

Dokumentácia k projektu

Analýza zadania

Cieľom tohto projektu bolo vytvoriť skript, ktorý bude schopný zvýrazniť podľa zadaného formátovacieho súboru, syntax vstupného textu. Formátovací súbor je špecificky syntaxne zapísaný súhrn pravidiel, podľa ktorých používateľ očakáva výsledok vo forme pridaných HTML elementov.

Postup riešenia

Po spustení skriptu sa ako prvé spracujú používateľské argumenty. Pomocou získaných dát od používateľa sa nastaví dôležité premenné, ktoré z veľkej miery ovplyvňujú ďalšie chovanie skriptu.

V prípade korektného zadania argumentov podľa zadania program pokračuje ďalej kontrolou a spracovaním formátovacieho súboru pokiaľ bol zadaný. Súčasťou spracovania formátovacieho súboru je aj preklad jazyka regulárnych výrazov opisujúce zadanie, do jazyka regulárnych výrazov jazyka PHP5. Výstupom tejto funkcie je pole spracovaných informácií obsahujúcich preložený regulárny výraz a zadané formátovacie kľúčové slová.

Za predpokladu správnosti zápisu formátovacieho súboru a nenájdení žiadnej chyby má skript pripravené podmienky na začatie čítania vstupného súboru a aplikovania HTML formátovacích elementov.

Spracovanie argumentov

O správne vyhodnotenie všetkých argumentov sa stará funkcia *arguments*. Jej funkčnosť je naprosto priehľadná. Vyhľadáva očakávaný argument v zozname zadaných argumentov. Ak ho nájde, tak vráti spracovaný argument, ak nie vracia false.

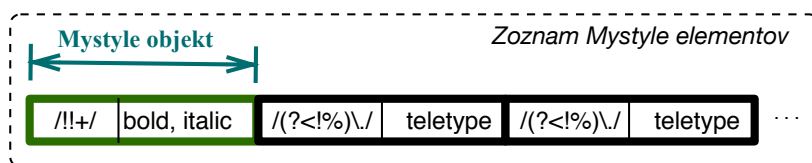
Preklad regulárnych výrazov

Preklad výrazu špecifikujúci zadanie bola jedná z najväčších problémov. Po dlhom zvažovaní rôznych metód som sa rozhodol pre vytvorenie slovníka. Vytvoril som regulárne výrazy, ktoré vyhľadávajú práve sporné zápisy daných operácií a nahrádzajú ich za správne. Tieto preklady sa vykonávajú vo svojom špecifickom poradí. Dosiahnutý výsledok je preložený regulárny výraz, ktorý korektne dokáže spracovať funkcia na vyhľadávanie podľa regulárnych výrazov v PHP.

Spracovanie formátovacieho súboru

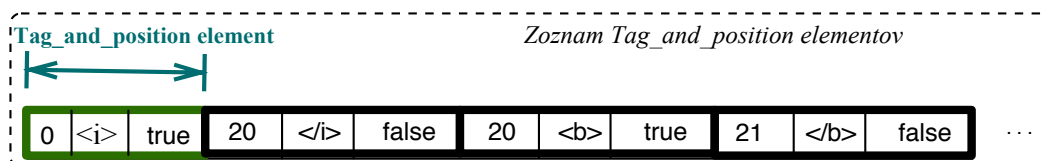
Spracovanie je prirodzene podmienené zadaním daného formátovacieho súboru. V prípade jeho nedostupnosti špecifikuje chovanie zadanie.

Za vykonanie spracovania zodpovedá funkcia *process_format*. Formátovací súbor sa načítava po riadkoch. Každý riadok je rozdelený podľa prvého tabulátora na regulárny výraz a formátovanie. To je následne vkladané do objektu *Mystyle*. Výsledkom funkcie je pole objektov *Mystyle* ktorý obsahuje regulárny výraz a požadované formátovania.



Vyhľadanie pozície HTML elementov

Implementovaná metóda iteruje nad polom objektov *Mystyle* a vyhľadáva všetky zhody regulárneho výrazu vo vstupnom súbore za pomoci funkcie *preg_match_all*, ktorú obsahuje PHP. Prevedie sa zoznam pseudoelementov na reálne HTML elementy a pre každú zhodu sa vytvorí nový objekt *Tag_and_position*. V ňom je priamo uložená pozícia, kde treba daný element vložiť, presný text HTML elementov a či sa jedná o otváracie elementy alebo zatváracie. Výsledkom operácie je zoznam objektov.

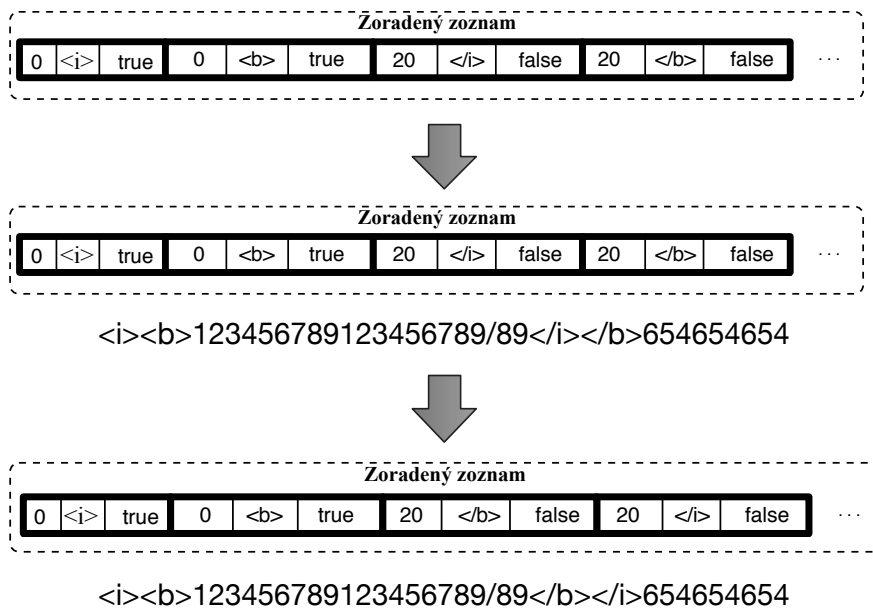


Metóda vkladania elementov

V prípade, že program už vyhodnotil každý regulárny výraz a našiel pozíciu, kde je nutné element umiestniť, ďalej program pracuje už len s týmto zoznamom informácií.

Aby program zabezpečil postupné vkladanie elementov bez straty, či prepočítavania ďalších pozícií, po každom vložení celý tento zoznam program zoradí od najväčšej po najmenšiu pozíciu. Avšak metóda, ktorá by takýto zoznam zoradila, musí pracovať stabilne. PHP neobsahuje žiadnu metódu, ktorá by spĺňovala túto požiadavku preto som implementoval pre tento účel BubbleSort. Poslednou operáciou je invertovanie poradia všetkých zatváracích elementov práve na jednej pozícii ak sa ich tam nachádza viac. Toto je nutné práve kvôli tomu, že zatvárací element sa do zoznamu vkladá v rovnakom čase ako otvárací pre daný regulárny výraz.

Po vysporiadaní sa s týmto problémom je pripravená ideálna pôda pre vkladanie. Program prejde každú položku a od konca vloží všetky elementy zaradom ako sú v zozname pripravené.



Rozšírenie NQS

Rozšírenie som zahrnul do funkcie, ktorá prekladá regulárne výrazy. Pomocou regulárneho výrazu najprv nájdem všetky opakovania dvoch a viacerých operátorov “+” a nahradím ich za jedno a podobne s operátor “*” a následne všetky zhody striedajúcich sa “+*+*+” nahradím za “*”.