Constructing Knowledge Through Python Coding

Our Intro to Python computer science course content emphasizes students applying and exploring the information presented. To provide a thorough introduction to Python programming, a code editor accompanies each page with new concepts so students can see for themselves how the computer responds to code.

In addition, the content provides code snippets to get students started as well as suggested avenues for investigation. At the end of each module, students are asked to apply the skills they learned in a lab.

Auto-graded assessments and instant learner feedback

Students receive immediate, rich feedback. In addition to correctness feedback (i.e. right or wrong), students will also see an explanation with the complete solution.

There are a wide variety of questions - all of which are auto-graded, giving students a sense of their understanding of the material right after they are introduced to it and as they attempt harder and harder problems. Auto-graded assessments also include five coding exercises at the end of each module, where students are evaluated on the code they write.

Lowering the Barrier to Entry

This Python intro course material reflects the need for computer science education to meet students where they are. Like any specialized community, computer science has its own jargon. The formal teaching of computer science should not burden students with the assumption that they are fluent in this special language.

The material is presented in smaller units that are more manageable for the students. The same vocabulary and concepts are covered, but in a more approachable way - state things as plainly as possible, and, when appropriate, use images, tables, or lists.

Another way in which our intro to Python programming content is more approachable is that it uses many small programs instead of one large program. Research shows that a variety of smaller problems increase student performance and reduce stress. Using many small programs leads to students spending a sufficient amount of time on their work, and they do not wait until the last moment to begin their work.

Encouraging Customization Through Modularity

This content is not a one-size-fits-all solution. Instead, it implements a modular format. Natural breakpoints occur in the curriculum where instructors can make necessary changes. Instructors can rename, re-order, or remove units and can author new material using Codio's powerful curriculum content authoring tools, giving complete flexibility when designing the learner's experience.

What's Included

- Interactive content in Python with visuals and minimal text for maximum hands-on engagement
- Example code snippets that can be copied
- A fully-featured web-based IDE
- Run code with the click of a button
- Use Code Visualizer to see "under the hood"

Parsons Problems and other formative assessments to help students check their understanding