

Programmierkurs Python I – WS 10/11

Übung 8

1 Prettyprinting für XML (3 +1 Punkte)

Schreibe einen „Pretty-Printer“ für XML. Das Programm soll eine XML Datei so formatieren, dass jedes Element auf einer eigenen Zeile steht und verschachtelte Elemente eingerückt werden. Leere Elemente sowie Elemente, deren Inhalt ausschließlich Text ist, können auf einer Zeile ausgegeben werden. Der Einfachheit halber kannst Du leere Tags (`<tag attr="blah"/>`) weitgehend ignorieren und so tun, als wären es zwei Tags mit leerem Text dazwischen. (`<tag attr="blah"></tag>`)

```
<kalender>
  <termin>
    <tag>Freitag</tag>
    <start>8.30</start>
    <ende>10.00</ende>
    <was>Python I</was>
  </termin>
</kalender>
```

Bonusaufgabe: Versuche leere Tags zu erkennen, und schreibe das Programm so, dass leere Tags auf einer eigenen Zeile als leere Tags ausgegeben werden.

2 Einfache HTML-Ausgabe (3 + 1 Punkte)

Schreibe ein Programm, dass einfache HTML-Dokumente hübsch formatiert ausgeben kann: Text soll nach 78 Zeichen umgebrochen werden (bzw. an dem ersten Leerzeichen nach dem 78. Zeichen), und Listen (``, ``) solle als Listen dargestellt werden. (`` steht für numerierte Listen, `` für Listen mit einfachen Aufzählungszeichen.) Der Einfachheit halber kannst Du davon ausgehen, dass Listenelemente ihrerseits keine Listen enthalten. In dem ausgegebenen Text sollen keine HTML-Tags mehr erscheinen.

```
<ol>
<li>Eins </li>           wird zu      1. Eins
<li>Zwei</li>           2. Zwei
</ol>
```

Bonus: Erweitere das Programm so, dass es auch verschachtelte Listen geeignet dargestellt werden können.

```

<ol>
<li>Erstens </li>
<li>Zweitens</li>
<ul>
<li> Zweieindrittel</li>   wird zu
<li> Zweizweidrittel</li>
</ul>
<li>Drittens</li>
</ol>

```

```

1. Erstens
2. Zweitens
* Zweieindrittel
* Zweizweidrittel
3. Drittens

```

3 HTML-XHTML-Konverter (Bonus, 3 Punkte)

XHTML ist die Variante von HTML, die in korrektem XML gehalten ist. Einige Unterschiede zwischen den beiden HTML-Varianten sind:

- In HTML wird nicht zwischen Groß- und Kleinschreibung unterschieden, während in XHTML alle Elemente klein geschrieben werden müssen.
- In HTML gibt es Elemente mit optionalem Abschluss-Tag (z.B. `<body>`, `<p>`, ``). In XHTML muss ein Start-Tag immer ein entsprechendes Ende-Tag haben.
- Ausserdem gibt es in HTML einige „leere“ Elemente (ohne Inhalt), die kein Abschluss-Tag haben dürfen (z.B. `
`). In XHTML müssen diese Elemente geschlossen werden (z.B. `
`).
- Es reicht aus, wenn die folgenden Tags betrachtet werden: `p`, `li`, `ol`, `body`, `br`
 - `p` schließt ein vorhergehendes `p`
 - `li` schließt ein vorhergehendes `p`, `li`
 - `ol` schließt ein vorhergehendes `p`, `li`
 - `html` schließt ein vorhergehendes `body`
- Entities (z.B. „Ü“) können mit der Methode `handle_entityref(self, name)` behandelt werden.

Implementiere ein Programm, das HTML nach XHTML konvertiert. Der Konverter darf unvollständig sein: die resultierende Datei muss nicht allen Anforderungen an XHTML genügen, soll aber Groß/Kleinschreibung und die syntaktischen Anforderungen an die oben genannten Tags berücksichtigen.

Abgabe bis Donnerstag, 16.12.10, 14:00 Uhr