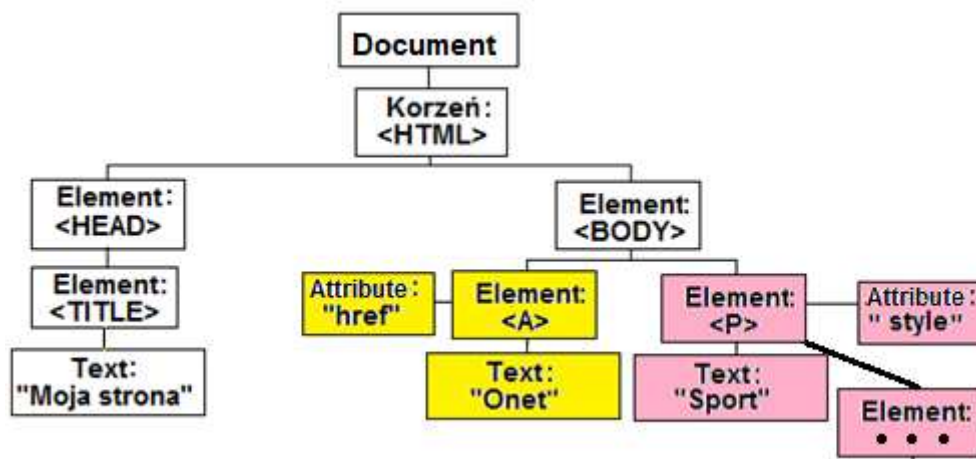


JAVAScript - obiekty HTML

DOM – model dokumentu HTML

Dokument HTML składa się z obiektów (standardowych i utworzonych przez użytkownika).

Głównym obiektem jest **document**, posiada strukturę drzewiastą składowych (obiektów wewnętrznych-znaczników na stronie) – tworzących tzw. **DOM** (*Document Object Model*).



Jak widać elementy mają hierarchię, mogą się zagnieżdżać ("rodzice" i "dzieci").

Obiekt document

Obiekt **document** jest ważnym obiektem *JavaScript*. Zawiera on informacje o bieżącej stronie i dostarcza wielu sposobów wyświetlania strony HTML.

Składnia:

document.właściwość

lub

document.metoda

Wybrane **właściwości**:

Właściwość	Opis
linkColor	określa kolor odsyłaczy
bgColor	określa kolor tła
fgColor	określa kolor tekstu
lastModified	zawiera datę i czas ostatniej modyfikacji dokumentu
location	zawiera bieżący adres URL dokumentu
title	zawiera zawartość znacznika <TITLE>
body	zawiera element(obiekt) body
domain	zawiera nazwę domenową serwera, który załadował
links	zawiera kolekcję wszystkich hipertączy w dokumencie – kolekcja zawiera atrybut length (liczba elementów kolekcji)
title	zawiera tytuł dokumentu
URL	zawiera pełny URL dokumentu

Niektóre metody:

Metoda	Opis
write() writeIn()	wypisanie tekstów, wartości wyrażeń w dokumencie
<i>dostęp do elementów</i>	
getElementById("id")	dostęp do wszystkich elementów o konkretnej wartości atrybutu id , najczęściej jest jeden taki
getElementsByTagName("znacznik")	dostęp do wszystkich elementów o typie znacznika, np. P albo DIV

getElementsByTagName("nazwa")	dostęp do wszystkich elementów o konkretnej wartości atrybutu name
<i>tworzenie nowego elementu</i>	
createTextNode()	tworzenie węzła tekstowego
createElement()	tworzenie elementu

Metodę **write** znamy – metody tworzenia elementów poznamy później.

Przykład dostępu do właściwości, wyświetlania i zmiany wartości:

```
<HTML><HEAD> <TITLE> Przykłady wykorzystania obiektu document</TITLE></HEAD>
<BODY style="font-size:18px">
<P>Przykłady wykorzystania obiektu document</P>
<P> Elementy na stronie</P>
<A href="http://www.youtube.com">Youtube</A>
<A href="http://www.facebook.com">Facebook</A>
<PRE>
<SCRIPT language="JavaScript">
//ustalanie właściwości dokumentu
document.bgColor="yellow";
document.linkColor="red";
document.fgColor="green";
//wyświetlanie właściwości
document.writeln("Data utworzenia:",document.lastModified);
document.writeln("URL:",document.URL);
document.writeln("location:",document.location);
document.writeln("Tytuł:",document.title);
//elementy kolekcji w pętli
for (k=0;k<document.links.length;k++)
    document.writeln("Link ",k,":",document.links[k]);
</SCRIPT>
</PRE></BODY></HTML>
```

Dostęp do elementów strony

Tworzenie referencji do istniejącego elementu (obiekту) – zmiennej reprezentującej element na stronie - najwygodniej wykonać instrukcją *JavaScript*:

var element=document.getElementById ("identyfikator");
pobierz element dokumentu o pewnym id

Ważniejsze właściwości tak pobranego elementu:

element.childNodes	kolekcja węzłów dla elementu - zawiera atrybut length (liczba elementów kolekcji)
element.style	dostęp do ustawiania atrybutu <i>style</i> elementu
element.attributes	kolekcja atrybutów elementu - mają właściwości length – liczba atrybutów name – nazwa [nr].value – wartość atrybutu
element.innerHTML	wartość węzła tekstowego

Ważniejsze metody – które wspomagają dołączanie do strony i usuwanie elementów – poznamy je na następnym ćwiczeniu:

element.appendChild(element_dołączany)	dołączanie elementu do innego elementu (na końcu)
element.cloneNode(true)	powielenie elementu
element.removeChild(element_dziecko);	usunięcie elementu
element.replaceChild(el_nowy,el_stary);	podmiana elementu innym elementem

Atrybut **style** jest również obiektem, poniżej ważniejsze właściwości:

Właściwość	Opis działania
backgroundColor	ustalenie lub pobranie koloru tła elementu
border	ustawienie (np. "1px solid red") lub pobranie cech obramowania
borderStyle	ustawienie (np. "solid") lub pobranie stylu obramowania
borderColor	ustawienie (np. "red") lub pobranie koloru obramowania
borderBottom	dolna krawędź obramowania – j.w.
borderBottomColor	dolna krawędź obramowania – kolor
borderBottomStyle	dolna krawędź obramowania – kolor
borderBottomWidth	dolna krawędź obramowania – kolor
borderLeft borderTop borderRight	podobnie – j.w.
margin	marginies np. "12px 20mm 2 cm 20px" – (górny prawy dolny lewy)
marginBottom marginLeft marginRight marginTop	
width	szerokość elementu (np. "40px" lub "4cm") - ustalenie lub pobranie
height	wysokość elementu – jak wyżej - ustalenie lub pobranie
color	ustawienie (np. "14px solid red") lub pobranie koloru czcionki
fontFamily	ustawienie (np. "Arial") lub pobranie rodziny czcionki
fontSize	ustawienie (np. "15px") lub pobranie rozmiaru czcionki
fontStyle	ustawienie (np. "italic") lub pobranie stylu czcionki
fontWeight	ustawienie (np. "bold") lub pobranie grubości czcionki
textAlign	ustawienie (np. "center") lub pobranie wyrównania tekstu

Zauważmy, że cechy stylu mają podobne nazwy jak w CSS – tylko zamiast myślnika mamy drugie słowo pisane dużą literą, np. w CSS "**font-size**", w *JavaScript* – "**fontSize**". Nadając wartości cechom zawsze je traktujemy jako stałe tekstowe, czyli piszemy np: "12px" lub '12px'.

Pełny opis dla JavaScript np. http://www.w3schools.com/jsref/dom_obj_style.asp

Pełny opis dla CSS np. <http://www.signs.pl/html/o/style.php>

Zapis reprezentujący wartość danej cechy stylu:

zmienna_elementu.**style**.nazwa_właściwości

Na przykład:

```
var d1= document.getElementById("s1"); //pobranie elementu posiadającego id="s1"
document.write(d1.style.fontSize); //wypisanie jego rozmiaru czcionki
```

Na początek szerszy przykład do analizy:

```
<HTML> <HEAD> <TITLE> Przykłady dostępu do elementów </TITLE>
</HEAD>
<BODY style="font-size:18px">
<P> Przykłady dostępu do elementów </P>
<P> Elementy na stronie utworzone w HTML</P>
<A href="http://www.youtube.com">Youtube</A>
<A href="http://www.facebook.com">Facebook</A>
<DIV name="xxx" id="ss" style="font-family:Verdana, Arial, Helvetica, sans-serif;">Napis 1</DIV>
<DIV name="yyy" id="rr" style="color:red">Napis 2</DIV>
<HR />
<PRE>
<SCRIPT language="JavaScript">
```

```

//pobranie referencji do pojedynczego elementu o znanym id
var d1=document.getElementById("ss");
document.writeln("DIV id=ss ma:",d1.attributes.length," atrybutów");
document.writeln("DIV id=ss ma:",d1.style.fontFamily);
//zmiana cechy stylu
d1.style.fontSize="60px";
document.writeln("DIV ss ma teraz 60px czcionkę ");
//inny sposób pobrania pewnej grupy elementów – kolekcja DIV-ów
var d2=document.getElementsByTagName("DIV");
var ile = d2.length; //ile elementów
document.writeln("Drugi DIV ma kolor czcionki:",d2[1].style.color);// liczymy od 0, więc 1 to drugi
document.writeln("Drugi DIV ma tekst:",d2[1].innerHTML);
document.writeln("Dodajemy mu górną zieloną ramkę");
d2[1].style.borderTop="2px solid green";
document.writeln("W dokumencie znajduje się " + ile+ " elementów DIV");
//wyświetlenie wybranych atrybutów – pętla
for (var k=0;k<ile;k++)
    for (var m=0;m<d2[k].attributes.length;m++)
    {
        if (d2[k].attributes[m].name=="id")
        {
            document.write("DIV:",k,"-nazwa atrybutu ",m,":",d2[k].attributes[m].name);
            document.writeln("-wartość atrybutu:",d2[k].attributes[m].value);
        }
    }
//znalezienie kolekcji elementów A
var d2=document.getElementsByTagName("A");
var ile = d2.length;// ile ich jest
document.writeln("W dokumencie znajduje się " + ile+ " elementów A");
//wyświetlenie wybranych atrybutów
for (var k=0;k<ile;k++)//pętla po elementach
    for (var m=0;m<d2[k].attributes.length;m++) //pętla po atrybutach
    {
        if (d2[k].attributes[m].name=="href")
        {
            document.write("A:",k," Numer atrybutu:",m," Nazwa:",d2[k].attributes[m].name);
            document.writeln(" Wartość atrybutu:",d2[k].attributes[m].value);
        }
    }
</SCRIPT>
</PRE></BODY></HTML>

```

Zadanie

- Wykonać przykłady i zrozumieć kolejne operacje skryptu. Wykonać zmiany w skrypcie, np. wymusić zmianę tła elementu DIV, rozmiar czcionki, jej kolor, treść węzła tekstowego itp.
- Sporządzić dokument **HTML** z kilkoma znacznikami, np. z obrazkiem i tabelą o 2 wierszach i 2 kolumnach. Nadać identyfikatory znacznikom oraz zdefiniować im wstępnie w atrybucie **style** kilka cech stylu i ich wartości.
- Wyświetlić wybrane cechy w dokumencie z informacją który element, jaka cecha stylu i jaka jest jej wartość.
- Nadać lub zmienić wartości wybranych cech stylu kodem *JavaScript*.
- Spróbować zamknąć poszczególne składowe działania kodu *JavaScript* w ramach własnych funkcji oraz spróbować wykonać te funkcje przez obsługę umieszczonych na stronie przyci-

sków. Przykładowo: przycisk z napisem "Czerwona ramka" po kliknięciu wykonuje funkcję, w której kod ustala czerwoną ramkę dla jakiegoś znacznika, np. tabeli.

6. Wykonać prosty ruch obiektu HTML, np. obrazka (znacznik IMG) przy pomocy dwóch funkcji i dwóch przycisków, zmieniając funkcjami cechę stylu **marginLeft**.