

# Kilka słów o JavaScript

# JavaScript (JS)

Jest to obiektowy skryptowy język programowania, stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych. Pod koniec lat 90. XX wieku organizacja ECMA wydała na podstawie JavaScriptu standard języka skryptowego o nazwie ECMAScript. Głównym autorem JavaScriptu jest Brendan Eich.

# Zastosowanie

Najczęściej spotykanym zastosowaniem języka JavaScript są strony WWW. Skrypty służą najczęściej do zapewnienia interaktywności poprzez reagowanie na zdarzenia, sprawdzania poprawności formularzy lub budowania elementów nawigacyjnych. Podczas wzbogacania funkcjonalności strony internetowej istotne jest, aby żaden element serwisu nie stał się niedostępny po wyłączeniu obsługi JavaScriptu w przeglądarce. Skrypt JavaScriptu ma znacznie ograniczony dostęp do komputera użytkownika (o ile nie zostanie podpisany cyfrowo). Niektóre strony WWW zbudowane są z wykorzystaniem JavaScriptu po stronie serwera, jednakże znacznie częściej korzysta się w tym przypadku z innych języków.

# JavaScript - zastosowanie

*Łączenie skryptów z dokumentem HTML: określana jest reakcja na predefiniowane (dla poszczególnych rodzajów elementów strony) zdarzenia. Nie tylko wciśnięcie przycisku formularza, ale również bardzo liczne inne zdarzenia myszy, klawiatury oraz stanu otwierania okna.*

- Umożliwienie modyfikowania wyglądu dokumentu, czy też np. nawigowania w historii bieżącego okna;
- Wykonywanie operacji zależnych od platformy (przeglądarka, system operacyjny);
- Wykonywanie prostych obliczeń po stronie klienta;
- Kontrola poprawności wypełnienia formularza;
- Określanie aktualnej daty i czasu;
- Generowanie lokalnie dokumentów HTML.

# Osadzanie

- W języku HTML za umieszczanie skryptów JS odpowiedzialny jest element z argumentem `type` o wartości `text/javascript` oraz argumentem `language` o wartości `javascript`. Atrybut `language` jest jednak przestarzały i zaleca się go pomijać.
- Jeżeli kod znajduje się w dokumencie XHTML, w celu uniknięcia zinterpretowania niektórych operatorów jako elementów składni języka należy otoczyć skrypt sekcją CDATA lub zapisać je w postaci encji. W XHTML nie można używać argumentu `language` w znaczeniu określenia wersji języka JS (atrybut, jeżeli jest użyty, powinien przyjąć dwuznakowe wartości opisane standardem ISO 639, np. EN, DE, PL).
- Zewnętrzne skrypty dodajemy także przy użyciu powyższego znacznika, uwzględniając jedynie parametr `src` z nazwą pliku zawierającego kod skryptu. Typ MIME dla samodzielnych plików JavaScript to `application/javascript` lub przestarzały `text/javascript`

# Środowisko działania skryptów

- Środowisko jest ograniczone do manipulowalnych przez język elementów dokumentu oraz do standardowych obiektów przeglądarki (okno, pasek adresu, status, ramki).
- Nie są możliwe tzw. operacje uprzywilejowane, jak dostęp do lokalnych plików czy drukowanie.
- Rodzaje zdarzeń - dotyczące:
  - myszy: `onMouseOver`, `onMouseOut`, `onClick`, `onMouseDown`, `onMouseUp`;
  - klawiatury: `onBlur`, `onFocus`, `onChange`, `onSelect`, `onSubmit`;
  - inne: `onLoad`, `onUnload`, `onError`, `onAbort`;
- Tradycyjnie dostępne następujące obiekty standardowe:
  - > `window`:
    - `location`, `frames`, `history`, `navigator`, `event`, `screen`;
  - `document`:
    - `links`, `anchors`, `images`, `filters`, `forms`, `applets`, `embeds`, `plug-ins`, `frames`
    - `scripts`, `all`, `selection`, `stylesheets`, `body`.

# Język JavaScript – podstawowe właściwości (1)

- Wyróżnione typy danych: "number", "string", "boolean", "function", "object", "undefined". Możliwość sprawdzenia typu: funkcja  
`string typeof(zmienna_lub_stala).`
- Operatory logiczne, arytmetyczne i bitowe -> składnia podobnie jak w Javie;
- Instrukcje warunkowe i pętle -> podobnie jak w Javie;
- Możliwość deklarowania w kodzie zmiennych i tablic zmiennych; w dowolnym miejscu kodu:  
`var zmienna = wartość; var zmienna; zmienna=wartość;`  
`var zmienna = new Array();`  
`var zmienna = new Array(wartość1, wartość2);`

# Język JavaScript – podstawowe właściwości (2)

- Dostęp do elementów tablicy; poprzez indeks – np.  
`tab1[0]="wartość";`  
Sprawdzenie rozmiaru – właściwość `length`:  
`for(var i=0;i<tab.length;i++) ...`
- Deklarowanie własnych klas – konstruktor:  
`function Licznik(poczatek)  
 { this.stan = poczatek;}`
- Metody własnych klas:  
`function zwiekszaj(){ this.stan++; }`
- Parametry metody deklarowane samą nazwą (bez wskazania typu). Możliwa jest rekurencja.
- Sprawdzenie wartości:
  - Przypisanie do `null`;
  - Funkcja `isNaN(wartość)` stwierdza niezgodność z formatem liczb.



# Włączanie skryptów do dokumentu

- Deklaracje używanych funkcji i globalnych zmiennych umieszczane zwykle w nagłówku dokumentu HTML (tag. `head` ). Oznaczony następująco:  
`<SCRIPT language="JavaScript"> kod </SCRIPT>`
- Wstawienie kodu z osobnego pliku:  
`<SCRIPT language="JavaScript" src="plik.js"> </SCRIPT>`
- Reagowanie na brak wsparcia dla JavaScript:
  - ukrywanie kodu w komentarzach, aby nie generował błędów w starych przeglądarkach:  
otwarcie komentarza HTML na początku wnętrza tagu `SCRIPT`;
- Reagowanie treścią na brak wsparcia skryptów:  
tag `<NOSCRIPT></NOSCRIPT>`

# Łańcuchy tekstowe i konwersje

- Symbole specjalne oznaczane w łańcuch przez „\”
- Tekst wytyczany cudzysłowami lub apostrofami
- Możliwość konwersji na *string*: `zmienna+="”`
- Obróbka tekstu:
  - `substring(indeksPocz, IndeksKońcowy)`
  - `indexOf(łańcuch)`
  - `escape(łańcuch)`, `unescape(łańcuch)` //URL encoding
- Konwersje liczb:
  - Zmiana reprezentacji: `(liczba)toString(podstawa)`
  - Odzyskanie liczby: `parseInt(łańcuch, podstawa)`,  
`parseInt(łańcuch)`, `parseFloat(łańcuch)`

# Jak przekazywać parametry?

- Składowanie parametru w pliku *cookie*
- Składowanie parametru w zmiennej innego okna lub ramki
- Składowanie parametru w zmiennej *top.name* (czyli w nazwie okna głównego)
- Doklejanie parametru do wołanego URL (jako tzw. *query string*)

# Obsługa okien i dialogów

- Standardowe dialogi:
  - Prosty komunikat: `alert(komunikat)`;
  - Potwierdzenie: `boolean confirm(Tekst zapytania)`
  - Prośba o wpisanie parametru:  
`prompt(Tekst zapytania, wartość domyślna)`
- Okna przeglądarki:
  - Otwarcie: `window.open(adres, tytuł, opcje)`  
np. `'left=50, top=50, width=500, height=400, toolbar=1, resizable=0'`
  - Zmiana rozmiaru: `resizeTo(szer, wys)`;
  - Pisanie wewnątrz okna: `ok1.document.writeln(tekst)`;
  - Sprawdzenie stanu i zamykanie: właściwość `closed`, metoda `close()`.

# Inne udogodnienia

- Obiekt Math:
  - Liczne funkcje i stałe matematyczne;
  - Metoda `random()`: `zmienna = Math.random()`;
- Obiekt Date: data, czas, możliwość walidacji daty:
  - konstruktor: `Date("December 31, 2000 23:59:59")`
  - albo numery roku, miesiąca, dnia;
- Funkcje:
  - `getDate()` -> zwraca dzień miesiąca z podanej daty;
  - `getDay()` -> dzień tygodnia;
  - `getHours()`, `getMinutes()`, `getSeconds()`;
  - `getMonth()`, `getFullYear()`;
  - `getTime()` -> milisekundy;
- Wykonanie z opóźnieniem:
  - `setTimeout('operacja()', milisekundy)`

# Obsługa formularzy

- Dostęp do obiektów zgodnie z ich hierarchią, możliwy wg atrybutu **name** lub **id** danej kontrolki;
- Użyteczne zdarzenia zdefiniowane dla elementów formularza: `onBlur`, `onChange`, `onClick`, `onSelect`
- Obsługa zdarzenia *onSubmit* formularza: powinna być funkcją logiczną; tylko w wypadku wartości `true` wysyłanie jest kontynuowane.
- Informacje dodatkowe lub kondensowane można „upakowywać” w pola ukryte (*hidden*) formularza.
- Metody `reset()` i `submit()` pozwalają symulować odpowiadające im polecenia użytkownika.

# Słowa zarezerwowane

- `var`, `const`, `typeof`, `function`;
- Praktycznie wszystkie słowa kluczowe Javy;
- Nazwy obiektów w środowisku przeglądarki: np. `location`, `navigator`;
- nazwy specjalne okien:  
**`top`** -> główne okno,  
**`self`** -> bieżące okno;

# Ograniczenia JavaScript

- Niemożność ukrycia kodu źródłowego: problem ochrony własności intelektualnej;
- Niedostatki w zakresie międzyplatformowego debuggera;
- Ograniczenia:
  - drukowanie i dostęp do lokalnych plików;
  - bezpośredni dostęp do plików na serwerze WWW.



# Aplety

- Trzy zasadnicze komponenty tej technologii:
  - język programowania;
  - system wykonawczy;
  - biblioteka klas;
- Możliwość wykonywania operacji na plikach;
- Możliwość cyfrowego podpisania apletu celem udostępnienia mu operacji uprzywilejowanych;
- Większa skalowalność, możliwość tworzenia bardziej wyrafinowanego kodu;
- Udostępnianie apletu:
  - bezpośrednio wskazanie w URL pliku .class zawierającego aplet;
  - osadzenie apletu w dokumencie HTML:  

```
<APPLET CODEBASE="serwer/scieżka"  
CODE="nazwaApletu.class">
```

# Cookies - zastosowanie

- Plik tekstowy przechowywany po stronie klienta. Tradycyjny termin informatyczny oznaczający nieprzejrzysty element danych przechowywany przez pośrednika.
- Zastosowanie:
  - kto odwiedza stronę? Ilu jest użytkowników, ilu nowych?
  - personalizacja strony;
  - zachowanie stanu współpracy ze stroną;
  - umieszczenie identyfikatora “koszyka”;
- Ciasteczka stają się użyteczne dopiero przy wykorzystaniu skryptów (po stronie klienta lub po stronie serwera)

# Cookies - ograniczenia

- Oparcie na protokole HTTP => każda para żądanie – odpowiedź jest niezależna:
  - Ciasteczko może być założone, usunięte lub zmienione na skutek odpowiedzi serwera, zaś zawarta w nim informacja zostanie przekazana przy następnym wezwaniu.
  - Stąd dopiero przy następnym wezwaniu jesteśmy w stanie zorientować się, że użytkownik nie przyjmuje ciasteczek.
- Serwer może zażądać tylko ciasteczka, które sam zapisał u klienta (zgodność domeny w żądaniach).
- Pierwotna specyfikacja RFC 2109 określała ograniczenie na liczbę przechowywanych ciasteczek: maks. 300 łącznie, maks. 20 z danej domeny;
- Główne ograniczenie – możliwość wyłączenia obsługi.

# Parametry ciasteczka

- Znaki niedrukowane, jak i pewne znaki zarezerwowane (“=” czy “;”) muszą zostać zapisane w postaci heksadecymalnej, poprzedzonej symbolem „%”. Wymagane w treści ciasteczka, tak samo jak w URL (zob. funkcja `escape()` w JavaScript).
- Własność `expires` – sposoby użycia:
  - odległa przyszła data => ciasteczko trwałe;
  - wybrana data przyszła => terminowe;
  - brak własności `expires` => do momentu zamknięcia okna przeglądarki;
  - data przeszła => usunięcie ciasteczka.
- Własności `path` i `server` => ograniczenie dostępu do informacji zawartej w ciasteczku. Muszą być zgodne co do ostatnich dwóch lub trzech członów z adresem twórcy ciasteczka. W przypadku braku jest ustawiany domyślnie najbardziej specyficzny zakres.
- `security = secure` => ciasteczko przesyłane tylko wtedy, jeśli ustanowione połączenie jest bezpieczne.

# Obsługa cookie w JavaScript

- Poprzez dostęp do właściwości `document.cookie`:  
jeśli mamy przygotowane parametry *nazwa*, *wartość*  
i datę *ważneDo*, to możemy ustawić:

```
document.cookie = nazwa+"="+escape(wartość)  
+ ";expire="+ ważneDo.toGMTString();
```

- Przy pobieraniu wartości użyteczne będą:
  - `document.cookie.length`;
  - Operacje na tekście:  
`indexOf(tekst)`, `indexOf(tekst, offset)`;  
`substring(pocz, koniec)`, `unescape(tekst)`.

# Priorytety operatorów

L.p.	Operatory	Symbole
1	indeks tablicy, wywołanie funkcji	[], ()
2	inkrementacja i dekrementacja, ustalenie znaku, negacja bitowa i logiczna, utworzenie obiektu, ustalenie typu zmiennej, usunięcie składowej	++, --, +, -, ~, !, new, typeof, delete
3	mnożenie, dzielenie, reszta z dzielenia	*, /, %
4	dodawanie, odejmowanie	+, -
5	przesunięcie bitowe w lewo, w prawo, w prawo z wypełnieniem zerami	<<, >>, >>>
6	mniejsze, większe, mniejsze lub równe, większe lub równe, porównanie typów	<, >, <=, >=, instanceof
7	równe, różne	==, !=
8	iloczyn bitowy	&
9	bitowa różnica symetryczna	^
10	suma bitowa	
11	iloczyn logiczny	&&
12	suma logiczna	
13	warunkowy	? :
14	operatory przypisania	=, +=, -=, *=, /=, %=, &=, ^=,  =, <<=, >>=, >>>=
15	rozdzielanie wyrażeń	.

# Zadania

- **Zadanie 1.** Napisać program, który dla wczytanej liczby całkowitej wypisuje, czy jest ona parzysta czy nieparzysta.
- **Zadanie 2.** Napisać program, który po wczytaniu dwóch liczb całkowitych odpowiada, czy jedna jest wielokrotnością drugiej.
- **Zadanie 3.** Napisać program, który dla trzech wczytanych liczb całkowitych odpowiada, czy mogą być one długościami boków trójkąta (a) prostokątnego, (b) równoramiennego, (c) równobocznego.

# Bibliografia

- <http://javascript.skryptoteka.pl>
- <http://pl.wikipedia.org/wiki/JavaScript>