



Towards Practical Programming Exercises and Automated Assessment in Massive Open Online Courses

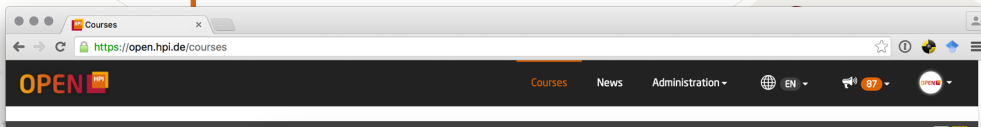
Thomas Staubitz, Hauke Klement, Jan Renz,
Ralf Teusner, Christoph Meinel
Hasso-Plattner-Institute



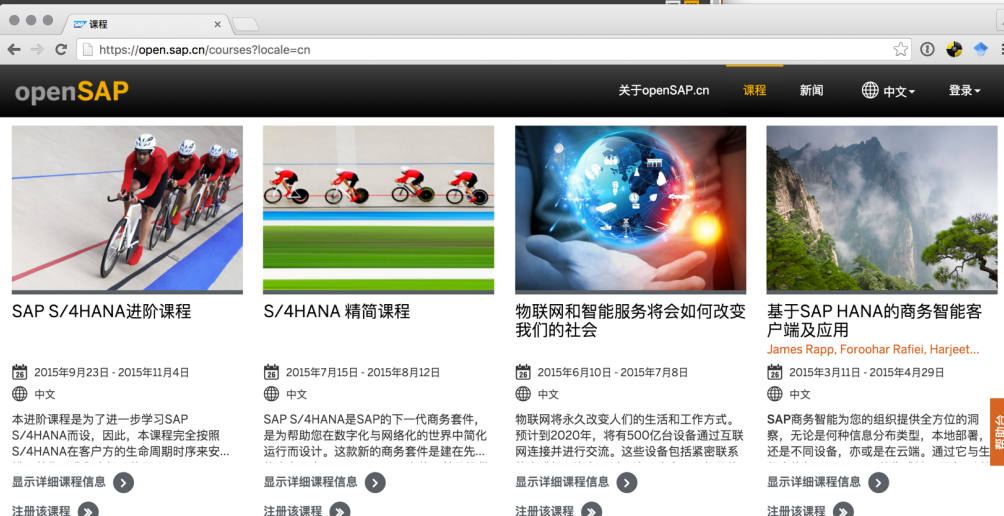
openHPI – Short Introduction

Course Production

Discuss in the
forum



Connect with
fellow learners

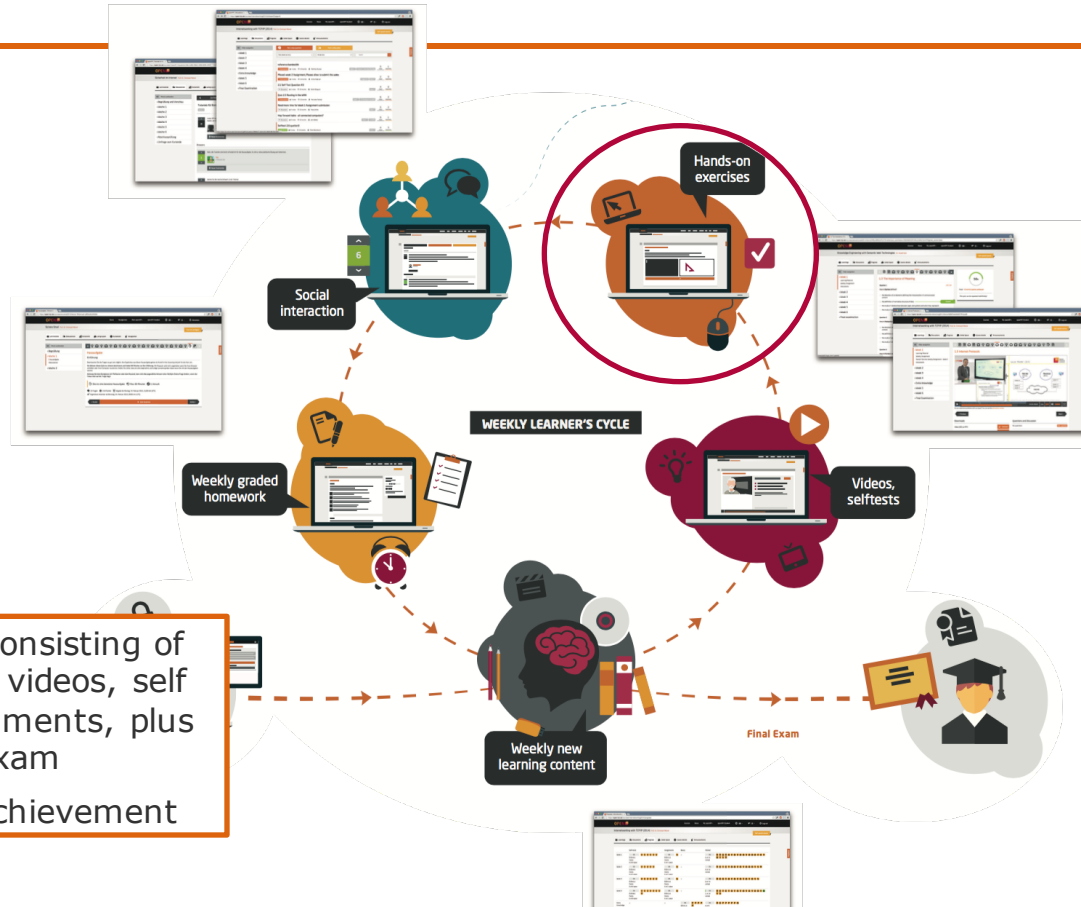


ed and
e HPI

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
2

openHPI – Short Introduction



Learning cycle consisting of weekly released videos, self tests, and assignments, plus a final exam

➔ Record of Achievement

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI

CodeOcean – Where we're heading at

Add 17 + 4

Write a code snippet that prints the sum of 17 + 4 to the screen

Results

1 test files have been executed.

Test File 1 (r111.py)

Passed Tests	1 out of 1
Score	1.00 out of 1.00
Feedback	Well done!
Program Output	Show

Score: 1.00 / 1.00

Aufgabe 4.1.2

Schreibe ein Programm, das der Reihe nach

1. eine Zahl
2. eine Rechenart (+, -, *, /)
3. eine zweite Zahl abfragt, und dann die beiden Zahlen verknüpft.

Ein möglicher Dialog könnte z.B. so aussehen

Erste Zahl: 17
Rechenart: +
Zweite Zahl: 4
Das Ergebnis von 17+4 ist 21

Aufgabe 4.3.2

Schreibe ein Programm, dass zwei Schiffe aufeinander zufahren lässt. Setze dazu das eine Schiff auf die Koordinaten (-100, 0), und das andere Schiff auf die Koordinaten (100, 0). Lasse sie dann aufeinander zufahren, indem sie sich in jedem Schritt immer um einen x-Unterschied von 1 aufeinander zubewegen.

python3 /usr/lib/python3.4/webpython.py -f aufgabe432.py

Why do we need it?

- Introductory programming courses attract large amounts of participants from all ages (Java für Einsteiger – 14,000, PythonJunior2014 – 10,000, PythonJunior2015 – 8000 Learners from 10 – 80 y.)
- Study/Survey by European Commission related to web skills:
 - IT professionals consider MOOCs the best way to learn such abilities.
 - Learners value practical experience over theoretical content
- Radical differences in the way different concepts are grasped by students
 - Learning to program does not only involve acquiring complex knowledge but also practical skills
 - Gaining programming expertise requires rigorous practice

(Berges et al, Feldman and Zelenski, Robins and Rountree, Vivahainen et al, Douce et al)

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI

CodeOcean – How it all began

- Literature review on practical programming tasks and automated assessment
 - Analysis of current practices in MOOCs – Pros and cons of different scenarios
 - Map existing tools and components
 - History of automated assessment tools
- Decision making **if**, **why**, and **what** should we develop

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI

Scenarios for Practical Exercises (1/3)

	Editor and Execution Environment	Assessment
S1:	Local (native or VM) 	In Platform 
S2:	Third party online tool 	Depending on tool (e.g. STEAP) 
S3:	In Platform 	In Platform 
S4:	In Platform/Browser 	Browser 

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
7

Scenarios for Practical Exercises (2/3)

	+	-
S1: 	- Scalability - Allows to use existing specialized or favorite tools	- Heterogeneity (Support) - Sparse data (final result only)
S2: 	- Homogeneity - Installation free - Low Barrier - Scalability outsourced	- Dependency on third party - Limited possibilities - Sparse data

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
8

Scenarios for Practical Exercises (3/3)

	+	-
S3: 	• Homogeneity • Installation free • Low Barrier • Rich data • Complete control • Wide range of supported languages	• Scalability issues • Costs • Server • Development • Security issues
S4: 	• Resource efficient	• Cheating possibilities • Limited set of languages

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI

Component Landscape

Educational online programming tools	• Codewars, CodingBat, CodeHunt • ...
Online development tools	• CodePen, jsFiddle, repl.it • ...
Web based IDEs	• Cloud9, Codio, Nitrous.IO • ...
SaaS/PaaS	• Web-based IDE (e.g. Cloud9) • Code hosting, issue tracking (e.g. GitHub) • Execution platform (e.g. Heroku)

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
10

Automated Assessment (1/2)

- Categories
 - Dynamic: Code is executed, results are analyzed
 - Static: Code is analyzed
- What is assessed?
 - Functionality, completeness, performance
 - Code quality and style
 - GUIs
 - Testing skills
- How is assessed?
 - I/O based: predefined input values → verification of expected output
 - Challenge: Output format might differ from expected while still producing the correct result (e.g. Sonnabend vs. Samstag, whitespace, spelling mistakes)
 - Based on testing frameworks: Unit testing, Acceptance testing

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
12

Automated Assessment (2/2)

■ Automated assessment ...

- ... increases the need for carefully designed assignments as poor design cannot be compensated by assessment
 - ➔ less time for grading
 - ➔ more time for designing
- ... the larger the size of the course, the better the ROI
- ... inevitable for MOOCs (Alternative: Peer Assessment)

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
13

Conclusion

- Practical programming tasks are essential for programming courses
- In the context of MOOCs automated assessment is key
- Comparing the benefits and drawbacks of the scenarios that have been introduced in the beginning:
 - We need a flexible solution that is able to handle things differently depending on a course's main target group
 - Beginners benefit more from a browser based environment
 - Advanced users will prefer familiar tools
- The landscape of existing programming languages is wide spread:
 - Providing a new programming environment for each course is not desirable
 - We need a solution that supports multiple programming languages

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
14

Future Work

- Introduction of our own tool: CodeOcean – A tool for browser based code execution and automated assessment
 - Implemented
 - Open Source (<https://github.com/openHPI/codeocean>)
 - Employed in three courses on openHPI
- In depth evaluation of the results of these courses, particularly analysis of the students' code submissions and feedback
- In depth comparison of the tools that have been employed in various MOOCs and the experiences that have been made.
- ...

Programming Exercises

Thomas Staubitz,
Research Associate,
HPI
15



Questions?

Thomas.Staubitz@hpi.de



Thomas Staubitz
Research Associate
Hasso Plattner Institute