    <p id="klawisze" style="border: 1px solid black;"></p>
  </body>
</html>
```

Wynik:



testD

D
false
false
true
68
test

Inne zdarzenia:

- onresize – przy zmianie rozmiaru elementu (najczęściej dla body)
- onload – gdy dany element zostanie załadowany
- onselect – przy zaznaczaniu danej treści
- unload – przy usuwaniu elementu (np. wyjściu ze strony)
- ...

Język JavaScript oferuje dużo metod związanych z operacją na **tekście i liczbach**, najważniejsze z nich to:

```
liczba.toString(); // zamiana liczby na tekst
"tekst: " + liczba; // zamiana liczby na tekst
parseInt(tekst); // zamiana tekstu na liczbę całkowitą – zwraca NaN w przypadku błędu
parseFloat(tekst); // zamiana tekstu na liczbę zmiennoprzecinkową – zwraca NaN w przypadku błędu
tekst.length; // zwrócenie długości tekstu
tekst[index]; // odwołanie się do konkretnej litery tekstu
tekst.indexOf(fraza); // wyszukanie indeksu danej frazy – zwraca -1 w przypadku niepowodzenia
tekst.lastIndexOf(fraza); // wyszukanie indeksu danej frazy (zaczynając od końca)
tekst.substr(odkąd, ileznaków); // wycina fragment tekstu określony przez parametry
tekst.substring(odkąd, dokąd); // wycina fragment tekstu określony przez parametry
tekst.replace(co, naco); // zmienia w tekście frazy określone przez parametry
```

Zadanie 2

Celem tego zadania jest utworzenie prostego kalkulatora. W celu realizacji zadania należy:

- Utworzyć dwa znaczniki input – pierwszy o id="liczba1" oraz drugi o id="liczba2".
- Utworzyć cztery przyciski ze zdarzeniem onclick wywołującym odpowiednio funkcję sumy, odejmowania, dzielenia i mnożenia (np. <button onclick="dodaj();">Dodawanie</button>)
- Utworzyć w skrypcie cztery funkcje (bez parametrów): suma, odejmowanie, dzielenie, mnożenie. Każda z funkcji powinna pobierać wartości z obu pól input (.value), zamieniać je na wartości liczbowe (parseFloat), wykonywać odpowiednie działanie i wyświetlać wynik w wybranym przez siebie znaczniku na stronie
- (opcjonalnie) dodać do znaczników input zdarzenie onkeypress, które przy zmianie wprowadzanego tekstu powinno wywoływać funkcję ostatnio wybraną przez kliknięcie w jeden z przycisków i obliczyć wynik.