We conducted an A/B testing in a regular Python course with 6,067 learners and offered tips for our treatment group. Based on our evaluation, we answer our research questions:

RQ1. How do tips influence the help-seeking behavior?

The introduction of contextual tips does not negatively affect the usage of peer-to-peer help systems. Instead, learners showed a higher chance of requesting comments when they also used tips. Some participants not finishing an exercise reduced the number of *Requests for Comments* when tips were available. This observation might suggest that tips were able to answer upcoming questions for them. Throughout the first half of the course, nearly 25% of learners regularly revealed our contextual hints while implementing an exercise.

RQ2. Which learners profit more from tips than others?

Learners self-identifying as beginners are the user group that benefits most from contextual tips. The higher the skill level at the beginning of a course is, the fewer tips are used. Our research emphasizes that more challenging exercises increase the learners' need for additional hints.

RQ3. Do tips have an impact on key metrics such as the completion rate, working times, or scores?

According to our evaluation, the mere availability of tips slightly increases the mean working time for learners by 2.7%. In particular, beginners using tips to get support spend more time within the exercise and are more likely to use other assistance features. Besides that, neither an impact of tips on the completion rate nor the scores of participants was discernible.

Overall, our findings highlight that tips are valued by novices as a relevant part of their help-seeking behavior. Contextual tips are an additional offer not impeding existing assistance features. Answers from subsequent surveys indicate the great potential tips can have and motivate us to continue researching the impact of tips. The introduction of tips as presented throughout this paper supports students to get contextual assistance within the learning environment of a MOOC.

REFERENCES

[1] Paolo Antonucci, Christian Estler, Durica Nikolić, Marco Piccioni, and Bertrand Meyer. 2015. An Incremental Hint System For Automated Programming Assignments. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, Vilnius Lithuania, 320–325. DOI:http://dx.doi.org/10.1145/2729094.2742607

[2] Kyle J. Harms, Jason Chen, and Caitlin L. Kelleher. 2016. Distractors in Parsons Problems Decrease Learning Efficiency for Young Novice Programmers. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. Melbourne, Australia, 241–250. DOI: http://dx.doi.org/10.1145/2960310.2960314

[3] Ville Isomöttönen, Antti-Jussi Lakanen, and Vesa Lappalainen. 2011. K-12 Game Programming Course Concept Using Textual Programming. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11*. Dallas, TX, USA, 459. DOI:http://dx.doi.org/10.1145/1953163.1953296

[4] Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2019. A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. *ACM Transactions on Computing Education* 19, 1 (Jan. 2019), 1–43. DOI:http://dx.doi.org/10.1145/3231711

[5] Ada S. Kim and Andrew J. Ko. 2017. A Pedagogical Analysis of Online Coding Tutorials. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. Seattle, WA, 321–326. DOI:http://dx.doi.org/10.1145/3017680.3017728

[6] Sándor Király, Károly Nehéz, and Olivér Hornyák. 2017. Some Aspects of Grading Java Code Submissions in MOOCs. *Research in Learning Technology* 25 (2017), 16. DOI:http://dx.doi.org/10.25304/rlt.v25.1945

[7] Samiha Marwan, Nicholas Lytle, Joseph J. Williams, and Thomas W. Price. 2019. The Impact of Adding Textual Explanations to Next-Step Hints in a Novice Programming Environment. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. Aberdeen UK, 520–526. DOI:http://dx.doi.org/10.1145/3304221.3319759

[8] Silvia Muller, Monica Babes-Vroman, Mary Emenike, and Thu D. Nguyen. 2020. Exploring Novice Programmers' Homework Practices: Initial Observations of Information Seeking Behaviors. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 333–339. DOI: http://dx.doi.org/10.1145/3328778.3366885

[9] Thomas W. Price, Yihuan Dong, Rui Zhi, Benjamin Paaßen, Nicholas Lytle, Veronica Cateté, and Tiffany Barnes. 2019. A Comparison of the Quality of Data-Driven Programming Hint Generation Algorithms. *International Journal of Artificial Intelligence in Education* 29, 3 (Aug. 2019), 368–395. DOI: http://dx.doi.org/10.1007/s40593-019-00177-z

[10] Sebastian Serth, Ralf Teusner, Jan Renz, and Matthias Uflacker. 2019. Evaluating Digital Worksheets with Interactive Programming Exercises for K-12 Education. In *IEEE Frontiers in Education (FIE)*. Cincinnati, USA. DOI:http://dx.doi.org/10.1109/FIE43999.2019.9028680

[11] Thomas Staubitz, Hauke Klement, Ralf Teusner, Jan Renz, and Christoph Meinel. 2016. CodeOcean - A Versatile Platform for Practical Programming Excercises in Online Environments. In *IEEE Global Engineering Education Conf. (EDUCON)*. Abu Dhabi, 314–323. DOI:http://dx.doi.org/10.1109/EDUCON.2016.7474573

[12] Ralf Teusner, Thomas Hille, and Thomas Staubitz. 2018. Effects of Automated Interventions in Programming Assignments: Evidence from a Field Experiment. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale - L@S '18*. London, UK. DOI: http://dx.doi.org/10.1145/3231644.3231650