Dziedziczenie prototypowe

W kursie poruszony został temat dziedziczenia prototypowego. Jeśli chcesz dowiedzieć się więcej lub rzucić okiem na inne przykłady, zajrzyj tutaj: http://bit.ly/kurs-js-prototype

Tematem nie poruszanym w tym kursie, a związanym również z programowaniem obiektowym w standardzie ECMAScript 5 są tzw. gettery i settery, o których więcej przeczytasz tutaj: http://bit.ly/kurs-js-define-property

Poniżej znajdziesz również referencje do kilku ważnych w kontekście programowania obiektowego funkcji:

call http://bit.ly/kurs-js-call
apply http://bit.ly/kurs-js-apply
bind http://bit.ly/kurs-js-bind

JSON

JSON to lekki, tekstowy format wymiany danych. Jego składnia przypomina bardzo zapis obiektów w języku JavaScript, jednak z pewnymi różnicami. Dokładny opis składni JSON znajdziesz tutaj: http://bit.ly/kurs-js-json

W języku JavaScript dostępny jest globalny obiekt `JSON`, który zawiera dwie przydatne metody. Ich referencje znajdziesz poniżej.

JSON.stringify http://bit.ly/kurs-js-json-stringify
JSON.parse http://bit.ly/kurs-js-json-parse

Przy ich opisach, zwróć uwagę na dodatkowe parametry, które mogą one przyjmować.

AJAX

AJAX to możliwość wysyłania żądań do serwera i otrzymywania odpowiedzi bez przeładowywania witryny. Całość opiera się na obiekcie `XMLHttpRequest`, którego dokładny opis znajdziesz tutaj: http://bit.ly/kurs-js-ajax

Dwie, bardzo ważne w kontekście technologii AJAX specyfikacje to **Same-origin policy** oraz **Cross-Origin Resource Sharing**. Ich dokładne opisy znajdziesz poniżej.

Same-origin policy http://bit.ly/kurs-js-same-origin-policy
CORS http://bit.ly/kurs-js-cors
FormData http://bit.ly/kurs-js-formdata
Lista zakazanych nagłówków przy użyciu z AJAX http://bit.ly/kurs-js-not-allowed-headers

Cookies

Ciasteczka to małe pliki tekstowe, zapisywane po stronie klienta, zazwyczaj na polecenie serwera, który wysyła wraz z odpowiedzią specjalny nagłówek `Set-Cookie`. Po ustawieniu ciasteczka, jest ono przesyłane do serwera z każdym żądaniem.

Ciasteczka mogą być również przypisywane i odczytywane z pomocą języka JavaScript.

Składnię nagłówka `Set-Cookie` ustawiającego ciasteczko znajdziesz tutaj: http://bit.ly/kurs-js-cookies-syntax

Sposoby przypisywania i odczytywania ciasteczek z poziomu kodu JavaScript (`document.cookie`) znajdziesz tutaj: http://bit.ly/kurs-js-cookies

Jeśli planujesz poeksperymentować z ciasteczkami np. z użyciem serwera lokalnego i języka PHP, zerknij na przydatną funkcję tego języka `setcookie`, dzięki której serwer przyśle do przeglądarki odpowiedni nagłówek: http://bit.ly/kurs-js-php-setcookie

Wyrażenia regularne

Wyrażenia regularne to bardzo potężne narzędzie w programowaniu. Dzięki nim możemy dopasować tekst do zdefiniowanego wzorca i np. wyszukać wszystkich wystąpień adresu e-mail w tekście czy zamienić notację Markdown na kod HTML. Wyrażeń regularnych nie brakuje oczywiście w języku JavaScript, a więcej na ich temat znajdziesz tutaj: http://bit.ly/kurs-js-regex

Bardzo dobry cheatsheet wyrażeń regularnych znajdziesz pod tym adresem: http://bit.ly/kurs-js-regex-cheatsheet

A najlepsze z dostępnych narzędzi do nauki wyrażeń regularnych znajdziesz tutaj (jest również dołączone w plikach źródłowych do tego kursu): http://regexr.com/

Dobre praktyki pracy z kodem JavaScript

Nie zapominaj nigdy o dobrych praktykach pracy z kodem. Jedną z nich jest korzystanie z tzw. “strict mode” języka JavaScript. Dzięki uruchomieniu skryptu lub funkcji w takim trybie, unikniesz wielu przypadkowych błędów. O “strict mode” możesz poczytać tutaj: http://bit.ly/kurs-js-strict-mode

Staraj się również regularnie sprawdzać poprawność pisanego kodu. Możesz w tym celu skorzystać z narzędzi online lub zainstalować ich wersje jako dodatki do popularnych edytorów tekstu. Szczególnie polecane lintery to JSLint oraz JSHint.

Pluginy powyższych narzędzi dla edytora Sublime Text znajdziesz tutaj: JSLint oraz JSHint. Bardzo dokładne omówienie edytora Sublime Text znajdziesz w jednym z warsztatów na eduweb.pl Kodowanie w Sublime Text 3.

A kiedy kod jest już gotowy do produkcji, warto go odpowiednio zoptymalizować, m. in. poprzez minifikację. Narzędzie online, które Ci w tym pomoże to np. JavaScript Minifier. Takie czynności warto jednak wykonywać w sposób zautomatyzowany, np. z użyciem automatora zadań Grunt.js. Jeśli jesteś ciekaw jak on działa, koniecznie rzuć okiem na nasz warsztat na ten temat: Wprowadzenie do Grunt JS.