OBIEKTY W JAVASCRIPT

WPROWADZENIE

- JavaScript jest językiem obiektowym
- Oferuje wbudowane obiekty oraz umożliwia definiowanie własnych obiektów
- Obiekt jest specjalnym rodzajem danych, mającym metody i właściwości
- Właściwości – wartości związane z obiektem (np. długość dla obiektu txt)
- Metody – działania, które mogą być wykonane na obiekcie (np. metoda toUpperCase dla obiektu str)

OBIEKT STRING

- Obiekt reprezentujący fragmenty tekstu
- Aby utworzyć obiekt typu String wystarczy zadeklarować zmienną tekstową:

```
var nazwa_zmiennej = "łańcuch_znaków"
nazwa_zmiennej = "łańcuch_znaków"
nazwa = new String("łańcuch_znaków")
```

- Przykład:

```
var tekst1 = "Jakiś tekst."
tekst2 = "Inny tekst !"
tekst3 = new String("Utworzenie nowego obiektu.")
```

- Korzystanie z właściwości obiektu

```
nazwa_zmiennej.właściwość
```

- Korzystanie z metod

```
nazwa_zmiennej.nazwa_metody(lista_argumentów)
```

METODY OBIEKTU STRING

- Zmiana stylu:
  - **big()** –powiększona czcionka
  - **small()** – pomniejszona czcionka
  - **bold()** – wytłuszczona czcionka
  - **italics()** – kursywa
  - **blink()** – tekst migający (nie działa w IE)
  - **fixed()** – czcionka o stałej szerokości
  - **strike()** – przekreślona czcionka
  - **fontcolor(*kolor*)** – zmiana koloru czcionki
    (*kolor* = nazwa koloru|rgb(n,n,n)|#xxxxxx)
  - **fontsize(*rozmiar_czcionki*)** – zmiana rozmiaru czcionki
  - **toLowerCase()** – małe litery
  - **toUpperCase()** – wielkie litery
  - **sub()** – indeks dolny
  - **sup()** – indeks górny
  - **link(*url*)** – tekst będzie wyświetlony jako hiperłącze do dokumentu o adresie *url*

METODY OBIEKTU STRING – C.D.

- Wyszukiwanie:
  - charAt(*pozycja*) –zwraca znak występujący na określonej pozycji (pierwsza pozycja to 0)
  - charCodeAt(*pozycja*) – zwraca kod Unicode znaku występującego na określonej pozycji
  - indexOf(*tekst*, *pozycja_początkowa*) – zwraca pozycję, od której zaczyna się *tekst,* poszukiwanie rozpoczyna się od *pozycji_początkowej* (wielkość liter jest istotna, gdy *tekst* nie występuje zwracana jest wartość -1, drugi argument opcjonalny)
  - lastIndexOf(*tekst*, *pozycja_końcowa*) – zwraca pozycję, od której zaczyna się *tekst*, tym razem poszukiwanie wstecz od *pozycji_końcowej* (wielkość liter jest istotna, gdy *tekst* nie występuje w danym łańcuchu zwracana jest wartość -1, drugi argument opcjonalny)
  - match(*tekst*) – podobne do indexOf(), ale zamiast pozycji zwraca *tekst* (null jeśli *tekst* nie występuje w ciągu, wielkość liter ma znaczenie)
  - search(*tekst*) – wyszukuje w ciągu wzorca *tekst* i podobnie jak indexOf zwraca pozycję pierwszego wystąpienia *tekst* (wartość "-1" jeśli *tekst* nie występuje w ciągu, wielkość liter istotna, ale można ją zignorować ustawiając flagę **i**, wówczas argument *tekst* umieszcza się w parze znaków / a nie " i a po nich flagę **i**

METODY OBIEKTU STRING - C.D.

- Zamiana tekstu:
  - **fromCharCode(*liczba_1*, *liczba_2*, ... , *liczba_n*)** – zwraca łańcuch znaków powstały z zamiany ciągu wartości Unicode na odpowiadające im znaki (metoda jest statyczna, co oznacza, że jedynym obiektem, dla którego może być wywołana, jest sam obiekt String)
  - **replace(*stary_tekst, nowy_tekst*)** – zamiana w danym ciągu podciągu *stary_tekst* na *nowy_tekst* (wielkość liter jest istotna, ustawienie flagi **i** zmienia sposób interpretacji wielkości liter, flaga **g** włącza zastępowanie wszystkich wystąpień podciągu *stary_tekst* w łańcuchu)
- Wyodrębnianie podciągu znaków:
  - **slice(*pozycja_początkowa*,*pozycja_końcowa*)** – wycinanie fragmentu tekstu od *pozycja_początkowa* do *pozycja_końcowa* (obydwa argumenty numeryczne, wartość ujemna oznacza pozycję od końca, musi być spełniony warunek *pozycja_początkowa* < *pozycja_końcowa*, drugi argument opcjonalny – brak oznacza, że *pozycja_końcowa* = koniec łańcucha)
  - **substr(*pozycja_początkowa*, *długość*)** - wycina z łańcucha liczbę znaków *długość* począwszy od *pozycja_początkowa (*ujemna wartość pierwszego argumentu oznacza pozycję od końca, drugi argument opcjonalny – brak= koniec łańcucha)
  - **subsrtring(*pozycja_początkowa*,*pozycja_końcowa*)** – podobnie jak slice
  - **split(*separator, liczba*)** – podzelenie łańcucha znaków na tablicę ciągów, pierwszy argument jest znakiem, który rozgranicza ciągi w łańcuchu znaków (np. "." przy podziale na zdania, " " przy podziale na wyrazy, "" przy podziale na litery, itd.) , drugi opcjonalny argument służy do określenia liczby ciągów

POZOSTAŁE METODY OBIEKTU STRING

- Łączenie tekstów:
  - **concat(*tekst1*, *tekst2*, ...)** – dodaje do obiektu łańcuchy znakowe będące argumentami metody (musi wystąpić co najmniej jeden argument)
- Inne metody:
  - **anchor(*nazwa_kotwicy*)** – ustawienie kotwicy o nazwie *nazwa_kotwicy* w pliku HTML
    - Przykład:
      - skrypt:
        ```
        <script type="text/javascript">
        var tekst="Jakiś tekst."
        document.write(tekst.anchor("kotwica1"))
        </script>
        ```
      - jest równoważny zapisowi HTML:
        ```
        <a name="kotwica1"> Jakiś tekst. </a>
        ```
  - **toSource()** – reprezentuje kod źródłowy obiektu (nie działa w IE)
  - **valueOf()** – zwraca prymitywną wartość obiektu String, wartość dzieciczona przez wszystkie obiekty potomne obiektu String, metoda zazwyczaj wywoływana automatycznie przez JavaScript

WŁAŚCIWOŚCI OBIEKTU STRING

- **length** – długość łańcucha
- **constructor** – referencja do funkcji tworzącej obiekt
- **prototype** – pozwala dodać do obiektu własne metody i właściwości
  - Składnia:

    obiekt.prototype.*nazwa* = *wartość*

OBIEKT DATE

- Obiekt Date służy do przechowywania informacji o dacie i godzinie oraz wykonywaniu na nich określonych operacji
- Definowanie obiektu przy użyciu słowa kluczowego **new**
  - Przykład:

    ```
    var Data = new Date()
    ```
- Początkowa wartość nowego obiektu to bieżąca data i bieżący czas
- Dodawanie wartości do dowolnej składowej daty (rok, miesiąc, dzień, godzina, minuty, sekundy, milisekundy) powoduje również uaktualnienie pozostałych składowych

METODY OBIEKTU DATE

- Odczyt bieżącego czasu i bieżącej daty:
  - **Date()** – podaje bieżące: datę i czas (składnia: tylko metoda bez obiektu)
- Odczyt daty:
  - **getDate()** – zwraca liczbę z zakresu 1÷31 oznaczającą dzień miesiąca
  - **getDay()** – zwraca liczbę z zakresu 0÷6 oznaczającą dzień tygodnia (0 – niedziela, 1 – poniedziałek, ...)
  - **getMonth()** – zwraca liczbę z zakresu 0÷11 oznaczającą miesiąc (0 – styczeń, 1 – luty, ...)
  - **getYear()** – zwraca dwu (między 1900 a 1999) lub czterocyfrowe oznaczenie roku (przed 1900 lub po 1999) – nie jest zalecana
  - **getFullYear()** – zwraca czterocyfrową liczbę reprezentującą rok
- Odczyt czasu:
  - **getHours()** – zwraca godzinę (liczba z zakresu 0÷23)
  - **getMinutes()** – zwraca minuty (liczba z zakresu 0÷59)
  - **getSeconds()** – zwraca sekundy (liczba z zakresu 0÷59)
  - **getMiliseconds()** – zwraca milisekundy (liczba z zakresu 0÷999)
  - **getTime()** – zwraca liczbę milisekund od 1 stycznia 1970
  - **getTimezoneOffset()** – różnica w minutach pomiędzy czasem lokalnym a GMT

METODY OBIEKTU DATE – C.D.

- Ustawianie daty:
  - **setDate(*dzień*)** – ustawienie dnia miesiąca (1÷31)
  - **setMonth(*miesiąc, dzień*)** – ustawienie miesiąca (0÷11), argument *dzień* opcjonalny
  - **setFullYear(*rok, miesiąc, dzień*)** – ustawienie roku (liczba czterocyfrowa), argumenty *miesiąc* i *dzień* opcjonalne
  - **setYear(*rok*)** – ustawienie roku (liczba dwu lub czterocyfrowa (metoda nie zalecana)
  - **setHours(*godzina, minuty, sekundy, milisekundy*)** – ustawienie godziny, argumenty 2, 3 i 4 są opcjonalne
  - **setMinutes(*minuty, sekundy, milisekundy*)** – ustawienie minut, argumenty 2 i 3 opcjonalne
  - **setSeconds(*sekundy, milisekundy*)** – ustawienie sekund, argument 2 opcjonalny
  - **setMiliseconds(*milisekundy*)** – ustawienie milisekund
  - ***setTime(liczba_milisekund)*** – ustawienie daty i czasu względem 1 stycznia 1970 o północy poprzez dodanie lub odjęcie (wartość ujemna) *liczby_milisekund*

METODY OBIEKTU DATE – C.D.

- Metody odczytywania i ustawiania czasu według UTC (Universal Coordinated Time) czas ustawiony przez World Time Standard:
  - getUTCDate()
  - getUTCDay()
  - getUTCMonth()
  - getUTCFullYear()
  - getUTCHours()
  - getUTCMinutes()
  - getUTCSeconds()
  - getUTCMiliseconds()
  - setUTCDate(*dzień*)
  - setUTCMonth(*miesiąc, dzień*)
  - setUTCFullYear(*rok, miesiąc, dzień*)
  - setUTCHours(*godzina, minuty, sekundy, milisekundy*)
  - setUTCMinutes(*minuty, sekundy, milisekundy*)
  - setUTCSeconds(*sekundy, milisekundy*)
  - setUTCMiliseconds(*milisekundy*)
- Znaczenie poszczególnych metod podobne do analogicznych metod przedstawionych na poprzednich slajdach

POZOSTAŁE METODY OBIEKTU DATE

- **parse(*data*)** – na podstawie łańcucha *data* metoda wyznacza liczbę milisekund pomiędzy *data* a 1 stycznia 1970 o północy (*data* jest łańcuchem znaków reprezentujących datę np. "Jan 07, 2007")
- **UTC(*rok*, *miesiąc*, *dzień*, *godzina*, *minuty*, *sekundy*, *milisekundy*)** – zamienia datę na liczbę milisekund od północy 1 stycznia 1970 według UTC (trzy pierwsze argumenty obowiązkowe, cztery ostatnie opcjonalne)
- **toString()** – zamienia datę na łańcuch znaków, który zwracany jest jako wynik np."Sun Jan 07 2007 15:14:42 GMT+0100"
- **toGMTString()** – zamienia datę wg czasu GMT i zmienia na łańcuch znaków (nie zalecane) np."Sun, 07 Jan 2007 14:14:42 GMT"
- **toUTCString()** – zamienia datę wg czasu UTC a następnie na łańcuch znaków "Sun, 07 Jan 2007 14:14:42 GMT"
- **toLocaleString()** – zamienia datę wg czasuu lokalnego i zamienia go na łańcuch znaków "7 styczeń 2007 15:14:42"
- **toSource()** – reprezentacja kodu źródłowego obiektu (nie działa w IE)
- **valueOf()** – zwraca prymitywną wartość obiektu Date podobnie jak dla obiektu String

WŁAŚCIWOŚCI OBIEKTU DATE

- **constructor** – właściwość będąca referencją do funkcji tworzącej obiekt
- **prototype** – właściwość umożliwiająca dodawanie nowych metod i właściwości

OBIEKT ARRAY

- Służy do zapamiętania tablicy, czyli zbioru wartości pod jedną zmienną
- Definiowanie tablicy:
  - sposób 1 (bez ograniczania liczby elementów):
    - var Tablica = new Array()
    - Tablica[0] = "element 1"
    - Tablica[1] = "element 2"
    - Tablica[2] = "element 3"
  - sposób 2 (z definiowaniem rozmiaru tablicy):
    - var Tablica = new Array(3)
    - Tablica[0] = "element 1"
    - Tablica[1] = "element 2"
    - Tablica[2] = "element 3"
  - sposób 3
    - var Tablica = new Array("element 1", "element 2", "element 3")
- Elementy tablicy mogą być typu łańcuchowego, logicznego (true i false), numerycznego lub innymi obiektami
- Dostęp do tablicy poprzez jej elementy np. Tablica[0]
- Modyfikacje wartości poprzez odwołanie się do elementów tablicy

  np. Tablica[0] = "nowa wartość"
- Do wszystkich elementów tablicy można odwołać się po kolei używając pętli **for ... in**

METODY OBIEKTU ARRAY

- Łączenie:
  - **concat(*tab1*, *tab2*, ...)** – zwraca połączone ze sobą tablicę, dla której metoda jest wywołana, i tablice (co najmniej jedną) będące argumentami metody (tablice wejściowe nie ulegają żadnym modyfikacjom podczas wywołania tej metody)
  - **join(*separator*)** – łączy elementy tablicy w jeden łańcuch rozdzielając je znakiem *separator* (argument opcjonalny – domyślnie przecinek)
  - **toString()** – konwertuje tablicę na łańcuch oddzielając elementy przecinkiem
- Porządek:
  - **reverse()** – odwraca porządek elementów w tablicy (modyfikuje oryginalną tablicę)
  - **sort(*sposób_sortowania*)** – porządkuje elementy tablicy alfabetycznie lub zgodnie z funkcją użytkownika *sposób_sortowania* (argument opcjonalny) np. sortowanie numeryczne:

```
function sortowanieNumeryczne(a, b)
{ return a - b}
```

METODY OBIEKTU ARRAY – C.D.

- Operacje na elementach:
  - **pop()** – usuwa i zwraca ostatni element tablicy (zmienia rozmiar tablicy)
  - **shift()** – usuwa i zwraca pierwszy element tablicy (zmienia rozmiar i zawartość tablicy)
  - **push(*nowy_element_1*, *nowy_element_2*, ...)** – dodaje na końcu tablicy 1 lub więcej nowych elementów i zwraca nowy rozmiar tablicy (zmianie ulega rozmiar tablicy, pierwszy argument obowiązkowy, pozostałe opcjonalne)
  - **unshift(*nowy_element_1*, *nowy_element_2*, ...)** – dodaje na początku tablicy 1 lub więcej nowych elementów i zwraca nowy rozmiar tablicy (zmianie ulega rozmiar tablicy, pierwszy argument obowiązkowy, pozostałe opcjonalne, nie działa poprawnie w IE)
  - **slice(*indeks_początkowy, indeks_końcowy*)** – metoda zwraca wybrane elementy tablicy pomiędzy *indeksem_początkowym* a *indeksem_końcowym* (pierwszy argument obowiązkowy, drugi opcjonalny, ujemne wartości oznaczają pozycję od końca, domyślny *indeks_końcowy* to koniec tablicy)
  - **splice(*indeks*, *liczba*, *nowy_element_1*, *nowy_element_2*, ...)** – metoda usuwa *liczbę* elementów tablicy począwszy od elementu o indeksie *indeks* i wstawia od tego miejsca wszystkie wymienione jako argumenty *nowe_elementy* (tylko dwa pierwsze argumenty są obowiązkowe)

POZOSTAŁE METODY OBIEKTU ARRAY

- **toSource()** i **valueOf()** – takie samo znaczenie jak dla obiektu String

WŁAŚCIWOŚCI OBIEKTU ARRAY

- **length** – do ustawia i zwracania rozmiaru tablicy
- **constructor** – jak w innych obiektach
- **prototype** – jak w innych obiektach
- **index**
- **input**

OBIEKT BOOLEAN

- Obiekt używany do konwersji dowolnych wartości na wartości logiczne (true, false)
- Definiowanie obiektu:

  var obiektBoolean = new Boolean()

  - brak wartości początkowej lub wartość początkowa: 0, -0, null, "", false, undefined oraz NaN oznaczają wartość false
  - pozostałe wartości (w tym łańcuch "false" traktowane są jak wartość true

METODY I WŁAŚCIWOŚCI OBIEKTU BOOLEAN

- Metody:
  - **toString()** – metoda zamienia wartość logiczną na łańcuch "true" lub "false", który zwracany jest jako wynik (metoda wywoływana automatycznie, gdy wartość logiczna musi być podana w postaci łańcucha znaków)
  - **toSource()**
  - **valueOf()**
- Właściwości:
  - **constructor**
  - **prototype**

OBIEKT MATH

- Obiekt Math umożliwia wykonywanie podstawowych operacji matematycznych
- Oferuje kilka stałych i funkcji matematycznych, które mogą być używane bez definiowania obiektu (metody statyczne obiektu Math np. Math.abs(-3.27))

METODY OBIEKTU MATH

- Funkcje arytmetyczne:
  - abs(x) – zwraca wartość bezwzględną z x
  - ceil(x) – zaokrąglenie x w górę do najbliższej liczby całkowitej
  - floor(x) – zaokrąglenie x w dół do najbliższej liczby całkowitej
  - round(x) – zaokrąglenie x do najbliższej liczby całkowitej
  - exp(x) - e^x
  - log(x) – logarytm naturalny z x
  - max(x,y) – maksimum z x i y
  - min(x,y) – minimum z x i y
  - pow(x,y) – x^y
  - sqrt(x) – pierwiastek kwadratowy z x

METODY OBIEKTU MATH – C.D.

- Funkcje trygonometryczne:
  - cos(x) – cosinus z x (wartości od -1 do 1)
  - sin(x) – sinus z x (wartości od -1 do 1)
  - tan(x) – tangens z x
  - acos(x) – arcus cosinus z $x \in [-1,1]$ (wartości od 0 do π radianów, wartość NaN dla argument spoza przedziału a dla acos(-1) = π
  - asin(x) – arcus sinus z $x \in [-1,1]$ (wartości między $-\pi/2$ a $\pi/2$, wartość NaN dla argumentu spoza przedziału, asin(1)= $\pi/2$)
  - atan(x) – arcus tangens z x (wartości między $-\pi/2$ a $\pi/2$)
  - atan2(x,y) – kąt θ punktu (x,y) jako numeryczna wartość między $-\pi$ a π

POZOSTAŁE METODY OBIEKTU MATH

- Funkcje losowe:
  - random() – losuje liczby z przedziału 0-1
- Metody standardowe:
  - toSource()
  - valueOf()

WŁAŚCIWOŚCI OBIEKTU MATH

- Stałe matematyczne:
  - E – stała Eulera ($e \cong 2{,}718$)
  - LN2 – $\ln 2 \cong 0{,}693$
  - LN10 – $\ln 10 \cong 2{,}302$
  - LOG2E – $\log_2 e \cong 1{,}414$
  - LOG10E – $\log_{10} e \cong 0{,}434$
  - PI - $\pi \cong 3{,}14159$
  - SQRT1_2 – pierwiastek kwadratowy z ½ $\cong 0{,}707$
  - SQRT2 – pierwiastek kwadratowy z 2 $\cong 1{,}414$
- Właściwości standardowe:
  - constructor
  - prototype

WŁAŚCIWOŚCI NAJWYŻSZEGO POZIOMU

- Mogą być używane w połączeniu z wszystkimi wbudowanymi obiektami JS
- Właściwości:
  - Infinity
    - wartość numeryczna reprezentująca +∞ lub - ∞
    - Infinity > 1.7976931348623157E+10308
    - -Infinity < -1.7976931348623157E+10308
    - cokolwiek podzielone przez Infinity daje 0
    - cokolwiek pomnożone przez Infinity daje Infinity
  - NaN – Not a Number
    - w przypadku metod operujących na liczbach NaN oznacza wartość, która liczbą nie jest
  - undefined
    - oznacza zmienną, której nie przypisano żadnej wartości

FUNKCJE WYSOKIEGO POZIOMU

- Kodowanie i dekodowanie:
  - **encodeURI(*łańcuch_URI*)** – zamienia *łańcuch_URI* na adres URI kodując znaki specjalne oprócz: , / ? : @ & = + $ #
  - **encodeURIComponent(*łańcuch_URI*)** – zamienia *łańcuch_URI* na adres URI kodując wszystkie znaki specjalne
  - **decodeURI(*URI*)** – dekoduje *URI* zakodowany przez encodeURI()
  - **decodeURIComponent(*URI*)** – dekoduje *URI* zakodowane przez encodeURIComponent()
  - **escape(*łańcuch*)** – koduje *łańcuch* znaków oprócz znaków specjalnych: * @ - _ + . / (nie powinna być używana do kodowania URI)
  - **unescape(*łańcuch*)** – dekoduje łańcuch zakodowany przez escape()
- Testowanie:
  - **isFinite(*liczba*)** – sprawdza, czy argument jest liczbą skończoną (zwraca wartość logiczną)
  - **isNaN(*liczba*)** – sprawdza, czy argument nie jest wartością liczbową (zwraca wartość logiczną)

FUNKCJE WYSOKIEGO POZIOMU – C.D.

- Konwersja:
  - **eval(*łańcuch*)** – analizuje *łańcuch* i traktuje go, jakby był fragmentem kodu
  - **Number(*obiekt*)** – zamienia wartość *obiektu* na wartość numeryczną (dla obiektu Date jest to liczba milisekund od 1 stycznia 1970 godz 0:00:00 UCT, jeśli wartości nie da się zamienić na liczbę, to zwracana jest wartość NaN)
  - **String(*obiekt*)** – zamienia wartość obiektu na łańcuch (zwraca taką samą wartość jak metody toString() dla poszczególnych obiektów)
  - **parseFloat(*łańcuch*)** – parsuje *łańcuch* i zwraca liczbę zmiennoprzecinkową lub NaN (znajdowany jest pierwszy znak nie będący białą spacją, jeśli nie jest on cyfrą to zwracana jest wartość NaN, w przeciwnym przypadku łańcuch jest analizowany do końca liczby i jest ona zwracana przez funkcję, tylko pierwsza liczba jest zwracana przez funkcję)
  - **parseInt(*łańcuch, podstawa*)** - parsuje *łańcuch* i zwraca liczbę całkowitą lub NaN (znajdowany jest pierwszy znak łańcucha nie będący białą spacją, jeśli nie jest on cyfrą, to zwracana jest wartość NaN, jeśli jest cyfrą to analizowane są kolejne cyfry, aż do końca liczby, która jest zwracana jako wartość funkcji; opcjonalny argument *podstawa* – liczba z zakresu 2 do 32 - oznacza system numeryczny, w którym zapisana jest liczba; gdy brak argumentu *podstawa* stosowane są reguły: początek 0x oznacza system szesnastkowy, początek 0 oznacza system ósemkowy, inne wartości oznaczają system dziesiętny)

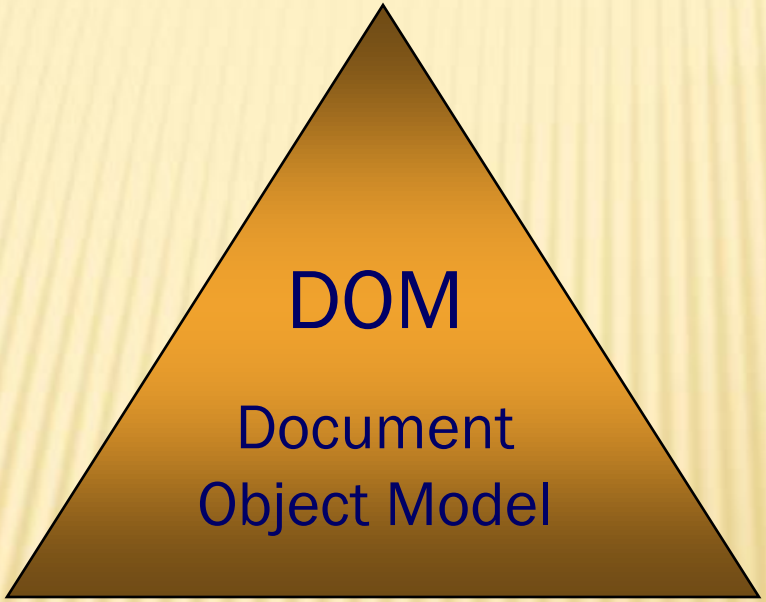
HTML DOM

HTML DOM - WPROWADZENIE

- HTML DOM = Document Object Model for HTML
- HTML DOM – standard W3C
- Określa standardowy zbiór obiektów dla HTML-a oraz standardowy sposób dostępu i przetwarzania dokumentów HTML
- Elementy HTML wraz z ich zawartością i atrybutami są dostępne poprzez DOM i mogą być dodawane, usuwane, zmieniane lub przesuwane.
- HTML DOM jest niezależny od platformy sprzętowo-programowej i języka programowania
- Różne części DOM:
  - Core DOM – standardowy zbiór obiektów dla dowolnego dokumentu strukturalnego
  - XML DOM – standardowy zbiór obiektów dla dokumentu XML
  - HTML DOM – standardowy zbiór obiektów dla dokumentu HTML
- Trzy poziomy DOM (DOM Level 1/2/3)

SCHEMAT APLIKACJI WEB PO STRONIE KLIENTA

Struktura – język znaczników
(HTML, XML, XHTML)



Interakcja
(JavaScript, VBScript,
pluginy, applety, ...)

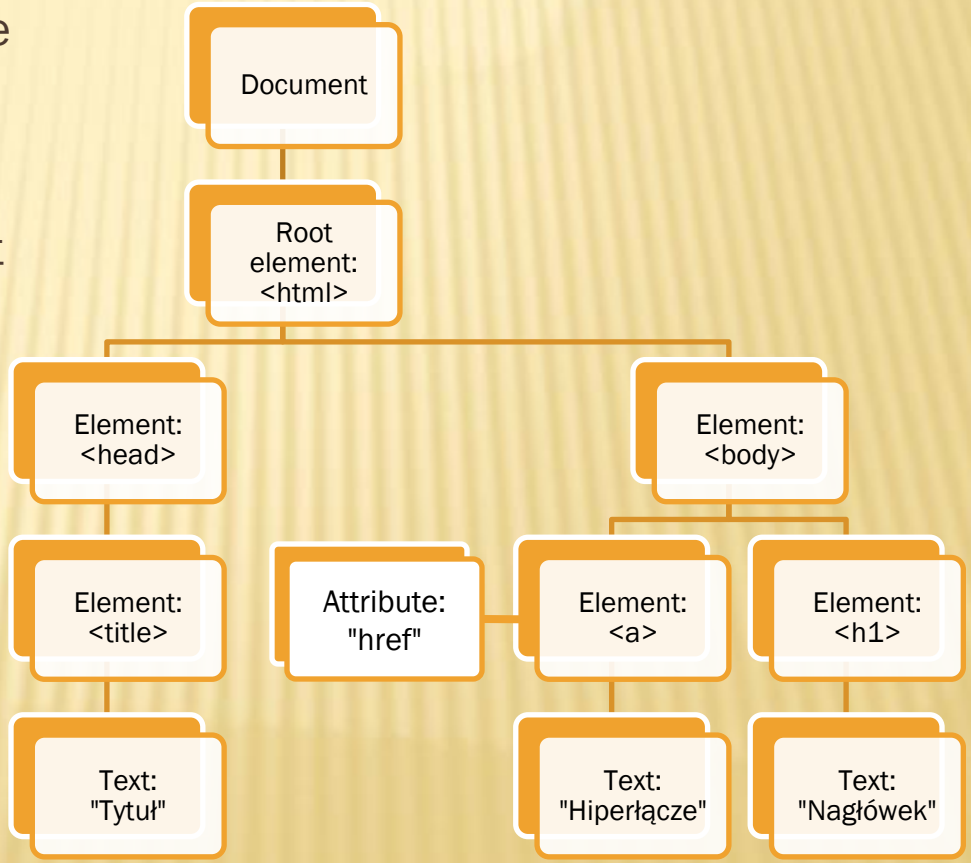
Prezentacja
(HTML, CSS, FLASH, ...)

DRZEWIASTA STRUKTURA DOKUMENTU HTML

- Dokument HTML wraz z jego elementami, atrybutami i wartościami w modelu DOM jest przedstawiany w postaci drzewa

Według DOM wszystko w dokumencie HTML jest węzłem:

- dokument – węzeł document
- każdy znacznik – węzeł element
- tekst w znacznikach – węzeł text
- każdy atrybut – węzeł attribute
- komentarze – węzeł comment



ZALEŻNOŚCI MIĘDZY WĘZŁAMI

- Wszystkie węzły są ze sobą powiązane
- Każdy węzeł (oprócz document) ma rodzica
- Większość węzłów ma co najmniej jedno dziecko
- Węzły są rodzeństwem, jeśli mają tego samego rodzica
- Potomkami węzła są jego dzieci, dzieci tych dzieci, itd.
- Poprzednikami węzła są jego rodzice, rodzice tych rodziców, itd.

DOSTĘP DO WĘZŁÓW

- Metody:
  - getElementById("id")
    - zwraca element o określonym ID
    - składnia: document.getElementById("id")
  - getElementsByTagName("znacznik")
    - zwraca wszystkie elementy w postaci listy węzłów
    - składnia: document.getElementByTagName("znacznik") lub document.getElementById("id").getElementByTagName("znacznik")
  - obydwie metody:
    - potrafią znaleźć w całym dokumencie dowolny element HTML
    - ignorują strukturę dokumentu
- Lista węzłów:
  - zazwyczaj wynik metody getElementsByTagName() zapamiętywany jest pod jakąś zmienną, która jest raktowana jak tablica (możliwość dostępu do wybranych wystąpień znacznika, ale także możliwość dostępu do wszystich w pętli for)
- Właściwości:
  - parentNode – na ogół używany do zmiany struktury dokumentu np. usunięcie węzła x.parentNode.removeChild(x)
  - firstChild oraz lastChild – przeważnie używane do zmiany wartości
  - document.documentElement – zwraca węzeł główny dokumentu
  - document.body – umożliwia bezpośredni dostęp do znacznika <body>

INFORMACJE O WĘZŁACH

- Właściwości:
  - nodeName – nazwa węzła:
    - dla elementu – nazwa znacznika
    - dla atrybutu – nazwa atrybutu
    - dla tekstu – wartość: #text
    - dla dokumentu – wartość: #document
  - nodeValue – wartość węzła
    - dla tekstu – ten tekst
    - dla atrybutu – wartość atrybutu
    - właściwość niedostępna dla dokumentu i elementów
  - nodeType – typ węzła
    - element – 1
    - atrybut – 2
    - tekst – 3
    - komentarz – 8
    - dokument - 9

PRZYKŁAD – ZMIANA KOLORU TŁA

```
<html>
<head>
<script type="text/javascript">
function ZmianaKoloru()
{
document.body.bgColor="yellow"
}
</script>
</head>
<body onclick="ZmianaKoloru()">
Kliknij w dowolnym miejscu!
</body>
</html>
```

OBIEKT WINDOW

- Obiekt wysokiego poziomu reprezentujący okno przeglądarki
- Tworzony automatycznie z każdym wystąpieniem znaczników <body> lub <frameset>
- Kolekcje:
  - frames[] – zwraca wszystkie nazwane ramki okna

WŁAŚCIWOŚCI OBIEKTU WINDOW

- **name** – nazwa okna [window.name="nazwa"]
- **status** – tekst zawarty na pasku stanu [window.status="tekst"] (nie działa w Firefoxie)
- **defaultStatus** – domyślny tekst paska stanu, który pojawi się podczas ładowania strony [window.defaultStatus="tekst] (nie działa w Firefoxie)
- **opener** – referencja do okna, z którego otworzono dane okno [window.opener]
- **top** – referencja do okna stojącego najwyżej w hierarchii [window.top]
- **self** – referencja do bieżącego okna [window.self]
- **parent** – okno rodzic [window.parent]
- **document** – obiekt Document [window.document. ...]
- **history** – obiekt History [window.history. ...]
- **location** – obiekt Location [window.location. ...]
- **closed** – logiczna wartość określająca, czy okno zostało zamknięte [window.closed]
- **length** – liczba ramek w oknie [window.length="liczba"]

POZOSTAŁE WŁAŚCIWOŚCI OBIEKTU WINDOW

- outerheight – zewnętrzna wysokość okienka, łącznie ze wszystkimi elementami interfejsu [window.outerheight="liczba_pikseli"] (nie działa)
- outerwidth – zewnętrzna szerokość okienka, łącznie ze wszystkimmi elementami interfejsu [window.outerwidth="liczba.pikseli] (nie działa)
- pageXOffset – pozycja X bieżącej strony względem lewego górnego narożnika obszaru wyświetlania okna [window.pageXOffset="liczba_pikseli"] (nie działa)
- pageYOffset – pozycja Y bieżącej strony względem lewego górnego narożnika obszaru wyświetlania okna [window.pageYOffset="liczba_pikseli"] (nie działa)
- scrollbars – logiczna wartość określająca widoczność pasków przewijania [window.scrollbars="true/false"] (nie działa)
- toolbar – logiczna wartość określająca widoczność paska narzędziowego [windows.toolbar="true/false"] (nie działa)
- statusbar – logiczna wartość określająca widoczność paska stanu [windows.statusbar="true/false"] (nie działa)
- personalbar – logiczna wartość określająca widoczność paska osobistego (katalogów) [windows.personalbar="true/false"]

METODY OBIEKTU WINDOW

- Otwieranie okna:
  - open()
  - składnia: window.open(URL, nazwa, opcje, historia)
    - URL (opcjonalny) – adres nowej strony (domyślnie about:blank)
    - nazwa (opcjonalny) – nazwa okna lub jedna z wartości:
      - _blank (domyślnie) – strona ładowana w nowym oknie
      - _parent – strona ładowana w ramce rodzica
      - _self – strona zastępuje bieżącą stronę
      - _top – zastępuje frameset
    - opcje (opcjonalny) – lista opcji oddzielanych przecinkami
      - channelmode=yes|no|1|0 – okno w kinowym trybie wyświetlania
      - fullscreen=yes|no|1|0 – okno w trybie pełnoekranowym (musi być również w kinowym)
      - directories=yes|no|1|0 – przyciski paska katalogów
      - menubar=yes|no|1|0 – pasek menu
      - status=yes|no|1|0 – pasek stanu
      - titlebar=yes|no|1|0 – pasek tyułowy (działanie w zależności od miejsca wywołania)
      - toolbar=yes|no|1|0 – pasek narzędziowy
      - location=yes|no|1|0 – pole adresu
      - scrollbars=yes|no|1|0 – paski przewijania
      - height=piksele – wysokość okna (min. wartość 100)
      - width=piksele – szerokość okna (min. wartość 100)
      - left=piksele – pozycja okna od lewej krawędzi ekranu
      - top=piksele – pozycja okna od górnej krawędzi ekranu
      - resizable=yes|no|1|0 – możliwość zmiany wielkości okna
    - historia (opcjonalny) – czy URL strony ma utworzyć nowy wpis w historii, czy też zastąpić bieżący
      - true – zastępowanie
      - false – nowy wpis

METODY OBIEKTU WINDOW – C.D.

- Zamykanie okna:
  - **close()** – zamyka bieżące okno [window.close()]
- Okienka wyskakujące:
  - **alert()** – okienko informacyjne (tylko przycisk OK) [alert("tekst_komunikatu")]
  - **confirm()** – okienko dialogowe z przyciskami OK i Anuluj – zwraca true gdy wybrano OK, false w przeciwnym przypadku [confirm("tekst_komunikatu")]
  - **prompt()** – okienko dialogowe z polem tekstowym [prompt("tekst_komunikatu","wartość_domyślna")] obydwa parametry opcjonalne – pierwszy oznacza tekst komunikatu drugi domyślną wartość pola tekstowego, zwracana jest zawartość tego pola
  - **createPopup()** – tworzy wyskakujące okienko [window.createPopup()] (tylko IE)
- Fokus:
  - **focus()** – ustawienie fokusa na bieżącym oknie [window.focus()]
  - **blur()** – usunięcie fokusa z bieżącego okna [window.blur()]
- Drukowanie:
  - **print()** – drukowanie zawartości bieżącego okna [window.print()] (zalecane przy stosowaniu ramek)

METODY OBIEKTU WINDOW – C.D.

- Metody związane z czasem:
  - **setTimeout()** – metoda używana by wywołać funkcję lub wyznaczyć wartość wyrażenia po określonej liczbie milisekund [setTimeout(kod, ms, język)]:
    - kod (wymagany) – wskaźnik do funkcji lub kodu do wykonania
    - ms (wymagany) – liczba milisekund do wykonania kodu
    - język (opcjonalny) – język skryptowy kodu: JavaScript|JScript|VBScript
  - **clearTimeout()** – cofa ustawienia metody setTimeout() [clearTimeout(identyfikator)] – identyfikator jest wcześniej zdefiniowaną przez identyfikator=setTimeout() zmienną
  - **setInterval()** – podobnie do setTimeout z tą różnicą, że kod jest wywoływany cyklicznie co ms milisekund [setTimeout(kod,ms,język)] – argumenty jak w setTimeout()
  - **clearInterval()** – cofa ustawienia setInterval() podobnie jak w przypadku clearTimeout()

METODY OBIEKTU WINDOW – C.D.

- Zmiana rozmiaru:
  - **resizeBy(szerokość, wysokość)** – zmiana rozmiaru okna o określoną liczbę pikseli wszerz (szerokość – obowiązkowy argument) i wzdłuż (wysokość – argument opcjonalny) – obydwa argumenty mogą być ujemne
  - **resizeTo(szerokość, wysokość)** – zmiana rozmiaru okna na nowy (szerokość x wysokość) określony w pikselach – pierwszy argument obowiązkowy, drugi opcjonalny
- Przesunięcie:
  - **moveTo(x,y)** – przesunięcie okna do pozycji x,y (współrzędne lewego górnego narożnika)
  - **moveBy(x,y)** – przesunięcie okna o określoną liczbę pikseli x,y względem bieżącego położenia
- Przewijanie:
  - **scrollBy(x,y)** – przewinięcie zawartości okna o określoną liczbę pikseli (paski przewijania nie mogą być wyłączone)
  - **scrollTo(x,y)** – przewinięcie zawartości okna do pozycji x,y

OBIEKT NAVIGATOR

- Informacje ogólne:
  - Obecnie obiekt Java Script a nie HTML DOM
  - Obiekt automatycznie tworzony przez Java Script
  - Zawiera informacje o przeglądarce
- Kolekcje:
  - **plugins[]** – referencje do wszystkich obiektów wbudowanych w dokument
- Metody:
  - **javaEnabled()** – zwraca wartość logiczną informującą o tym, czy przeglądarka obsługuje Javę [navigator.javaEnabled()]
  - **taintEnabled()** – zwraca wartość logiczną informującą o tym, czy przeglądarka ingeruje w dane [navigator.taintEnabled()]

WŁAŚCIWOŚCI OBIEKTU NAVIGATOR

- **appName** – nazwa przeglądarki
- **appCodeName** – nazwa kodowa przeglądarki
- **appVersion** – platforma i wersja przeglądarki
- **appMinorVersion** – podwersja przeglądarki (tylko IE)
- **cookieEnabled** – wartość logiczna określająca, czy włączono oobsługę plików cookie
- **onLine** – wartość logiczna określająca tryb pracy (on-line lub off-line) (tylko IE)
- **browserLanguage** – język przeglądarki (nie Firefox)
- **cpuClass** – klasa procesora po stronie klienta (tylko IE)
- **platform** – system operacyjny
- **systemLanguage** – domyślny język systemu operacyjnego (tylko IE)
- **userLanguage** – ustawienia naturalnego języka systemu operacyjnego (nie Firefox)
- **userAgent** – wartość nagłówka użytkownika agenta wysyłanego przez klienta do serwera

OBIEKT SCREEN

- Informacje ogólne:
  - obecnie obiekt Java Script a nie HTML DOM
  - tworzony automatycznie przez Java Script
  - zawiera informacje o ekranie
- Właściwośći:
  - **height** – wysokość ekranu
  - **width** – szerokość ekranu
  - **availHeight** – wysokość ekranu oprócz paska zadań
  - **availWidth** – szerokość ekranu oprócz pasaka zadań
  - **colorDepth** – głębia kolorów na urządzeniu docelowym lub buforze
  - **bufferDepth** – głębia kolorów bufora (nie ekranu – tylko IE)
  - **deviceXDPI** – pozioma rozdzielczość ekranu w DPI (tylko IE)
  - **deviceYDPI** – pionowa rozdzielczość ekranu w DPI (tylko IE)
  - **logicalXDPI** – normalna rozdzielczość ekranu w DPI (tylko IE)
  - **logicalYDPI** – normalna rozdzielczość ekranu w DPI (tylko IE)
  - **pixelDepth** – jakość kolorów – licznba bitów na kolor (tylko IE)
  - **fontSmoothingEnabled** – wartość logiczna określająca, czy użytkownik zdefiniował w Panelu Sterowania wygładzanie czcionek (tylko IE)
  - **updateInterwal** – właściwość określająca częstotliwość aktualizacji ekranu, można ją też ustawić (tylko IE)

OBIEKT HISTORY

- Informacje ogólne:
  - obecnie obiekt Java Script a nie HTML DOM
  - automatycznie tworzone przez Java Script
  - jest tablicą zawierającą te URL-e, które użytkownik odwiedził w oknie przeglądarki
  - obiekt History jest częścią okna Window i wówczas dostęp do metod i własności odbywa się następująco window.history.właściwość lub window.history.metoda
- Właściwości:
  - **length** – liczba elementów historii (IE i Opera – od 0, Firefox – od 1)
- Metody:
  - **back()** – załadowanie poprzedniej strony z historii
  - **forward()** – załadowanie następnej strony z historii
  - **go(liczba|URL)** – załadowanie określonej strony z historii

OBIEKT LOCATION

- Informacje ogólne:
  - obecnie jest obiektem Java Script a nie HTML DOM
  - automatycznie tworzony przez Java Script
  - zawiera informacje o bieżącym URL
  - jest częścią obiektu Window, więc dostęp do metod i właściwości odbywa się przez window.location.metoda i window.location.właściwość
- Metody:
  - **assign(URL)** – ładuje nową stronę o adresie URL
  - **reload() –** przeładowanie bieżącej strony
  - **replace(URL)** – zastąpienie bieżącej strony stroną o adresie URL

WŁAŚCIWOŚCI OBIEKTU LOCATION

- Odczyt lub ustawienie części URL:
  - href – pełen adres URL (możliwość ustawienia)
  - host – adres serwera i numer portu
  - hostname – nazwa serwera
  - protocol – protokół
  - port – numer portu
  - pathname – ścieżka dostępu
  - hash – część adresu od znaku #, czyli nazwa kotwicy
  - search – fragment ścieżki od znaku ?

OBIEKT DOCUMENT

- Informacje ogólne:
  - reprezentuje cały dokument HTML
  - daje dostęp do wszystkich elementów dokumenu
  - jest częścią obiektu window (dostęp window.document . ...)
- Kolekcje:
  - anchors[] – tablica wszystkich obiektów Anchor dokumentu
  - forms[] – tablica wszystkich obiektów Form dokumentu
  - images[] – tablica wszystkich obiektów Image dokumentu
  - links[] – tablica wszystkich obiektów Area i Link dokumentu

WŁAŚCIWOŚCI OBIEKTU DOCUMENT

- body – bezpośredni dostęp do elementu <body>
- cookie – ustawia lub zwraca wszystkie pliki cookie związane z bieżącym dokumentem
- domain – zwraca nazwę domenową bieżącego dokumentu
- lastModified – zwraca datę i czas ostatniej modyfikacji dokumentu (nie Opera)
- referrer – zwraca URL dokumentu, z którego wywołano dany dokument
- title – zwraca tytuł bieżącego dokumentu
- URL – zwraca URL bieżącego dokumentu

METODY OBIEKTU DOCUMENT

- write(łańcuch1, ...) – zapisuje do dokumentu wyrażenia HTML lub kod Java Script
- writeln(łańcuch1, ...) – jak write ale po każdym łańcuchu przejście do nowej linii
- open()
- close()
- getElementById()
- getElementsByName()
- getElementsByTagName()