



ONYX und PyRope

Prof. Dr. rer. nat. habil. Jochen Merker

Zeitraum

- seit 2015 ONYX
- seit 2023 PyRope in Jupyter Notebooks auf JupyterHub-Server

Module und Einsatz

Module:	Einsatz:
<ul style="list-style-type: none"> • ONYX: 	<ul style="list-style-type: none"> • Prüfungsvorleistungen
N541 Höhere Mathematik I	
N247 Numerische Mathematik	
N586 Mathematik III	<ul style="list-style-type: none"> • Laborversuch
<ul style="list-style-type: none"> • PyRope: 	
N509 Höhere Mathematik II	<ul style="list-style-type: none"> • freiwillige Übung in Stochastik-Tutorium zum Nachholen von Schulstoff

Prof. Dr. rer. nat. habil. Konrad Schöbel

Zeitraum

- 2005-2009 WIMS
- seit 2019 ONYX
- seit 2020 NBgrader
- seit 2021 PyRope

Module und Einsatz

Module:	Einsatz:
<ul style="list-style-type: none"> • WIMS: 	<ul style="list-style-type: none"> • Prüfungsvorleistung und Prüfung • Prüfungsvorleistung und Prüfung
<ul style="list-style-type: none"> • ONYX: 	
Lineare Algebra	<ul style="list-style-type: none"> • Laborversuch
<ul style="list-style-type: none"> • NBgrader: 	
Analysis I + II	<ul style="list-style-type: none"> • noch in Entwicklung, demnächst als freiwilliges Zusatzangebot
<ul style="list-style-type: none"> • PyRope: 	
Lineare Algebra	
Signal- und Datenanalyse	
demnächst Analysis und Lineare Algebra	

Erstellungs- und Betreuungsaufwand

- ONYX:**
- mind. einige Stunden, ein Tag für eine durchschnittliche Aufgabe, mehrere für komplexere Aufgaben
- PyRope:**
- 15 Minuten für einfache Aufgaben, einige Stunden für komplexere
 - Coden von Aufgaben viel einfacher als Aufgabenerstellung

Bei **ONYX** verbringt man die meiste Zeit mit Klicken und Ausprobieren, während man sich bei **PyRope** den Großteil der Zeit über die didaktische Gestaltung Gedanken macht.

Zusammenfassung Erfahrungen

Vorteile PyRope:

- nahezu grenzenlose Möglichkeiten bei der Aufgabengestaltung
- einfaches Einbetten von Links, Grafiken oder Videos durch Markdown
- Syntax-Highlighting
- kurzer Workflow bei der Aufgabenerstellung
- ausgeklügeltes Auto-Scoring
- automatisiertes Testen von Aufgaben mit der Musterlösung sowie mit zufälligen Eingaben
- einfaches Versionieren und Teilen von Aufgaben über GitHub

- typisierte Eingabefelder mit direktem Feedback bei Syntaxfehlern
- hohe Konfigurierbarkeit

Nachteile von PyRope:

- bisher keine Speicherung von Nutzerdaten (in Entwicklung) oder Einbindung in LMS (geplant)
- Vorkenntnisse in Python nötig

In [4]: templates.PythagoreanTheorem().run()

A right angled triangle has a hypotenuse of length 5 and a leg of length 4.

- How long is the other leg?

- What is the area of the triangle?

Submit

02

Herzlich Willkommen zu etwas Stochastik aus der Schule

In diesem Jupyter-Notebook finden Sie einige Aufgaben, anhand derer Sie die eigentlich in der Schule vermittelten Themen aus der Stochastik wiederholen oder kennenlernen können.

Sie können jede der fünf Aufgabensammlungen starten, indem Sie in die zugehörige (In-)Zelle klicken und Shift+Return drücken.

Das dabei verwendete Software-System wird zur Zeit (weiter-)entwickelt, bitte geben Sie uns nach Absolvieren der Aufgaben ein Feedback mittels des Links <https://bildungsportal.sachsen.de/onyxeditor/published/FragebogenEvaluationPyRope1>

```
[1]: import pyrope
pool = pyrope.ExercisePool()
path = 'StochastikTutor1.py'
pool.add_exercises_from_file(path)
runner = pyrope.ExerciseRunner(pyrope.JupyterFrontend())
runner.run(pool)
```

Grundkurs Stochastik

Willkommen zum Grundkurs Stochastik!

Zunächst der Begriff: *Stochastik* kommt aus dem Altgriechischen und bedeutet soviel wie: *Die Kunst des Vermutens*.

01

01 Stochastik Tutorium

02 Aufgabe zum Satz des Pythagoras

03 zugehöriger Code Ausschnitt zu Bild 02

```
class PythagoreanTheorem(Exercise):
    def parameters(self):
        n, n = random.sample(range(1,5), k=2)
        return dict(a=abs(n**2 - n**2), b=2*n*n, c=n**2 + n**2)

    def problem(self, a, b):
        return Problem('
        A right angled triangle has a hypotenuse of length
        leg of length <b></b>.

        * How long is the other leg?
        <<a>>

        * What is the area of the triangle?
        ... <<a>>
        a =Int(treat_none_manually=True),
        A =Rational(elementwise=False),
        )

    def the_solution(self, a, b, c):
        return dict(a_a=a, A_a=a*b/2)

def score(self, a, b, c, a_, A_):
    if a_ is None:
        return 0
    score = 0
    if a_ == a:
        score += 2
    if a_ == a**2 or a_**2 == b**2 + c**2:
        score += 1
    if A_ == a*b/2:
        score += 2
    elif A_ == 0.5*a*b:
        score += 1
    return score

def feedback(self, a, b, c, a_, A_):
    if a_ == a**2:
        return '
        You forgot to take the square root
        after applying the Pythagorean Theorem.
        ...'
```

03



jochen.merker@htwk-leipzig.de

konrad.schoebel@htwk-leipzig.de

www.htwk-leipzig.de