

Podstawy JavaScript – ćwiczenia

Kontekst:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl"
lang="pl">
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
    <title>Dodawanie</title>
    <script type="text/javascript">

      </script>
  </head>
  <body>
    <p>Treść dokumentu</p>
  </body>
</html>
```

Wstawianie treści skryptu

Treść skryptu w dokumencie **html** wstawiamy między znaczniki

```
<script type="text/javascript">
</script>
```

W dokumencie **xhtml**, aby skrypty nie kolidowały ze składnią, należy w treści skryptu zamieniać znaki "&" oraz "<" na encje znakowe & oraz <; np. zamiast:

```
<script type="text/javascript">if (2 < 4) {}</script>
```

należy napisać:

```
<script type="text/javascript">if (2 &lt;; 4) {}</script>
```

Innym sposobem zapobieżenia takiej kolizji jest otoczenie treści całego skryptu blokiem CDATA

```
<script type="text/javascript"><![CDATA[ if (2 < 4) {}]]></script>
```

Uniwersalną metodą jest umieszczenie treści skryptu w osobnym pliku – nie trzeba wówczas stosować encji znakowych ani otaczać treści skryptu blokiem CDATA. Podczas złączania „w locie” dokumentu html/xhtml ze skrytem JavaScript to przeglądarka zadba automatycznie o to, by nie doszło do kolizji składni.

Jeśli treść skryptu znajduje się w osobnym pliku, należy odwołać się do niego następująco:

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

gdzie SRC wskazuje nazwę pliku ze skrytem i/lub ścieżką do niego.

Skrypt „Witaj”

Zadaniem tego skryptu jest wyświetlenie okienka z napisem „Witaj” po załadowaniu strony do przeglądarki.

Za wyświetlenie okna dialogowego w języku javascript odpowiedzialna jest funkcja `alert()`, która w nawiasach przyjmuje argumenty do wyświetlenia.

Skrypt jest wykonywany zawsze w odpowiedzi na jakies zdarzenie. Zdarzenie przy wczytaniu strony do przeglądarki nazywa się `onLoad`. By wywołać kod javascript po tym zdarzeniu, należy przy znaczniku, po znaku równa się, wpisać nazwę wywoływanej funkcji.

```
<body onLoad="alert( 'Witaj' )">
```

W powyższym przypadku kod skryptu został wprowadzony bezpośrednio w elemencie (w tym wypadku `body`). W przypadku bardziej złożonych funkcji konieczne jest ich stworzenie w osobnym miejscu.

Zadania:

- utworzyć pustą, bez treści, stronę html i wprowadzić alert do znacznika `body` (j.w.)
- otworzyć stronę w przeglądarce

„Interaktywny” skrypt Witaj

JavaScript może pobierać dane z formularzy umieszczonych na stronie html (*o formularzach patrz „Formularze html – ćwiczenia”*).

DOM

By odczytać dane z formularza należy „dostać się” do jego elementów. Do elementów dokumentu (a nawet powyżej dokumentu – do samego okna przeglądarki) można dostać się dzięki **DOM** (*Obiektowemu Modelowi Dokumentu, ang. Document Object Model*).

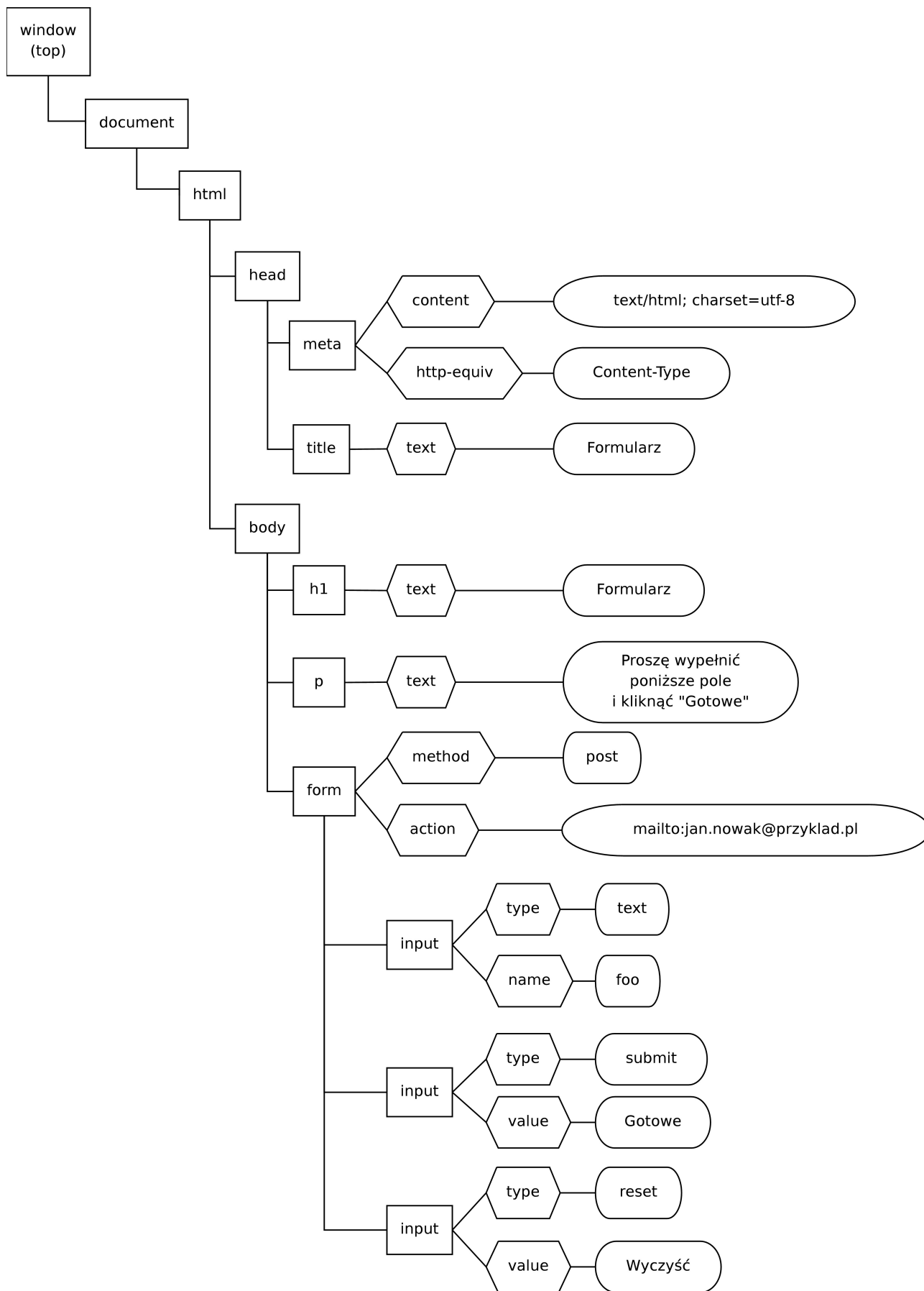
Przykładowa, prosta strona html:



Źródło tej strony:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Formularz</title>
  </head>
  <body>
    <h1>Formularz</h1>
    <p>Proszę wypełnić poniższe pole i kliknąć "Gotowe".</p>
    <form action="mailto:jan.nowak@przyklad.pl" method="post">
      <input type="text" name="foo" />
      <input type="submit" value="Gotowe" /> <input type="reset" value="Wyczyść" />
    </form>
  </body>
</html>
```

Drzewo DOM tej strony:



W zapisie wykorzystywanym m.in. w JavaScript stosuje się zapis „płaski”, gdzie poszczególne obiekty podaje się od lewej do prawej, od najwyższego w hierarchii, przez kolejne gałęzie do celu, a oddziela się poszczególne elementy pojedynczymi kropkami. Np. do wartości pola „foo” w przedstawionym powyżej formularzu, można dostać się w ten sposób:

```
top.document.forms[0].foo.value
```

lub nieco krócej

```
document.forms[0].foo.value
```

Zapis „forms[0]” oznacza „pierwszy z formularzy w dokumencie”.

Na skróty

Jeśli element ma ustawiony atrybut `id`, to można go znaleźć za pomocą `getElementById(id)`

np. `<p id="wazny">To jest ważny paragraf</p>`

w treści skryptu można się do niego odwołać przez:

```
document.getElementById("wazny")
```

Więcej informacji nt. DOM: <http://kurs.browsehappypol.pl/JavaScript/DOM>

Zmienne

Aby móc korzystać w JavaScript z wartości pobranej z formularza należy przypisać ją do zmiennej:

```
var nazwa_zm = top.document.formularz.nazwa_elementu.value;
```

– korzystając ze słowa kluczowego `var` (ang. *variable*) została utworzona zmienna o nazwie `nazwa_zm` i, przy użyciu operatora przypisania „=”, przypisano jej wartość jednego z pól w formularzu.

Tak ustawioną zmienną wykorzystuje się następnie w skrypcie po prostu wpisując jej nazwę. Przy wykonaniu skryptu to przeglądarka podstawia właściwą wartość pod zmienną.

Funkcje

W JavaScript istnieje wiele wbudowanych funkcji (poleceń), takich, jak wcześniej wykorzystany „`alert()`”. Można też samodzielnie definiować w treści skryptu potrzebne funkcje – pod jedną nazwą grupujące kolejne działania, które skrypt powinien wykonać. Funkcje w języku JavaScript definiuje się:

```
function nazwa_funkcji() {  
    treść funkcji  
}
```

Zadania

Utworzyć stronę, na której użytkownik będzie mógł wprowadzić swoje imię i po kliknięciu na przycisk „Przywitajmy się” zostanie wyświetlone okienko „Witaj <imię>”

- utworzyć formularz w dokumencie html
- stworzyć skrypt odczytujący wprowadzone dane i wyświetlający okienko dialogowe
- otworzyć stronę w przeglądarce, wprowadzić dane do pola i kliknąć na przycisk

Formularz:

```
<form name="witaj" action="" method="post" onsubmit="return false">
    imię: <input type="text" name="imie" /> <br />
    <input type="button" value="Przywitajmy się"
onClick="witam()" />
</form>
```

W formularzu `action=""` oznacza odwołanie do samego siebie. Ponieważ nie będziemy w tym przykładzie wykorzystywać przycisku `submit`, akcja przypisana do `onSubmit` to zwrócenie „false”.

Do wprowadzenia danych wykorzystamy pole typu tekstowego.

Tworzymy też przycisk (`input type="button"`), na którym będzie napis „Przywitajmy się”, a po jego kliknięciu zostanie wywołana funkcja „`witam()`”.

Teraz między znacznikami `<script>` (w `<head>` naszego dokumentu) należy utworzyć funkcję `witam()`

```
function witam() {
    var imie = top.document.witaj.imie.value;
    alert ( 'Witaj '+imie );
}
```

W ramach stworzonej funkcji wykorzystana została wbudowana w JavaScript funkcja `alert()`, do której zostały przekazane argumenty `'Witaj '` oraz `imie`, i nakazano, przez znak `+` połączenie obu ciągów znaków w jeden tekst. Proszę zauważyć, że ciąg znaków, który ma zostać wyświetlony tak, jak jest, czyli „Witaj” wraz ze spacją, został ujęty w apostrofy, a nazwa zmiennej, pod którą „ukrywa” się właściwa treść do wyświetlenia została zapisana **BEZ** apostrofów/cudzysłówów.

Prosty kalkulator

Zadaniem kalkulatora jest obliczanie wyniku dwóch, wprowadzonych przez użytkownika na stronie liczb. Użytkownik powinien mieć możliwość wybrania działania (dodawanie, odejmowanie, mnożenie, dzielenie).

Uwagi do formularza html:

aby dać użytkownikowi możliwość wyboru należy skorzystać z typu pola „select”:

```
<select name="nazwa_elementu">
<option value="wartosc1">Tekst, który się wyświetli</option>
<option value="wartosc2">Inny tekst, który się wyświetli</option>
</select>
```

`wartosc1`, `wartosc2` – to wartości, które będzie można przekazać do skryptu

Uwagi do skryptu

1. formularze przechowują dane w postaci ciągów znaków – wpisanie do pola formularza „1” oznacza znak „1”, a nie liczbę. Wymuszenie w skrypcie zmiany typu można zrobić w następujący sposób (znak „1” został pobrany do zmiennej `a`):
`a = a - 0`
2. aby zdecydować, jakie działanie powinno być wykonane należy użyć instrukcji `switch`, która ma postać:

```
switch ( nazwa_zmiennej ) {  
    case wartosc_zmiennej : działania, które powinny być  
wykonane, gdy zmienna nazwa_zmiennej ma wartość równą  
wartosc_zmiennej;  
        break  
    case inna_wartosc_zmiennej : działania, które powinny być  
wykonane, gdy zmienna nazwa_zmiennej ma wartość równą  
wartosc_zmiennej;  
        break  
}
```

3. By przypisać elementowi formularza wynik działania funkcji należy zastosować normalne przypisanie:

```
top.document.nazwa_formularza.nazwa_elementu.value = wynik;
```

W prostych działaniach można też pominąć zmienną i od razu przypisać wynik działania:

```
top.document.nazwa_formularza.nazwa_elementu.value = a + b;
```