

Marco - Feature #1106

Math Escaping Script

30.05.2024 11:11 - Tobias Mai

Status:	Erledigt	Beginn:	30.05.2024
Priorität:	Normal	Abgabedatum:	
Zugewiesen an:	Birte Reich	% erledigt:	0%
Kategorie:		Geschätzter Aufwand:	0.00 Stunde
Zielversion:		Aufgewendete Zeit:	0.00 Stunde

Beschreibung

@Marco, das ist kein dringliches Ticket für die nächsten Wochen (und deswegen erstmal mir zugewiesen). Ich kopiere mein Test-Skript hier mit ins Ticket und dir in Sciebo, das schon einen guten Teil des Problems löst. Bevor wir das Angehen sollten wir noch mal über das Ticket reden, denn es ist schon recht komplex.

Der Konverter soll erweitert werden und zwar wie folgt:

1. Ein Skript erstellen, dass in einer Tex-Datei alle Vorkommen von Formeln in HTML-Umgebungen setzt und Backslashes automatisiert escaped * Mir bekannte Umgebungen sind jegliche Mathe-Umgebungen aus Tex (\$, \$\$, \langle, \lfloor, ...) dazu mindestens noch eqnarrays * Es muss sorgfältig geprüft werdne, ob das alle Umgebungen sind.
2. Der Konverter nutzt das Skript beim Kompilieren, indem alle Tex-Dateien vorbehandelt werden und dann erst der ttm drüber läuft * Mir erscheint es sinnvoll, Arbeitskopien der Tex-Dateien zu erstellen und auch das Inhaltsverzeichnis für die digitale Version direkt auf die Arbeitsversion zu setzen * Eine gute Namensextension wird "_Formeloptimiert" sein
3. Der Konverter sollte sich möglichst nicht verschlucken, nachdem er mal nicht zu ende durchgelaufen ist. Abbruch durch den User oder auch Tex-Fehler.
4. Die neue Art Formeln zu setzen, "vergisst" alle Befehle, die vorher definiert wurden und nicht originäres Latex sind. Das geht bei \R los und weiter mit \eur usw. * Wir müssen zuerst diese Befehle ALLE sammeln und danach eine Lösung finden, sie anders zu definieren ** Lösung 1: Zu beginn jedes Bereichs eine Liste mit den Definitionen als Tex-Befehl einfügen, damit MathJax die Definitionen kennt ** Lösung 2: Man kann auch auf JavaScript-Ebene Befehle für MathJax definieren, das ist möglicherweise der elegantere Weg
5. Testung: Das ganze muss sorgfältig getestet werden, dh. Sichtprüfung des gesamten Materials. Überall. KV, LV, in Lösungen, alle Bereiche in jedem Kapitel.
6. Zeitplan: Idealerweise ist die Umstellung im Herbst fertig, sodass der der Moodle-Workaround dafür abgestellt werden kann.(Wenn man in Moodle den Filter Mathjaxloader auf eine Version ab ca. 4.1.09 zurückdreht, funktioniert es in Moodle. Ab 4.1.10 wurde eine Funktion ergänzt, die uns die MathML-Klammern kaputt macht.)

Historie

- #1 - 23.07.2024 13:01 - Tobias Mai
- Zugewiesen an wurde von Tobias Mai zu Birte Reich geändert
- #2 - 21.01.2025 11:18 - Birte Reich
- Status wurde von Neu zu Erledigt geändert

Dateien

test.py	4,23 KB	30.05.2024	Tobias Mai
---------	---------	------------	------------