

directory index building - 5

engelbert gruber

12.04.2020

x^{MULTI} GRADED STEPS

17:18

Namenssortierung ohne Dateierweiterung

Damit report.pdf vor report-1.pdf sortiert wird.

pathlib.Path hat ein Attribut "stem" das den Dateinamen ohne Dateierweiterung enthält.

vorher

```
# sortiert nach Dateiname
liste.sort(key=lambda p: p.name.lower())
wr_html_file("index-name.html", liste, lambda p: p.name[0].lower())
```

nachher

```
# sortiert nach Dateiname
liste.sort(key=lambda p: p.path.stem.lower())
wr_html_file("index-name.html", liste, lambda p: p.path.stem[0].lower())
```

17:30

Dateien im root immer oben

Jede Seite soll zuerst die allgemeinen Dokumente auflisten.

Beim Einlesen der Verzeichnisse eine eigene Liste `root_files` für die Dateien die nicht in einem Unterverzeichnis sind anlegen.

```
liste = []
root_files = []
...
    if e.is_dir(): # Unterverzeichnisisinhalt holen
        for e1 in e.iterdir():
            if e1.suffix != ".info": # nicht anzeigen
                liste.append(Document(e1))
        else:
            root_files.append(Document(e)) # an liste anhaengen
```

und diese Liste dann an `wr_html_file` zusätzlich übergeben und dort vor der `liste` ausgeben.

17:43

Etwas css.

17:55

Auf der Autorensseite auch nach Dateinamen sortieren

```
# Ausgabe html: sortiert nach autor
liste.sort(key=lambda p: str(p.author + p.path.stem).lower())
wr_html_file("index-author.html", root_files, liste, lambda p: p.author)
```

Funktioniert vorerst.

17:59

Plus 15 Minuten Dokumentation in docs übertragen.