

# Code.org CSP '16-'17 Revisions Summary

## Background

In the '15-'16 school year Code.org ran a 100 teacher pilot of our new [CS Principles curriculum](#). The primary goal of the pilot was to collect and incorporate feedback on the new curriculum, tools, and professional development in preparation for the official AP launch in the '16 - '17 school year. In general both the curriculum and professional development have been very well-received, and we were happy to find many of the decisions we made while developing the course affirmed by students and teachers. In compiling this feedback we were also able to identify several key areas in which we could improve the course to make it more manageable to teach and engaging for students. This document summarizes how we plan to address and incorporate the most prevalent items of feedback we received from pilots as we begin working on the '16-'17 version of the curriculum.

## Feedback themes

The five most significant themes of the feedback we received are summarized below:

1. [Lack of pacing suggestions and too much material to cover in 1 school year](#)
2. [Need more formative and summative assessment supports](#)
3. [Unit 2: The Internet, is too long and monotonous for students and does not have enough opportunities for creation.](#)
4. [Classrooms want programming earlier](#)
5. [Course needs more front-matter to fulfill common requests](#)

## Addressing feedback and Course re-organizing

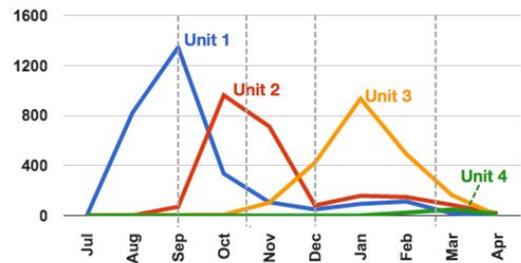
In order to respond to the themes outlined above, especially those related to pacing, length, and student engagement, the team conceived of ways to trim and re-organize the content of the course. The left side of the diagram below shows the course structure of our pilot year curriculum. Each unit naturally fell into a pattern of having 2 "chapters", often with each culminating in a practice PT.

The right side of the diagram shows the course structure for the '16-'17 AP launch year of CSP. As you can see, the course content is largely the same: the same sets of learning objectives are addressed, most of the lessons will be the same, only grouped and sequenced slightly differently. The plan represents a significant change in the flow of the course, but not in the content or materials. The diagram below highlights the key points in the v2 plan. This plan has been shown to a select group of pilots who understand and support these

## Feedback by the numbers

Method	Volume
In-person academic year workshops	20 one-day workshops
Support tickets coming in through mails to <a href="mailto:csproject@code.org">csproject@code.org</a> or <a href="http://studio.code.org">studio.code.org</a>	~172 tickets solved
Forum posts <i>Not including submissions required for fulfilling professional development requirements</i>	~1300+ forum posts
1:1 teacher interviews conducted by Outlier	26 out of 26

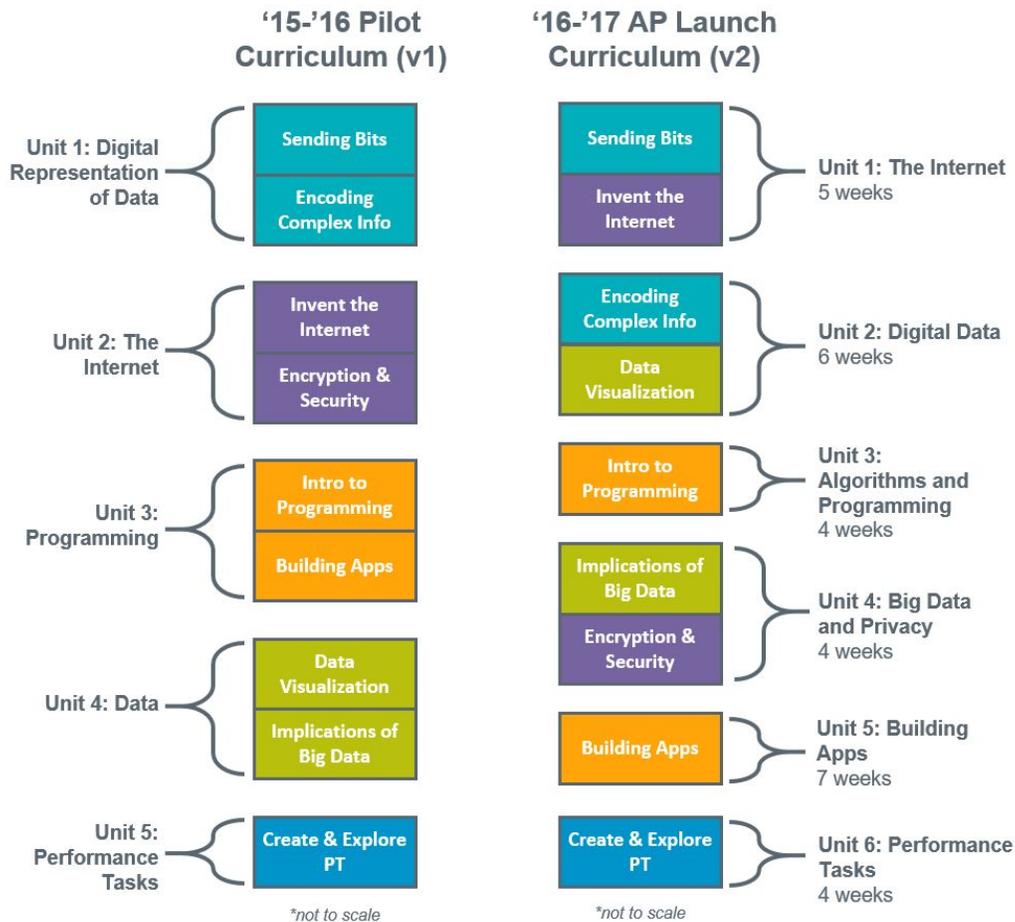
Code Studio data showing start date per unit for # of students in pilot cohorts. Vertical lines show target start dates.



NOTE: most schools have not started Unit 4 yet.

Data collected Aug. 2015 - Mar. 2016

changes.



## Commentary on New Units

### Unit 1: The Internet

We saw an opportunity to connect the protocol-building activities that students were already doing in the “Sending Bits” chapter of v1 Unit 1 all the way to the full invention of the Internet for one complete thought.

### Unit 2: Digital Data

The work that students do to understand how complex information is encoded, such as images, and compressed text is a natural precursor to manipulating, visualizing, and extracting meaning out of structured data. Students will more actively create computational artifacts in this unit than in v1, therefore partially addressing Theme #3.

### Unit 3: Algorithms and Programming

In v2, we moved material on encryption out of the Internet unit. This opened the opportunity to address theme #4 - getting to programming earlier. The programming unit in v1 naturally broke into two parts. The first part, an introduction to basic concepts using turtle drawing, will now land here. It also allows us to partially address

theme #3 - opportunities for creation early in the course. More complex programming topics will now come later Unit 5.

### Unit 4: Big Data and Privacy

In v1, these two topics were split apart but as we've all seen in the news recently, the topics of data, privacy, and encryption easily form a coherent and compelling story together. Theme #3 is partially addressed by this unit.

### Unit 5: Building Apps

In the v1 sequence of the curriculum, students finish all of the programming they will do in the course about 4-6 weeks before attempting the Create Performance Task. While we haven't directly received this feedback that this a problem, we believe that ending the course with programming will be a smoother transition into the completing the actual Create Performance Tasks.

### Unit 6: Performance Tasks

Unit 5 in v1, the only change is the order to the Tasks  
(Create now comes first).

*A longer version of this report, explaining changes and responses to feedback is available [here](#).*