



WYKŁAD 1

JĘZYK JAVASCRIPT

DEFINICJA JAVASCRIPT

(oficjalna nazwa ECMA-262, ECMAScript 6 - czerwiec 2015r.)

Skryptowy język programowania, którego celem jest dodanie dynamiki do możliwości HTML'a.

Umożliwia:

- manipulację wyglądem i położeniem elementów HTML;
- zmiany zawartości elementów HTML (innerHTML);
- pobieranie danych z formularzy i sprawdzanie ich poprawności;
- asynchroniczne ładowanie danych na stronę (Ajax);

CECHY JAVASCRIPT

- **język skryptowy** - nie musi być kompilowany do kodu maszynowego;
- ze względów bezpieczeństwa nie można zapisywać na dysku komputera, na którym przeglądana jest dana strona;
- wszelkie odwołania do funkcji i obiektów wykonywane są w trakcie wykonywania programu;
- pozwala na odciążenie serwerów i ograniczenie zbędnych danych, wysyłanych przez Internet;
- działa po stronie przeglądarki użytkownika.

ZAMIESZCZANIE KODU W HTML

Kod JavaScript musi być zawarty pomiędzy znacznikami HTML'a `<script>` i `</script>`

Można go umieszczać w dowolnym miejscu dokumentu, ale o ile to tylko możliwe, umieszcza się go albo na początku strony w sekcji HEAD albo na końcu.

- **w części `<head>`** – jeśli chcemy, żeby pliki załadowały się na początku – pliki kluczowe dla działania strony;
- **na końcu przed zamknięciem `</body>`** – jeśli chcemy, żeby pliki nie opóźniały ładowania elementów strony;

PRZYKŁAD

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

 kod skryptu

```
</script>
```

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

```
<noscript>Ta witryna wymaga obsługi skryptów.</noscript>
```

DOBRE PRAKTYKI JAVASCRIPT

Dobłą praktyką jest:

- zamieszczanie kodu JavaScript w zewnętrznych plikach .js;
- zamieszczanie kodu JavaScript na końcu dokumentu;
- ładowanie asynchroniczne plików .js poprzez dodawanie atrybutu `async` do znacznika `<script>` o ile nie zakłóci to działania strony.

```
<script async src="plik.js"> </script>
```

Zewnętrzne pliki są łatwiejsze do zarządzania, do tego poprzez umieszczenie ich w pamięci przeglądarki (cache) strony szybciej się ładują.

WYŚWIETLANIE INFORMACJI

JavaScript nie posiada wbudowanych żadnych funkcji wyświetlających efekty działań.

Można to osiągnąć na cztery sposoby wykorzystując:

- okno z komunikatem `window.alert("cześć")`,
- dokument HTML `document.write("cześć")`,
- element HTML `innerHTML`

```
document.getElementById("demo").innerHTML = "to jest demo";
```

- konsolę przeglądarki `console.log("cześć")`.

WYŚWIETLANIE INFORMACJI

Liczbę trzeba umieścić bezpośrednio w nawiasie okrągłym:

```
document.write(1);
```

Napis musi być ujęty w znaki cudzysłowu lub apostrofu:

```
document.write("abc");
```

```
document.write('abc');
```

W jednej instrukcji można wyświetlić wiele oddzielnych liczb, tekstów i zmiennych – łączenie odbywa się poprzez "+".

```
document.write("abc " + 2.0 + " def " + 3);
```


ZMIENNE

```
var zmienna;
```

```
var liczba =5;
```

Zmienne tworzone są za pomocą słowa kluczowego *var* lub przypisanie wartości. Nazwa zmiennej może być dowolna, choć musi spełniać następujące zasady:

- musi zaczynać się od litery lub znaku podkreślenia,
- może zawierać jedynie litery, cyfry i znak podkreślenia,
- nie może zawierać znaków narodowych (czyli dopuszczalne są jedynie znaki alfabetu łacińskiego),
- w nazwach rozróżniane są małe i duże litery.

JS NOTACJA WIELBŁĄDZIA (CAMEL CASE)

Podczas tworzenia zmiennych programiści wykorzystują różne sposoby na łączenie kilku słów w jedną nazwę zmiennej:

- **Łączniki:** first-name, last-name, master-card, inter-city;
- **Podkreślenia:** first_name, last_name, master_card, inter_city.
- **Camel Case:** FirstName, LastName, MasterCard, InterCity.

W JavaScript najpopularniejsza jest notacja wielbłądzia z pierwszą literą małą:

- **Camel Case:** firstName, lastName, masterCard, interCity.

Łączniki nie są dozwolone w JavaScript – interpretuje je jako operację odejmowania.

TYPY DANYCH

```
var length = 16; // Number
```

```
var lastName = "Johnson"; // String
```

```
var cars = ["Saab", "Volvo", "BMW"]; // Array
```

```
var x = {firstName:"John", lastName:"Doe"}; // Object
```

Typy danych są ważne z punktu widzenia programowania – bez typów danych komputer nie wiedziałby, jakie operacje może wykonać.

```
var x = 16 + 4 + "Volvo"; // 20Volvo
```

```
var x = "Volvo" + 16 + 4; // Volvo164
```

FUNKCJE JAVASCRIPT

Funkcja to blok kodu wywoływany na stronie:

- Przez zdarzenie (np. kliknięcie przycisku);
- Przez inną funkcję;
- Automatycznie (tzw. self invoked)

```
function nazwaFunkcji(arg1, arg2)
{
    var x = arg1*arg2;
}
var x = nazwaFunkcji(2, 5);
```

RETURN W FUNKCJACH

Wewnątrz funkcji argumenty zachowują się jak zmienne lokalne, tzn. ich zmiany nie wpływają na kod poza funkcją.

Return pozwala na przekazanie wyniku działania w funkcji na zewnątrz.

```
function nazwaFunkcji(arg1, arg2)
{
    var x = arg1*arg2;
    return x;
}
var x = nazwaFunkcji(2, 5);
```

OBIEKTY JAVASCRIPT

Obiekt najłatwiej sobie można wyobrazić jako "pojemnik", wewnątrz którego umieszczone są zmienne (właściwości) i funkcje (metody).

Obiektem może być kot. Kot będzie miał między innymi takie właściwości, jak:

cat.age , **cat.breed** , **cat.color**

oraz takie metody jak:

cat.crazyMode() i **cat.purrMode()**

Wszystkie koty mają te same właściwości, ale różne wartości tych właściwości.

Podobnie jest z metodami – wszystkie koty je mają, jednak wywołują je w różnych momentach.

PRZYKŁAD OBIEKTU

```
var cat = {  
  age: 7,  
  breed: "NFO",  
  color: "black",  
  crazyMode: function() { window.alert("Au! It's crazy cat! "); },  
  purrMode: function() { window.alert("Oh, nice kitty!"); }  
};  
cat.crazyMode();
```

ZDARZENIA

Zdarzenia w HTML to rzeczy, które przydarzają się elementom HTML lub dokumentowi.

To na przykład przesunięcie kursora myszy, kliknięcie, załadowanie strony, opuszczenie strony itp.

Każdy element strony ma przypisany zestaw obsługiwanych przez niego zdarzeń, a każdemu z nich można przypisywać własne procedury obsługi, napisane w JavaScript.

ZDARZENIA

Obsługę danego zdarzenia można przypisać na dwa sposoby.

1. Jeśli jest bardzo krótka i może być zapisana w jednej linii kodu, można ją przypisać bezpośrednio do znacznika obsługującego dane zdarzenie – **nie zalecane**:

```
<znacznik zdarzenie="instrukcja;">
```

2. W przypadku dłuższych procedur zdarzeniu należy przypisać jedynie wywołanie funkcji JavaScript:

```
<znacznik zdarzenie="nazwaFunkcji();">
```

```
<p onclick="nazwaFunkcji();">
```

LISTA TYPOWYCH ZDARZEŃ

(więcej: http://www.w3schools.com/jsref/dom_obj_event.asp)

- onfocus - kiedy element otrzymuje fokus;
- onblur - kiedy element traci fokus;
- onchange - kiedy element traci focus i zmienia się wartość;
- onclick - kiedy element został kliknięty;
- onmouseover - kiedy kursor myszy wejdzie w obszar elementu;
- onmouseout - kiedy kursor myszy opuści obszar elementu;
- onselect - podczas zaznaczania fragmentu tekstu;
- onsubmit - podczas wysyłania formularza;
- onload - kiedy przeglądarka zakończy ładowanie strony lub ramki;
- onunload - kiedy przeglądarka usuwa bieżący dokument.

DZIĘKUJĘ ZA UWAGĘ