JavaScript - korzenie

Dowiesz się o historii JavaScript, jego dialektach i wersjach. Poznasz złe i dobre strony języka, a gdy zaskoczy Cię działanie któregoś z jego elementów, będziesz wiedział, gdzie szukać rozwiązania. Z łatwością rozróżnisz JavaSctipt od JScript, usłyszysz też o implementacji JavaScript w nowoczesnych przeglądarkach i na urządzeniach mobilnych. Całość doprawią informacje o kompresji i dołączaniu JS do Twoich stron.

- standard ECMA
- 0.1 + 0.2 !== 0.3 czyli dobre i złe strony ECMA,
- przeszłość i przyszłość języka
- JScript
- JavaScript 1.5
- JavaScript 2.0
- słowo o przeglądarkach
- ładowanie JavaScript - model tradycyjny, asynchronous loading, lazy loading
- kompresja kodu - GZIP, packer, obfuscicating

Zaawansowane podstawy

Podstawy JavaScript w nieco innym ujęciu. Na tacy podane tylko i wyłącznie przydatne, i specyficzne dla JS podstawowe elementy języka. Przeanalizujemy krok po kroku struktury, bez których zaawansowany programista JavaScript nie może się obejść. Porozmawiamy też na temat złych praktyk.

- jak pisaliśmy JavaScript - dobre i złe strony
- zmienne i typy
    - typeof
    - boolean, number, string, object
    - undefined
    - null
    - NaN

    - Infinity
- operatory
- tablice
- funkcje
- pętle i warunki
- wyrażenia regularne
- falsy values
- == versus ===
- słowo o operatorach bitowych

Garbage collector

Obiektowość w JavaScript

JavaScript językiem obiektowym, tylko wspierającym obiektowość, czy jedną wielką haszmapą? Przyjrzymy się modelowi obiektowemu w JavaScript od podszewki, rozwiązując i analizując, jak zwykle, jak najwięcej zadań. Prototype chaining od teraz bez tajemnic!

- obiektowość w JavaScript - wszystko (prawie) jest obiektem!
- object literals
- funkcje jako konstruktory
- .constructor
- .prototype
- badanie obiektów
- toString()
- instanceof
- operator delete
- obiekty wbudowane

Scope

Czyli kiedy i gdzie widoczne są zmienne.

Closures

Kiedy się o nim mówi, wzbudza wśród developerów gęsią skórkę. Niepotrzebnie. Mechanizm closures to łatwa i niezwykle użyteczna metoda obrony przed niedoskonałościami języka, co postaram się udowodnić.

this

"this" – czyli obiekt versus kontekst? Omówię dokładnie wszelkie zastosowania tego wyrażenia, zmierzymy się z jego definicją i przeanalizujemy konteksty, w ramach których definiujemy obiekty i funkcje.

Funkcje jako obiekty

Funkcje w JavaScript zachowują się także jak obiekty – dowiesz się jak w pełni wykorzystać ten fakt.

- obiekt Function
- deklaracja deklaracji nierówna
- funkcje anonimowe
- funkcje wywoływane automatycznie
- new fn
- arguments i obiekty array-like
- fn.length
- funkcje call i apply
- ciekawe zastosowania return

Tablice

Tablice – czyli kiedy zawodzą tradycyjne metody zaczerpnięte z innych języków programowania. Dowiesz się, jak optymalnie manipulować tablicami w JS, poznasz też najprzydatniejsze funkcje, a wszystko po to, by Twój kod był jak najbardziej wydajny. Prędkość ma znaczenie!

- brak typu "array"
- tablice asocjacyjne
- new Array(100)
- usuwanie elementów tablicy
- kopiowanie tablic
- mergowanie tablic
- forEach
- maksymalna i minimalna wartość w tablicy

Klasyczna obiektowość w JavaScript

JavaScript nie jest i nigdy nie był językiem z klasycznym modelem obiektowym. Brak słówek takich jak public i private, tradycyjnego dziedziczenia i wielu innych szczegółów sprawia, że aby uzyskać efekty znane z innych języków, należy wykazać się pomysłowością lub pójść na skróty. Zrobimy to i przy okazji przedyskutujemy, czy na pewno jest to w JS potrzebne!

- własności publiczne
- własności prywatne
- własności uprzywilejowane
- dziedziczenie prototypowe
- dziedziczenie funkcyjne
- abstrakcja klas
- przeładowanie metod
- wybrane wzorce projektowe
- namespacing
- JS a inne języki

Eval

With

- fakty i mity

Callbacki

Timery

- definicja
- while(true)
- setInterval
- setTimeout
- wydajność i pomiary dokładności

bind()

currying

chaining

- piszemy bibliotekę podobną do jQuery

Optymalizacja

Jako język skryptowy, JS jak żaden inny prosi się o rozsądne techniki optymalizacyjne. Dowiesz się, jak uzyskać najszybszy kod, poznając od podstaw

zachowanie kluczowych wyrażeń języka i wiele ciekawostek, które z reguły są pomijane przy budowie aplikacji.

- środowisko ma znaczenie
- pętle for i while
- eval
- memoization
- funkcje a zmienne
- zmienne prywatne a składowe obiektów
- scope chaining
- yielding

Dziwolągi i ciekawostki

- operator "+"
- operator "++"
- undefined jako zmienna
- 0.1 + 0.2 = ?
- co chowa się w arguments?
- NaN i isNaN
- wyrażenie ~~
- new Boolean(false)
- for in
- "in"
- "" + {}
- isFunction()

DOM

- wprowadzenie
- najważniejsze cechy
- zastosowanie

- wycieki pamięci

Debugowanie

- Firebug
- JSLint
-dynaTrace AJAX

Inspiracje

- projekty
- blogerzy
- strony
- narzędzia