

Integration von Python in X_YL^AT_EX am Beispiel von Katalogeinträgen

Lukas C. Bossert, Uwe Ziegenhagen

Viele Dissertationen in der Archäologie enthalten am Ende der Arbeit einen Katalog, in dem die untersuchten Daten in einem bestimmten System aufgeschlüsselt präsentiert werden. Ein solcher Katalog kann aus Bildern, Bohrproben, Architekturelementen etc. bestehen.

Eine händische Erstellung der einzelnen Einträge bspw. über `\section` oder `\subsection` und anschließend in einer `items`-Umgebung ist nicht effizient, fehleranfällig und nur bei wenigen Katalogeinträgen einsetzbar. Es muss zudem berücksichtigt werden, dass die vorgegebenen Kategorien nicht für jeden Katalogeintrag passend sind, sodass Kategorien leer bleiben und dann im Katalogeintrag nicht auftauchen sollen. Dabei soll der Code des Katalogeintrags möglichst viel redundante Tipp-Arbeit abnehmen, wie sie bspw. bei Maßeinheiten vorkommt. Darüber hinaus sollen alle Einträge immer gleich formatiert sein und ihr Aussehen global verändert werden können.

Nachdem eine Lösung gefunden wurde, die allen bisher genannten Anforderungen entspricht (siehe unten), sollten in einer Kategorie alle Seitenzahlen enthalten sein, auf denen im Haupttext auf den Katalogeintrag verwiesen wird. Ein Lösungsansatz sah die Nutzung von `glossaries` vor: Dafür musste allerdings für jeden Katalogeintrag ein eigenes Glossar angelegt werden, was nicht nur viel händische Arbeit bedeutet, sondern auch die Zählerkapazitäten von T_EX überforderten.

Es musste also eine andere Lösung her und ich (Lukas) erinnerte mich an einen Beitrag, den Uwe Ziegenhagen in dtk 1/2015 vorgestellt hatte: Damals ging es darum, bei einem Werkkatalog die Erwähnung des Stückes im Haupttext anzugeben. Prima, genau das, was ich gesucht habe und ich fand den damals verwendeten Code auf seiner Webseite.¹ Allerdings sah ich nun, dass die Aufgabe damals darin bestand, nur *eine* Erwähnung im Haupttext anzugeben, was mit `label` und `ref` bewerkstelligt werden konnte. Ich brauche jedoch alle Erwähnungen im Haupttext. Meine Emailanfrage bei Uwe nach einer Möglichkeit mehr als nur eine Erwähnung anzeigen zu lassen, beantwortete er mit dem Vorschlag, mittels Python das Problem zu lösen.

Um gemeinsam an dem hybriden Konstrukt von X_YL^AT_EX und Python wurde ein Repository auf der Plattform `github`² erstellt, die unabhängig der Programmierspra-

¹ <http://uweziegenhagen.de/?p=3020>

² <https://github.com/LukasCBossert/DTK-TeX-Python>

che einen exzellenten Austausch und eine detaillierte Versionskontrolle bereithält, sodass sich kollaboriertes Arbeiten sehr gut konzipieren lässt.

Katalogeintrag

Als Aufgabe soll ein Katalog zu den Häusern in Pompeji angelegt werden, der Auskunft über den Namen des Hauses, dessen Verortung und Grundstücksgröße geben soll. Zudem soll eine kurze Beschreibung enthalten sein und die Angabe über die Innenausstattung, die wiederum aufgeschlüsselt werden kann auf die Untergruppen Mosaik, Wandgemälde und Skulptur. Zum Schluss soll im Katalogeintrag angezeigt werden, auf welchen Seiten des Fließtextes das Haus genannt wird.

Die gefundene Lösung für die Umsetzung der Anforderungen funktioniert mit dem Paket `keyval`.³ Dafür werden in der Präambel verschiedene Schlüssel (keys) definiert. In der Grundversion sieht die Definition eines Eintrages wie folgt aus:⁴

```
\define@key{family}{key}{#1}
```

Für das konkrete Beispiel wird die `key`-Familie (`family`) mit `catalogue` angegeben, der Schlüssel (`key`), was einer Kategorie im Katalog entspricht, als `house` bezeichnet und als Resultat soll der Wert im Makro `\KVhouse` gespeichert werden:

```
\define@key{catalogue}{house}{\def\KVhouse{#1}}
```

Dem Beispiel entsprechend können alle Kategorien als Schlüssel angelegt werden:

```
1 \makeatletter
2 \define@key{catalogue}{house}{\def\KVhouse{#1}}
3 \define@key{catalogue}{label}{\def\KVlabel{#1}}
4 \define@key{catalogue}{description}{\def\KVdescription{#1}}
5 \define@key{catalogue}{location}{\def\KVlocation{#1}}
6 \define@key{catalogue}{size}{\def\KVsize{#1}}
7 \define@key{catalogue}{interior}{\def\KVinterior{#1}}
8 \define@key{catalogue}{interiorM}{\def\KVinteriorM{#1}}
9 \define@key{catalogue}{interiorW}{\def\KVinteriorW{#1}}
10 \define@key{catalogue}{interiorS}{\def\KVinteriorS{#1}}
11 \define@key{catalogue}{literature}{\def\KVLiterature{#1}}
12 \makeatother
```

Das Aussehen eines Katalogeintrages wird separat mit dem Makro `\newcommand\catalogueentry[1]...` definiert: Darin wird zunächst festgelegt, dass eine neue Gruppe beginnt (`\begingroup`), sodass es zu keinen Problemen mit den jeweils

³ Basierend auf der Idee vorgestellt auf: <http://tex.stackexchange.com/a/254336/98739>

⁴ Vgl. <http://www.tug.org/tugboat/tb30-1/tb94wright-keyval.pdf>

definierten Schlüsseln kommt, da diese für jeden Katalogeintrag neu definiert werden. Dann sollen die Einträge im Flattersatz gesetzt werden (`\RaggedRight`) und schließlich die Angabe, welche Schlüssel Familie (hier `catalogue`) auszulesen ist.

```
1 \newcommand\catalogueentry[1]{%
2 \begin{group}
3 \RaggedRight
4 \setkeys{catalogue}{#1}
5 ...
```

Es folgt in der Definition die Verarbeitung der einzelnen Schlüssel: Da nur die Ausgabe einer Kategorie erfolgen soll, wenn diese auch mit Informationen versehen ist, wird dies über die Abfrage `\ifdef` erledigt. Der Inhalt der Kategorie `house` wird als `\section`-Titel verwendet und, wenn vorhanden, mit einem `\label` versehen. Die weiteren Kategorien sollen in einer `labeling`-Umgebung aufgelistet werden. Die Definition wird mit `\endgroup` geschlossen.

```
1 ...
2 \ifdef{\KVhouse}{\section{\KVhouse
3   \ifdef{\KVlabel}{\label{\KVlabel}}{}}{}}
4 \begin{labeling}{Beschreibung}
5   \ifdef{\KVdescription}{\item[Beschreibung] \KVdescription}{}
6   \ifdef{\KVlocation}{\item[Verortung] \KVlocation}{}
7   \ifdef{\KVinterior}{\item[Ausstattung] \KVinterior
8     \begin{labeling}{Wandgemalde}
9       \ifdef{\KVinteriorM}{\item[Mosaik] \KVinteriorM}{}
10      \ifdef{\KVinteriorW}{\item[Wandgemalde] \KVinteriorW}{}
11      \ifdef{\KVinteriorS}{\item[Statuen] \KVinteriorS}{}
12    \end{labeling}
13    }{}
14   \ifdef{\KVsize}{\item[Größe] \SI{\KVsize}{\meter\squared}}{}
15   \ifdef{\KVliterature}{\item[Erwähnungen] \KVliterature}{}
16 \end{labeling}
17 \endgroup
18 }
```

Mit dieser Einstellung lassen sich die Katalogeinträge schon sehr gut darstellen. Im Fließtext des Hauptdokuments kann man an gewünschter Stelle mit `\catalogueentry` einen Katalogeintrag eintragen. So wird aus den folgenden Angaben

```
\catalogueentry{%
  house={Haus des M. Fabius Rufus},
  label={haus:M-Fabius-Rufus},
  size={172},
  description={Haus besteht aus mehreren Einzelgebuden.},
```

```
location={Regio VII, Insula 16, Eingang 17--22.},
interior={Reicher Fundkomplex.},
interiorM={S/W-Mosaik},
interiorW={Dionysius mit einer Mnade, Narzissus und ein Cupido, Hercules und
↪Deinira, etc.},
interiorS={Bronzene Statue eines Epheben},
}
```

ein Katalogeintrag, der so aussieht:

Haus des M. Fabius Rufus

Beschreibung Haus besteht aus mehreren Einzelgebäuden.

Verortung Regio VII, Insula 16, Eingang 17–22.

Ausstattung Reicher Fundkomplex.

Mosaike S/W-Mosaik

Wandgemälde Dionysius mit einer Mänade, Narzissus und ein Cupido, Hercules und Deinira, etc.

Statuen Bronzene Statue eines Epheben

Größe 172 m²

Wie man bei diesem Beispiel sieht, ist die Reihenfolge, in der die Kategorien angegeben werden, irrelevant, da die Definition in der Präambel entscheidend ist. Der Wert bei `size` wird intern sogleich an das vordefinierte `\SI`-Makro mit entsprechender Einheit (m²) übergeben. Ähnlich kann auch mit anderen Angaben verfahren werden (bspw. bei Abbildungen können die `\label` an ein vordefinierteres Macro `\cref` übergeben werden).

Bei dem oben beschriebenen Katalogeintrag zum ›Haus des M. Fabius Rufus‹ sind alle Kategorien ausgefüllt. Wenn eine Kategorie nicht ausgefüllt ist, wird sie nicht ausgegeben. Allerdings besteht die Kategorie `interior` [Ausstattung] zusätzlich aus drei Unterkategorien (`interiorM` [Mosaike], `interiorW` [Wandgemälde], `interiorS` [Statuen]). Diese sollen auch dann angezeigt werden, wenn `interior` selbst nicht definiert ist.

Ein solcher Fall tritt beim ›Haus des Wilden Ebers‹ auf. Im Fließtext ist der Katalogeintrag wie folgt ausgefüllt:

```
\catalogueentry{%
house={Haus des Wilden Ebers},
label={haus:Wilden-Ebers},
size={54},
```

```

description={Renovierung nach Erdbeben 62\,n.\,Chr.},
location={Regio VII, Insula 4, Eingang 48, 43},
interiorM={S/W-Mosaik},
interiorW={Venus, Leda und der Schwan, Ariadne und Theseus},
}

```

So wird daraus:

Haus des Wilden Ebers

Beschreibung Renovierung nach Erdbeben 62 n. Chr.

Verortung Regio VII, Insula 4, Eingang 48, 43

Ausstattung Mosaik S/W-Mosaik

Wandgemälde Venus, Leda und der Schwan, Ariadne und Theseus

Größe 54 m²

Um zu diesem Ergebnis zu kommen, ist es notwendig, mit Booleschen Operatoren zu arbeiten. Dafür müssen bei den key-Definitionen drei Operatoren eingeführt werden:

```

1 \newbool{@KVinteriorM}%Mosaik
2 \newbool{@KVinteriorW}%Wandgemlde
3 \newbool{@KVinteriorS}%Statue

```

Diese Booleschen Operatoren werden bei der Definition der einzelnen Einträge eingebaut und als true gesetzt, wenn dieser Eintrag mit Informationen versehen wird. Konkret sieht die Modifikation so aus:

```

1 \define@key{catalogue}{interiorM}{\def\KVinteriorM{#1}\booltrue{@KVinteriorM}}
2 \define@key{catalogue}{interiorW}{\def\KVinteriorW{#1}\booltrue{@KVinteriorW}}
3 \define@key{catalogue}{interiorS}{\def\KVinteriorS{#1}\booltrue{@KVinteriorS}}

```

Zudem müssen die Booleschen Operatoren bei der Ausgabe der Katalogeinträge eingesetzt werden, sodass geprüft wird (`\ifbool{expr}`), ob einer oder mehrere der Operatoren auf true gesetzt ist.

```

1 \ifdef{\KVinterior}{%
2   \item[Ausstattung] \KVinterior
3   \ifbool{bool{@KVinteriorM}}
4     or bool{@KVinteriorW}
5     or bool{@KVinteriorS}}{%
6   \begin{labeling}{Wandgemlde}
7     \ifdef{\KVinteriorM}{\item[Mosaik] \KVinteriorM}{%

```

```

8      \ifdef{\KVinteriorW}{\item[Wandgemlde] \KVinteriorW}{}
9      \ifdef{\KVinteriorS}{\item[Statuen] \KVinteriorS}{}
10     \end{labeling}
11   }{}
12   {\ifboolexpr{bool{@KVinteriorM}
13     or bool{@KVinteriorW}
14     or bool{@KVinteriorS}}{%
15     \item[Ausstattung]%
16   \begin{labeling}{Wandgemlde}
17     \ifdef{\KVinteriorM}{\item[Mosaike] \KVinteriorM}{}
18     \ifdef{\KVinteriorW}{\item[Wandgemlde] \KVinteriorW}{}
19     \ifdef{\KVinteriorS}{\item[Statuen] \KVinteriorS}{}
20   \end{labeling}
21 }{}

```

Mit dieser modifizierten Ergänzung der Katalogeintragsdefinition werden die Unterkategorien von `interior` ausgegeben, auch wenn `interior` selbst nicht definiert ist.

Was noch fehlt ist die Ausgabe der Seiten, auf denen der Katalogeintrag erwähnt wird, wofür der Eintrag `literature` bereit gestellt wird. Dafür wird der Umweg über, oder besser gesagt die Integration von Python gewählt.

Exkurs Python

Integration