Our goal for this language is simple: create an accessible, visually based language that can be easily used by elementary and early middle schooler students while still displaying more of the actual syntax of a standard programming language (in our case, Python). Most block coding languages, like Scratch, have more simplistic syntax on the blocks, writing commands in English rather than allowing the student to see the language behind the blocks. These traditional block coding languages certainly have merit. They are a great way to get students excited about the field of computer science and introduce the problem-solving skills that are important to programming. However, the jump from block coding to typing code can be quite jarring. The intent of tryPy is to provide a more student-oriented transition to typing code by combining the easy-to-conceptualize blocks with Python.

We chose Python as the language for our intermediate blocks due to its wide usage and simpler syntax. Since the code will appear in blocks, it makes the most sense to first implement intermediate block coding using a language with easier-to-visualize syntax than other languages like C. Additionally, since Python is so widely used, it is a skill that will be widely useful. Like with any spoken language, it is easier to learn a language if you begin young and tryPy is a great way to introduce it early in elementary school.

To gather more information about the potential utility of such a language, we interviewed an elementary school teacher, who will be referred to as Mrs. C. Mrs. C uses block coding to introduce compute science concepts to her students and participates in Hour of Code with third, fourth, and fifth graders each year. In our interview, we asked questions regarding the utility of block coding as it is being used in the classroom now, the usefulness of introducing compute science concepts to students at a young age, and potential utility of an intermediate block language such as tryPy.

Our first questions to Mrs. C were regarding block coding as it is currently implemented. Mrs. C uses Hour of Code to reach students outside of her own classroom, so to share insight on the largest sample size, many of her responses were surrounding Hour of Code over the past 3 years. Mrs. C primarily focused on two activities with students during Hour of Code. Of the two activities, one was geared toward the younger students (third and fourth grade) and the other was geared toward the older students (fourth graders with previous coding experience and fifth graders). Mrs. C noted that between the two activities, despite being in the same block language, there were certain differences in how you need to connect blocks to complete the tasks. She said this caused some confusion in students who had done both options. Other than the difference in the syntax in the two activities, Mrs. C thought the blocks were a great way to introduce programming to the students. There was an option to see the code behind what the blocks were doing, which was too great of a jump from the blocks for students to fully understand what the code did but was interesting feedback on current implementation of blocks.

Our next questions to Mrs. C were regarding block coding and the introduction of computer science concepts to young students. In her opinion, block coding is a good introduction to computer science and has certainly inspired an increased interest in computer science within her students. She even mentioned that some teachers now use block coding projects as a reward for students if they complete other classwork early since the students enjoy it so much. When asked about the skills that block coding has helped her students, learn, Mrs. C said that outside of computer and coding skills, participating in Hour of Code and otherwise using block coding has helped teach her students about perseverance and the importance of trying different solutions to solving a problem. She also noted that it helps students learn about problem solving skills that apply to all other fields, both academic and based in real-life.

We closed out our interview with Ms. C by asking about the utility of an intermediate block programming language. We explained the concept of tryPy and its visualization of code that is syntactically similar to Python. She was interested in the concept and agreed that it would be most useful as a “half-step” from pure block coding to fully typed coding. She said that from her experience, students should still start out with block coding, like Scratch, and then move into an intermediate block language such as tryPy in fifth grade (if they have sufficient block coding experience) or middle school. With the increased realism of tryPy, students would be able to more easily transition from blocks to typing, and the retention of interest between elementary and middle school would likely increase.

Our new language, tryPy has the potential to revolutionize the way computer science is taught to young students. It would bridge the gap between purely block coding and purely typed coding, making the transition smoother for students who start working with blocks and maintaining students’ interest in computer science throughout their elementary and middle school years. Through an interview with a teacher who currently uses block coding in her classroom, we have verified these thoughts from an educator’s point of view. tryPy brings increased accessibility to students in elementary school while also increasing interest in computer science as it appears in high school, college, and beyond.