

Weiterentwicklung Dokuwiki - Geführte Übungen

Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

Weiterentwicklung Dokuwiki - Geführte Übungen 2

Kurzbeschreibung 2

Mockup 2

 Beispielseite 2

Widerstand eines Drahts 2

Features 3

Syntax-Beispiel 4

Das ist eine Überschrift. `$$` Sie enthält aber einen

`http://wiki.mexle.org` `{\text{Link zum MEXLE}}` } `$$` Etwas umständlich,

aber würde als Quickfix auch für andere syntaxreduzierte Bereiche gehen. 6

 Selbiges klappt auch in Strings, wie dem 'title' eines Callouts. `$$` Auch hier wären Links,

 z.B. auf eine `http://wiki.mexle.org` `{\text{MEXLE-Seite}}` } `$$` möglich. 6

Weiterentwicklung Dokuwiki - Geführte Übungen

Kurzbeschreibung

In diesem Dokuwiki sind verschiedene Lern/Lehrmaterialien für Studierende hinterlegt. Es soll für verschiedene Fächer geführte Übungen erstellt werden. Darunter sind Übungsaufgaben zu verstehen, die Studierende durchrechnen sollen, um im Anschluss - je nach Antwort - auf verschiedene weitere Aufgaben oder Dokumente geführt zu werden.

Für die zusätzlich gewünschten Features ist die Entwicklung eines Plugins notwendig.

Mockup

Beispielseite

Widerstand eines Drahts

Berechnen Sie den ohmschen Widerstand R eines Kupfer-Drahts ($\rho = 1.59 \frac{\Omega \cdot \text{mm}^2}{\text{m}}$) mit dem Durchmesser $d = 7.25 \text{ mm}$ und der Länge $l = 10 \text{ m}$!

$0 \text{ m}\Omega$

[Ergebnis als mathem. Formel anzeigen](#) Ergebnis prüfen

Es wurde noch kein Wert eingegeben. Bitte tragen Sie den korrekten Zahlenwert ein.

Features

Studierendensicht:

1. Auf einer Wikiseite ist für den Studierenden eine Aufgabenstellung beschrieben. Diese enthält verschiedene, variable Werte, sowie Eingabefelder für ein Ergebnis und Buttons.
2. Nachdem der Studierende die Aufgabe gelesen hat, soll er das Ergebnis berechnen und in die Eingabefelder eintragen. Dabei können Text/Zahlenfelder oder Dropdownfelder zur Verfügung stehen.
3. Der Student kann im Anschluss über Knopfdruck seine Eingabe in MathJax-Darstellung rendern lassen.
4. Wenn der Student meint seine Antworten wären passend, kann er mit einem weiteren

Knopfdruck seine Eingabe überprüfen lassen.

5. Nach der Überprüfung kann entweder auf der gleichen Seite eine Rückmeldung angezeigt werden (z.B. ob das Ergebnis richtig oder falsch ist, bzw. die Notwendige Formel o.ä.) oder direkt eine neue Seite geladen werden.

Bearbeitersicht:

1. Eine Basis dafür kann das `Plugin:const` sein. Für die Buttons kann auf `plugin:bootswrapper` zurückgegriffen werden.
2. Dem Bearbeiter soll es möglich sein, Werte für Aufgaben vorzugeben.
 1. Es soll eine Liste als Wertebereich vorgegeben werden können, aus der zufällig ein Wert ausgewählt wird.
Im Mockup: Länge $\$l\$$ aus der Liste (10, 12, 15, 20)
 2. Es soll eine n-dimensionale Liste als Wertebereich vorgegeben werden können, aus der zufällig eine n-1 Gruppe ausgewählt wird.
Im Mockup: Der Materialname (Kupfer) und der Wert ρ (1.59) gehören zusammen. Diese wurden aus der Liste (("Silber", 1.59), ("Kupfer", 1.79), ("Aluminium", 2.78)) ausgewählt. In den Textangaben in der Liste sollen alle Zeichen nutzbar sein, also auch Komma und Anführungszeichen. Das kann z.B. durch Verwendung von Backslash als Escapezeichen geschehen.
 3. Es soll eine Zufallszahl mit Grenzen und Nachkommastellen vorgegeben werden können:
\\Im Mockup: Der Durchmesser d soll sich aus einer Zufallszahl zwischen 5 und 10 mit zwei Nachkommastellen ergeben, mit einer Schrittweite von 0.05
3. Dem Bearbeiter soll es möglich sein, Eingabefelder für Aufgaben vorzugeben. Die Auswahl/Eingabe soll ausgewertet werden.
 1. Es soll Text/Zahlenfelder oder Dropdownfelder geben.
 2. Eingaben über Text/Zahlenfelder sollen als Formeln interpretiert werden können (also z.B. $4+5$ oder e^4), jedoch ohne Zugriff auf Variablen (z.B. die Länge l).
 3. Die Interpretation als Formel bei Eingaben über Text/Zahlenfelder soll auch unterbunden werden können.
 4. Für die Text/Zahlenfelder kann der Bearbeiter eine maximale Anzahl an Nachkommastellen oder eine maximale Anzahl an Zeichen angeben.
 5. Bei Dropdownfeldern soll eine zufällige Reihenfolge einer Liste möglich sein.
 6. Optional soll eine Teilmenge aus einer Liste auch darstellbar sein. In diesem Fall ist zu prüfen, wie eine korrekte Antwort durch den Bearbeiter markiert und durch das Plugin immer eingefügt werden kann.
1. Für die finale Auswertung des Ergebnisses muss der Bearbeiter Aktionen angeben.
 1. Es soll eine Möglichkeit bestehen die festzustellen, wie weit das eingegebene Ergebnis vom korrekten Ergebnis abweicht.
 2. Es ist vorzusehen, dass je nach Ergebnisart (leer, Falsch, richtig) sich weitere Unterbereiche der Wikiseite öffnen sollen. Die kann z.B. durch ein Setzen von `collapse ids` für die `collapse` Komponenten des bootswrapper möglich sein.

Syntax-Beispiel

Im Folgenden ist ein Beispiel für eine Syntax dargestellt. Von dieser kann abgewichen werden.

```
<const>
rhoEinheit          =  $\frac{\omega \cdot \text{mm}^2}{m}$ 
```

```

l           = (10, 12, 15, 20)
d           = %RANDOM%[min=1,max=2, digits=2] * 5
(Material, rho) = (("Silber", 1.59), ("Kupfer", 1.79), ("Aluminium",
2.78))
erg         = rho * l / (d^2 *pi/4)

(praefix, prxMult) = ((m , 0.001), ( , 1), (k, 1000), (M,
1000000))[randomOrder=no]
naechsteSeite()    = CollapseFuerNochKeineLoesungEingegeben,
CollapseFuerRichtigeLoesung, CollapseFuerFalscheLoesung
ergAbwBool        = eingebeErg * ((abs((eingebeErg * prxMult - Erg) / Erg)
< 0.01) + 1) // workaround, um 0 für keine Eingabe ,
1 für ein korrektes Ergebnis,

eingebeErg       = 0[digits = 3, evaluation=no]
</const>

```

===== Widerstand eines Drahts =====

Berechnen Sie den ohmschen Widerstand R eines `Material`-Drahts ($\rho = \rho$ `rhoEinheit`) mit dem Durchmesser $d = d$ mm und der Länge $l = l$ m!

```

<result>
%%eingebeErg%% %%prxMult%% \Omega$
</result>

```

```

<input-form> erg </input-form> <input-form> prx_mult </input-form> $\Omega$

```

```

<btn type="default">[[|Ergebnis als mathem. Formel anzeigen]]</btn>
<btn type="default">[[|%NaechsteSeite(ergAbwBool)%|Ergebnis prüfen]]</btn>

```

```

<collapse id="CollapseFuerNochKeineLoesungEingegeben"
collapsed="true"><well>
Es wurde noch kein Wert eingegeben. Bitte tragen Sie den korrekten
Zahlenwert ein.
</well></collapse>

```

```

<collapse id="CollapseFuerRichtigeLoesung" collapsed="true"><well>
Ja, %%erg%% ist richtig! Toll gemacht.
</well></collapse>

```

```

<collapse id="CollapseFuerFalscheLoesung" collapsed="true"><well>
%%eingebeErg%% ist leider nicht richtig. Die korrekte Lösung lautet %%erg%%.
Es ist die Formel ... zu nutzen.
</well></collapse>

```

`\require{\hyperref}`

Das ist eine Überschrift. `\\` `\\` Sie enthält aber einen `{\href{http://wiki.mexle.org}}{\text{Link zum MEXLE}}` } `\\` `\\` Etwas umständlich, aber würde als Quickfix auch für andere syntaxreduzierte Bereiche gehen.

Selbiges klappt auch in Strings, wie dem 'title' eines Callouts. `\\` `\\` Auch hier wären Links, z.B. auf eine `{\href{http://wiki.mexle.org}}{\text{MEXLE-Seite}}` } `\\` möglich.

From:

<https://mexle.te.hs-heilbronn.de/> - MEXLE Wiki

Permanent link:

https://mexle.te.hs-heilbronn.de/studentische_arbeiten/weiterentwicklung_dokuwiki_gefuehrte_uebungen?rev=1657744325

Last update: 2022/07/13 22:32

