Einführung in Sage

Kurzreferenz

Überlebensregeln

- Mehrere Befehle in einer Zeile trennen: `;`
- Bei Eingaben, die über mehrere Zeilen gehen, kann ein Zeilenumbruch durch `<ENTER>` erreicht werden.
- Das Auswerten eines Blocks erfolgt mit `<SHIFT>+<ENTER>`.
- Ein neues Eingabefeld erhält man durch klicken auf den blauen, horizontalen Balken

Nützliches

- `_` refenziert die letzte Ausgabe (Warnung: unübersichtlich!).
- Löschen aller eigenen Variablen und Zurücksetzen auf den Anfangsstatus: `reset()`
- Aktivieren des Feldes *Typeset* lässt alle Ausgaben von LaTeX rendern.
- Dokumentation mit HTML und LaTeX-Formeln:`<SHIFT>+<KLICK>` auf den blauen Balken startet WYSIWYG-Editor.
- Publish: Im Notebook kann durch klicken des *Publish*-Reiters das Notebook für alle offen gelegt werden.

Hilfefunktionen

- Autocompletion : mit der `<TAB>`-Taste erhält man alle möglichen Funktions- und/oder Variablen-Namen im gegebenen Kontext.
  Dies gilt insbesondere auch für Objektfunktionen (`object.function()`)
- `<command>?` : gibt ausführliche Hilfe zu `command` an.
- `<command>??` : gibt den source-code des `command` an.
- `help(<command>)` : öffnet ein Hilfefenster zu `command`.
- `search_doc('<begriff>')` : Sucht in der Hilfe nach <begriff>.
- Dokumentation:
  - Sage (lokal): `file:///usr/local/sage_4.7.2/devel/sage-main/doc/output/html/en/index.html`
  - Sage: `http://www.sagemath.org/doc/index.html`
  - Python: `http://docs.python.org/`

Dictionaries

- Deklarieren eines Dictionaries:

```
d = {<Index1>:<Wert1>,<Index2>:<Wert2>,...}
```

- Zugriff auf Index:

```
d[<Index>]
```

Zahlen

- Wichtige Funktionen

`abs`	Absolutbetrag
`ceil`	Aufrunden
`floor`	Abrunden
`round`	Runden
`sqrt`	Wurzel
`digits`	Anzahl Stellen

Matrix

- Deklaration

```
matrix([<n>,<m>,[a11,...],[a21,..],..])
```

Vektor

- Deklaration

```
vector([v1,v2,..])
```

Funktionen

- Deklaration

```
def <Name><(a,b,..)>:
    <Code-Block>
    return <ret>
```

Liste

- Konstruktion

```
liste = [a,b,c,...]
liste = list(<sequence>)
```

map_threaded(): Rekursive Auswertung der Funktion auf das Objekt

```
map_threaded(<Funktion>,<Objekt>)
```