Creating a Python Curriculum for Middle and High School

Today's Presentation

- Introduction
- What is curriculum?
- Developing curriculum for diverse populations: barriers to entry
- Developing curriculum for high-school beginners
- Why another coding curriculum?
- Experiments and insights

Introduction

- Former Music Teacher with EdD in Music Education Curriculum and Assessment
- Currently Software Engineer in Test, writing Python programs
- Love how tech, arts, education, and opportunities intersect. Code as creativity.

Defining Curriculum

Curriculum may incorporate the planned interaction of pupils with instructional content, materials, resources, and processes for evaluating the attainment of educational objectives

Vision

Python as a class offered **during** the public school day.

Influence the direction of technology education in the mainstream curriculum.

Make the consumer the creator, and make teachers learning-partners. Take the “sage” off the stage.

Curriculum for Diverse, General Populations

Barriers to Entry

- Lack of interest
- Varying levels of exposure to technology
- Mixed grade levels = different maturity
- Different reading and mathematics abilities
- Teachers with varying or no experience in tech

More Barriers to Entry

- Tech curriculum happening in a variety of ways, but all part-time: once a week, after school, paid programs

- Lack of support for teachers who want to teach technology, but are afraid to learn a programming language

- Equipment problems: lack of working computers and working networks; cost of new equipment

Developing curriculum for high school students

- Lessons need to be aligned to local and national educational standards (in the United States)

- Assessment efforts should, in the best circumstances, reflect educative or authentic practices to engage students

- High school curriculum ought to prepare and inspire students to explore on their own

- Give students many practice opportunities prior to assessment

Teach Code Pilot Program

- One teacher, 25 students (randomly/involuntarily placed), one programming teacher (me) in highest-poverty urban district. Used Mac computers and codeskulptor.org (web - based Python IDE) due to technology restrictions

- Lessons created in pairs with teacher, then by teacher and students

- Students built programs of increasing difficulty, with their own program project at the end of class

Common Core 1

Common Core Standards Intro to Python

English Language Arts Standards » Science & Technical Subjects » Grade 9-10

CCSS.ELA-Literacy.RST.9-10.3 Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.

CCSS.ELA-Literacy.RST.9-10.4 Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to*grades 9–10 texts and topics*.

CCSS.ELA-Literacy.RST.9-10.5 Analyze the structure of the relationships among concepts in a text, including relationships among key terms (e.g., *force, friction, reaction force, energy*).

CCSS.ELA-Literacy.RST.9-10.7 Translate quantitative or technical information expressed in words in a text into visual form (e.g., a table or chart) and translate information expressed visually or mathematically (e.g., in an equation) into words.

CCSS.ELA-Literacy.RST.9-10.10 By the end of grade 10, read and comprehend science/technical texts in the grades 9–10 text complexity band independently and proficiently.

Common Core 2

English Language Arts Standards » Writing » Grade 9-10

CCSS.ELA-Literacy.W.9-10.2 Write informative/explanatory texts to examine and convey complex ideas, concepts, and information clearly and accurately through the effective selection, organization, and analysis of content.

CCSS.ELA-Literacy.W.9-10.2a Introduce a topic; organize complex ideas, concepts, and information to make important connections and distinctions; include formatting (e.g., headings), graphics (e.g., figures, tables), and multimedia when useful to aiding comprehension.

CCSS.ELA-Literacy.W.9-10.2d Use precise language and domain-specific vocabulary to manage the complexity of the topic.

CCSS.ELA-Literacy.W.9-10.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. (Grade-specific expectations for writing types are defined in standards 1–3 above.)

CCSS.ELA-Literacy.W.9-10.6 Use technology, including the Internet, to produce, publish, and update individual or shared writing products, taking advantage of technology's capacity to link to other information and to display information flexibly and dynamically.

Common Core 3

English Language Arts Standards » Speaking & Listening » Grade 9-10

- CCSS.ELA-Literacy.SL.9-10.1 Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grades 9–10 topics, texts, and issues, building on others' ideas and expressing their own clearly and persuasively.

- CCSS.ELA-Literacy.SL.9-10.1a Come to discussions prepared, having read and researched material under study; explicitly draw on that preparation by referring to evidence from texts and other research on the topic or issue to stimulate a thoughtful, well-reasoned exchange of ideas.

- CCSS.ELA-Literacy.SL.9-10.1b Work with peers to set rules for collegial discussions and decision-making (e.g., informal consensus, taking votes on key issues, presentation of alternate views), clear goals and deadlines, and individual roles as needed.

- CCSS.ELA-Literacy.SL.9-10.1c Propel conversations by posing and responding to questions that relate the current discussion to broader themes or larger ideas; actively incorporate others into the discussion; and clarify, verify, or challenge ideas and conclusions.

Sample Lesson Plans

Unit: Learn to Program in Python	Lesson Title: What is Python?
Overarching Question: How is Python used in simple programming tasks?	
Daily Question/Aim: Why use Python to program computers?	

CCSS: Common Core Standards Aligned with Lesson							
RST LIT	9-10.3	9-10.4	9-10.5	9-10.7	9-10.10		
Writing	9-10.2	9-10.2a	9-10.2d	9-10.6	9-10.7	9-10.8	
Speaking/Listening	9-10.1	9-10.1a	9-10.1b	9-10.1c	9-10.1d	9-10.2	9-10.5
Language	9-10.6						
Materials Needed: Powerpoint, Smartboard, computers with internet connection							

Sample Lesson Plan: Part 2

Mini Lesson/Introduction:

1. Use powerpoint to introduce the very basics of Python programming
2. Go over the details about codeskulptor.org carefully with students.

Student Practice Opportunities:

1. Allow students to use codeskulptor.org either along with powerpoint presentation OR afterward so they can practice the code examples, code types, as well as the program functions

Formative Assessment Strategies:

1. Use thumbs up, side, or down during instruction to find out how students are getting along with the software.
2. Use checks (minus, regular, plus) to gauge student immediate understanding of software IDE
3. Use exit cards/e-mails to find out questions

Summary:

1. Students should know two basic aspects of the Python programming language, variables and strings
2. Students are made aware of some Python vocabulary that they will be using casually BEFORE being asked to work with this formally
3. Students will begin to work with codeskulptor.org and learn to work with the internet software to understand how to write code and get a response about if their code is correct or incorrect.
4. FOLLOW UP: In agile meeting next Monday, students can present challenges and goals.

Sample Lesson: Making Cookies

- The Task: Print out a cookie recipe (a lesson in variables, strings, variable assignment, syntax, and printing)

- Introductory Lessons: Variables, strings, variable assignments, printing variables and strings together using proper syntax

- Pacing Problems: More turtle than rabbit. Students felt frustrated by doing too much and taking things too fast.

- Student (mis)Understandings: Without a combination of direct instruction, powerpoints and notes, sample code, and practice code with their own inputs, students felt lost

Making More Cookies

The next step asked students to perform math using the recipe. Here is an example of the task in the IDE they used

```
1  #These are my variables for the cookies
2  #HINT: add the ingredients here
3
4  unsalted_butter = .5
5  brown_sugar = 1.5
6
7
8  #These are my mathematical equations for the cookies
9  #Hint: add the formula here and print it out
10 print "Use", unsalted_butter * 3 ,"cups of unsalted butter."
11 print "Use", brown_sugar *3 ,"cups of brown sugar"
12
13
14
```

```
Use 1.5 cups of unsalted butter.
Use 4.5 cups of brown sugar
```

CodeSkulptor was built by Scott Rixner and is based upon CodeMirror and Skulpt.

Responses of Pilot Participants

- Alice (teacher): “This was one of the best classes of my teaching career.”

- Cathy (Assistant Principal, Business): “The response we have had from our students has been very positive. We have every level of student in our Computer Science classes and they have all had great successes…The level of student centered learning demonstrates that challenging problems are motivation when approached in a safe learning environment.”

- Parents (via Alice): “I explained that I didn't think there was another high school in NYC that offers this type of class within the actual school day. I also explained that we are creating a "living curriculum.” Each of the three parents were so happy that their child was in this class. One mother said this is the only class her son talks about from the moment he comes home until he goes to bed. They were so happy to see their child excited about school enough to talk to them at home about the class.”

Responses: Students

- Programming should be taught earlier, possibly 10th grade.
- More PowerPoints for explanation and more practice problems.
- Have a programming sequence, maybe Programming 1 and Programming 2.
- They said (more than one) they now had the ability to build a program. This was something they hadn't tried or done before.
- They would like the opportunity explore other languages.
- One student said that he had trouble really understanding Python and what we were doing in class. Once he was able to use Python, he became more confident in other areas, not just in his ability in this class.

Dalwinder's Response

"Over the progression of our programing class, I have learned a great deal of information. I feel that this class has given me the intricate knowledge necessary to cultivate programs and all sorts of things that deal with the personal computer. We, as a class, have learned how to use python; **python is a programing language that is intended for starters. It can be used to produce basic programs such as a calculator or even a modest game like hangman.**

Over the course of this class, **I learned how to design applications and make them useful. We learned how to construct a basic four-function calculator.** I thought that it was going to be an uninteresting class, but I recognized how useful it is in the work that I do. In my unused time I build computers for people who want them. This class presented me with what it's like to program from scratch. My work is more on the physical end of building computers but **thanks to programming I learned the basic properties of software. I endorse this class to students who are interested in this field.**"

Lessons Learned

- The good: in the end, students and teacher found great value and skill in exploring a programming language

- The bad: not so much terrible as experimental: navigating the presentation of material to new programmers while myself digging through concepts

- Next steps: refine the lessons so that they are taught in a way that allows students to slowly build mastery while experimenting with concepts

Questions?

Thank you!