STUDENTS at the CENTER HUB

PROGRAMMING FOR PERSONALIZATION: Using Computer Science Resources in Support of a Student-Centered Learning Environment

What are the key skills humans must possess in order to be considered literate?[1] This question has taken on even more urgency for educators in the 21st century as they try to help students make use of rapidly evolving technologies and prepare for jobs that do not yet exist. The centrality of digital literacy (including programming, game design, etc.) is still the subject of debate in everything from blogs to research papers. But even as the debate continues, valuable resources and tools are increasingly available to help students program, code, produce and create their own products, while building skills that can help them pursue careers in today's labor market.

Digital literacy and computational thinking skills do not have to be limited to a computer science classroom—there are many ways to enhance these skills across discipline, and with limited expertise.

The goal of this e-guidebook is to provide a set of educational resources that support educators in building digital literacy competencies both in- and outside computer science classrooms. Along with tool introductions, we also provide accompanying strategies/tips for utilization of these resources to support and enhance the personalized, student-centered learning environments that educators are trying to create.

When considering tools to include for this guide, we sought to provide diverse project possibilities that incorporate computer science instruction across academic disciplines. This set of five resources were selected based on popularity and ease of access. Several more related resources are included in the "Additional Resources" section for interested educators to explore and use.

[1] http://www.edutopia.org/literacy-computer-programming

Contents

RESOURCES TO EXPLORE IN THIS TOOLKIT

We want to include more resources here! Please contact us if you'd like to suggest a CS resources/tools/platforms for us to include.

1

SCRATCH >

Scratch (game design) is a visual programming language designed for all skill levels, offering the ability to design stories, games, and use pre-developed "blocks" of code to enable various commands. These "blocks" of code can be dragged together to create a program, but users can also write their own unique programs. Scratch is often geared toward younger audiences, but the projects shared online reflect its usefulness for a range of ages, subject matter, expertise and interests. Projects can be as simple as an animated slideshow using basic keyboard cues ("When the space key is pressed, change backdrop"), or quite complex, such as a video-sensing project that incorporates variables and conditional statements. Scratch has a vibrant community of users, who are active on sites like ScratchEd, a growing platform for educators to share stories, swap resources, and connect with other Scratch users and enthusiasts.

OPPORTUNITIES FOR SCRATCH TO SUPPORT STUDENT-CENTERED STRATEGIES FOR LEARNING

Scratch enables a **personalized learning experience**—students can share information in ways that are exciting and playful for them, such as embedding familiar images or adding music clips.

The Scratch interface **invites exploration**—a helpful question mark icon can be dragged onto any programming block, resulting in immediate help in the form of a tutorial or video demonstration. Students can work at their own pace and seek help when they need it.

The program works on a variety of platforms, which can **support anytime, anywhere learning** and allow students to work on projects in a preferred format (like an iPad or even a mobile device).

Existing projects that are shared on Scratch's website **can be "remixed,"** allowing students to not only view code, or blocks of code within complicated projects, but more importantly, students can rewrite the code, or move said blocks of code around and get instant feedback onscreen. This "see inside" feature also allows students to support each other in troubleshooting and editing each other's code.

STRATEGIES FOR IMPLEMENTATION ACROSS DISCIPLINES

MATH

Students can work on and apply a range of mathematical concepts using Scratch. Scratch provides math "scripts" for concepts including averages, square root, prime numbers, factorials, and slope. ScratchEd offers a range of lessons that can be implemented from Pre-K – High School level mathematical concepts. For example, a pair of teachers from the Carroll School in Lincoln, MA who co-teach a 9th grade math class shared an assessment activity and rubric to complement their Scratch curriculum. The curriculum is a month-long unit they developed to teach algebra and geometry concepts through Scratch. Their contribution also includes a video of them reflecting on the lesson and the rubric. Their rubrics include the opportunity for students to engage in self-assessment as well as a one-on-one assessment conversation with their teacher.

SOCIAL STUDIES/HISTORY

Students can build a "chat bot" using Scratch. In social studies, have students assume the role of a historical figure and program a "chat bot" to interact with users. Users ask questions like, *What is your name? When were you born? What are you famous for?*, and students demonstrate knowledge by pre-programming accurate historical information. To get started on a similar assignment, check out this tutorial, Scratch Bot Assignment.

SCIENCE

Students in introductory science courses can create interactive stories or animations to model concepts like relativity or the solar system, or design games to have users test their knowledge of chemical equations. Scratch Studios–Science! (one of many project galleries) is a great resource for browsing existing projects that offer opportunities for "fun" and personalization while still demonstrating mastery of content. Doing a unit on gravity and inertia? Check out Extreme Environments: Building a Simple Lunar Lander. This project bundles computer science and engineering with the concepts of gravity and inertia wrapped within the narrative of building a lunar lander (in phases, simple, enhanced, and enhanced with joy stick!).

2

CODECADEMY >

Codecademy (web design) is an online platform that offers free coding classes in nine programming languages. While the courses are free, a “pro” version (which requires a license) offers quizzes, advisor support, and other personalized tools. The platform offers full-length courses (each about 10 hours long), in addition to a set of shorter coding projects that take 30 minutes each to complete. Both the short- and long-form courses can be implemented inside and outside the classroom. Short activities include “Building your own galaxy”, while the **web design** course is an example of a long-form course. The web design course is one of the more popular choices for students, as they can “own” and design their own space online.

OPPORTUNITIES FOR CODECADEMY TO SUPPORT STUDENT-CENTERED STRATEGIES FOR LEARNING

A **competency-based approach** requires students to complete certain tasks multiple times before moving forward—and if a student is struggling, the tool will provide opportunities to review previous content.

Educators can create classroom accounts at no charge to **track student performance from afar** (and jump in when necessary), providing opportunities for students to be responsible for their own learning and their progress, while getting support when they need it.

The activities are **hands-on** right away, inviting students to immediately begin exploring lessons, engaging with tasks, and learning.

STRATEGIES FOR IMPLEMENTATION

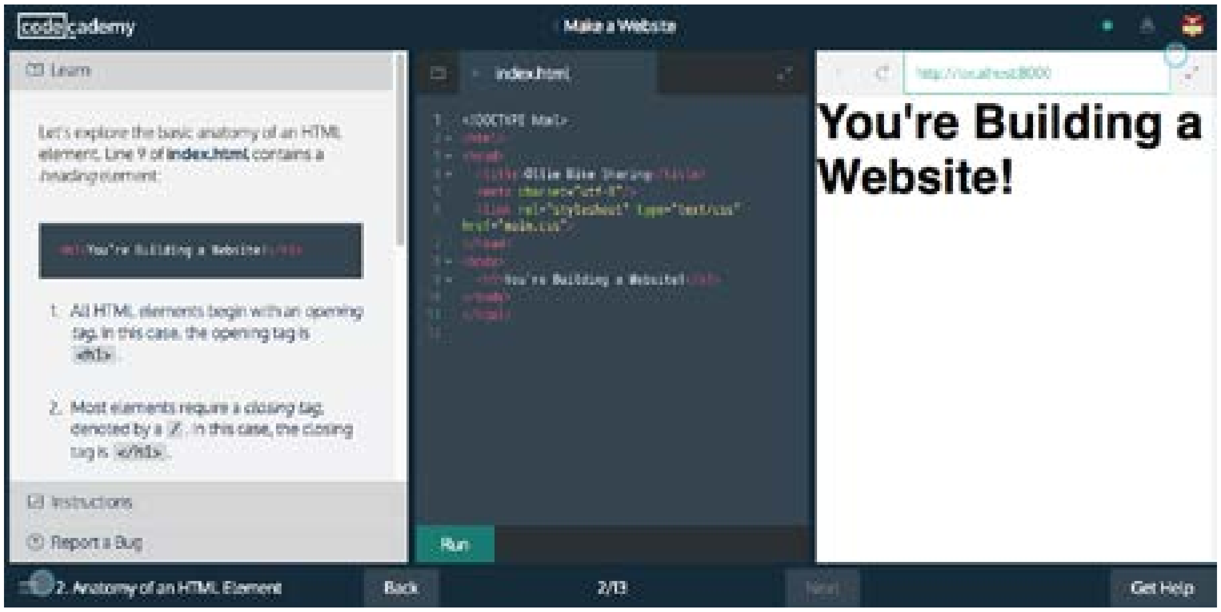
GENERAL USE ACROSS THE CURRICULUM

While Codecademy offers a range of courses, the 4-hour web design unit is one of the more versatile options across disciplines. Websites can easily fit in most classrooms and course assignments. Enabling students to create their own website, whether it be for an online portfolio for a semester’s worth of work, or to serve as an interactive “report,” students have control over the design and learn a new skill as well.

For example, "Animate Your Name" can be used as an icebreaker activity, providing a unique but challenging way to introduce themselves, their group team names, or their favorite subject.

Codecademy also offers a robust Teaching Resources section of their site to support implementation in the classroom. Resources include a level progression map, schemes of work, assessment levels and plenary quizzes.

Make a Website
Build four webpages from scratch while learning web development fundamentals
JavaScript
Python
Ruby
jQuery
BM
Op
Da
DT



Let's explore the basic anatomy of an HTML element. Line 9 of **index.html** contains a *heading element*:

```
<h1>You're Building a Website!</h1>
```

1. All HTML elements begin with an opening tag. In this case, the opening tag is `<h1>`.

2. Most elements require a closing tag, denoted by a `/`. In this case, the closing tag is `</h1>`.

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4     <title>Ollie Bike Sharing</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" type="text/css"
    href="main.css">
7   </head>
8   <body>
9     <h1>You're Building a Website!</h1>
10  </body>
11  </html>
12
```

You're Building a Website!

2. Anatomy of an HTML Element

3

TALEBLAZER >

TaleBlazer (augmented reality) invites users to design and play location-based mobile games through their augmented reality (AR) software platform. Augmented reality, like virtual reality, seeks to offer "real" simulations to convey information and experiences. TaleBlazer sees "location-based" games as a way to "situate games in the real world" and "engage people in experiences that combine real landscapes and other aspects of the physical environment with additional digital information supplied to them by smartphones." The online platform features a visual programming interface (like Scratch), while also providing a free cloud service to easily transfer projects from a computer to their smartphone. TaleBlazer also has a "Taleblazer for Teachers" section that outlines uses in educational settings.

OPPORTUNITIES FOR TALEBLAZER TO SUPPORT STUDENT-CENTERED STRATEGIES FOR LEARNING

TaleBlazer **invites and encourages anytime, anywhere learning**—projects and games can be accessed via mobile devices and are meant to incorporate the physical environment into their design. Additionally, once games and projects are downloaded, no internet is required to play.

Platform **enables educators to provide "just in time" feedback to students**, and supports peer-to-peer collaboration. Individuals can work at their own pace while still benefiting from collaboration and peer-mentoring when needed.

Students are **motivated to interact and learn about their surroundings**. Users can grow a connection and curiosity for their communities, as the tool's augmented reality features work with real GPS data. Students can base games off of familiar sites, or locations they're most curious about.

STRATEGIES FOR IMPLEMENTATION

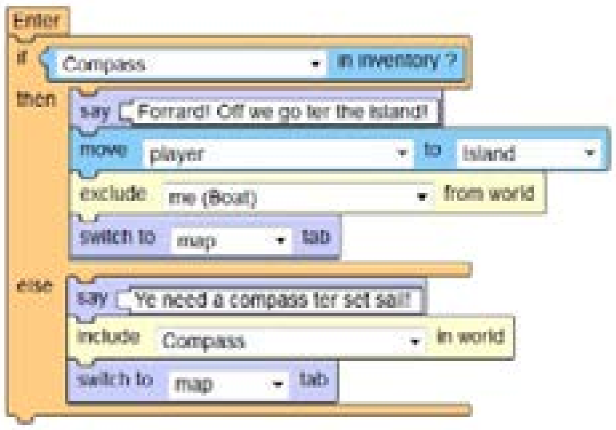
FIELD TRIPS

Students can design games that incorporate museum artifacts, or download and play with existing games made by (participating) museum staff. Museums and historical sites are designing games that promote exploration of their exhibits using devices. The Columbus Zoo and Aquarium, for example, created a game that allowed students to learn about climate change during live animal exhibits. Check out the game in TaleBlazer Place, within the mobile app.

In Brooklyn, students designed a game ("Helen's Treasures") where participants are asked to locate specific items within the Brooklyn Museum in order to learn about the game's protagonist, Helen (based off of a marble bust).

SCIENCE CLASS

Students can create games that encourage peers to take field notes and measurements of nearby parks or trails. In Lexington, Massachusetts, students used TaleBlazer (as part of their work with the virtual reality based curriculum EcoMUVE), to explore a local pond, tracking location-based data and uploading photos and texts relevant to their studies.



```
Enter
if    Compass  in inventory ?
then  say "Forrard! Off we go ter the island!"
      move player to Island
      exclude me (Boat) from world
      switch to map tab
else  say "Ye need a compass ter set sail!"
      include Compass in world
      switch to map tab
```

4

APP INVENTOR >

App Inventor (application development) is a “beginner” application development tool for users of all skill levels. The platform features a drag and drop, visual programming language (like **Scratch**) and allows users to build apps in a web browser. Students can even test their apps on their mobile device.

OPPORTUNITIES FOR APP INVENTOR TO SUPPORT STUDENT-CENTERED STRATEGIES FOR LEARNING

Instructional materials are **presented in multiple formats** that acknowledge a variety of learning styles—materials range from videos, tutorials, or “concept cards” to support a personalized learning environment. The starter tutorials are available in both video or text format, so all learners can get started in a way that excites and engages them.

Introductory resources are designed to provide students with an “early win” moment early in the learning process, helping students **develop agency and build confidence** over time.

App development invites **critical thinking** about a device they likely interact with for hours a day—inviting conversation on how their favorite apps were developed, the possibilities of design, and the variety of roles within the app industry.

STRATEGIES FOR IMPLEMENTATION

SPANISH CLASS

Students can create an app that can function as virtual flashcards. Following the QuizMe tutorial, students can create quizzes that reinforce vocabulary via images, sounds or text. In an introductory Spanish class, a student might design visual, randomized flashcards to study—for example, they could see a photo of an apple, and be prompted to type the word in Spanish, *manzana*.

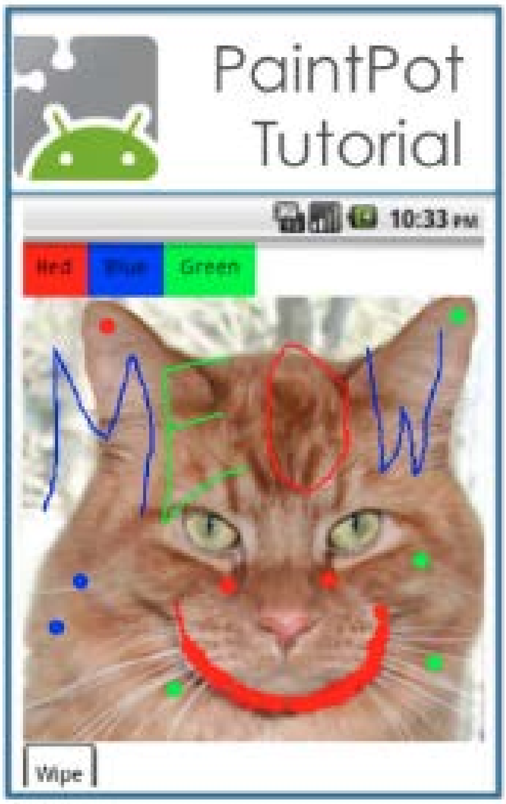
HISTORY CLASS

Use App Inventor’s Map It activity and have students learn how to record a list of addresses and view them on Google Maps. Students can input locations of historical events, where people were born, or features of their city from 100 years ago.

OUT OF SCHOOL

Apps are often used for anytime, anywhere learning. For example, Boston's Student Advisory Council launched the Boston Student Rights website and app designed to help their peers and families understand their student rights and school policies to combat discriminatory practice in schools. While not made with App Inventor, the tool provides similar feature support to help students create comparable platforms for their peers.

Display hidden components in Viewer
5:09 PM
Screen1
Eat all the aphids as fast as you can!
Up
Left
Right
Down



5

CODE.ORG >

Code.org (computer science fundamentals) is a non-profit “dedicated to expanding access to computer science, and increasing participation by women and underrepresented students of color.” Code.org offers instructional materials both for learners and for educators. For educators specifically, the site offers professional development resources at the elementary, middle school, and high school level. Code.org also offers advocacy tools to empower users to promote and value coding, regardless of expertise.

OPPORTUNITIES FOR CODE.ORG TO SUPPORT STUDENT-CENTERED STRATEGIES FOR LEARNING

The classroom resources incorporate **civic and career readiness**. School-time activities introduce computer science careers to students, in addition to the ethical and societal factors that come with digital responsibility and accessibility (or lack of accessibility). Due to Code.org’s mission, educators can introduce social justice as a key part of becoming a responsible 21st century citizen.

The coursework is designed with pause points and room for error, inviting students to **self-assess** their own work, and to **reflect** on personal areas of growth throughout each exercise.

Resources online emphasize coding as a means for **youth empowerment**—the site offers snapshots of students that use coding to amplify their voices and make an impact in their communities.

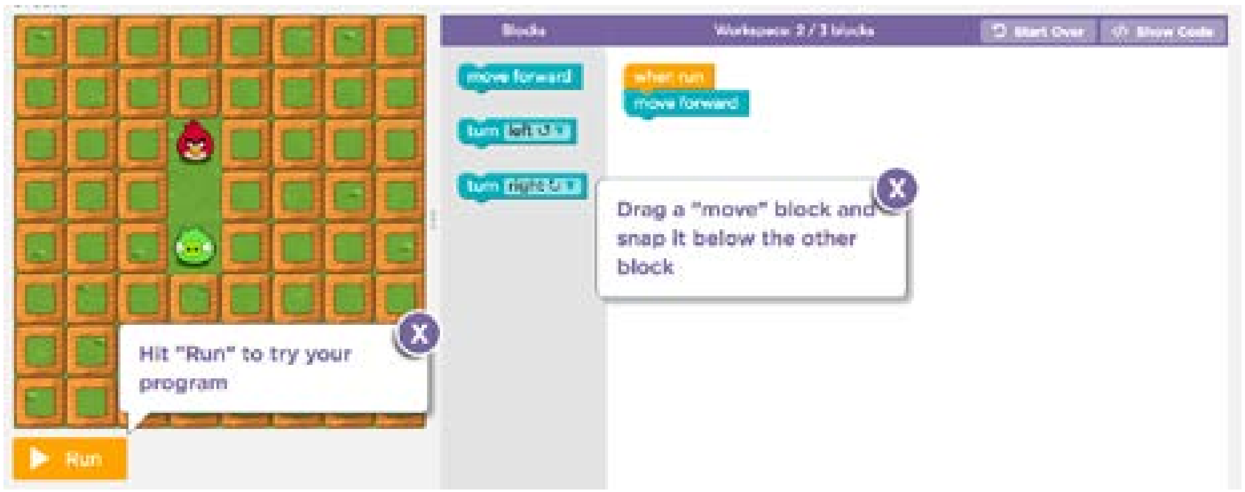
STRATEGIES FOR IMPLEMENTATION

HEALTH CLASS

Using Code.org’s Introduction to Data Unit, students can track information relating to their health. The unit allows students to reflect on where data comes from, what formats it comes in, and what kinds of other information can be extracted using computational tools. Following the lesson plan, students could design a data tracker that tracks the amount of water or soda consumed by their peers in a day, and reflect on the data overtime.

MATH CLASS

Incorporating data visualizations can make statistics playful. Code.org's Good and Bad Data Visualizations lesson plan introduces infographics and the best practices for creating them. They will also learn about how graphs can mislead or misrepresent information as part of this lesson. In a statistics class, students could select a dataset from a topic of interest, and model their own "bad" data visuals. Later, students can be invited to critique and challenge each visualization.



Drag a "move" block and snap it below the other block

Hit "Run" to try your program

6

ADDITIONAL RESOURCES

Code.org : Recommended Resources
https://code.org/educate/curriculum/3rd-party

Teaching Kids to Code : EdSurge Guide
https://www.edsurge.com/research/guides/teaching-kids-to-code

CSNYC : Resources
http://www.csnyc.org/resources

Open for All : 6 Computer Science Projects from MIT Media Lab that you can try today
http://www.makeuseof.com/tag/open-for-all-6-computer-science-projects-from-mit-media-lab-that-you-can-try-today/

Computer Science for Rhode Island
http://www.cs4ri.org/

Exploring Computer Science
http://www.exploringcs.org/

Code.org : Vermont facts and policy
https://code.org/advocacy/state-facts/VT.pdf
http://www.kidsvt.com/vermont/trailblazing-teachers-bring-computer-programming-into-the-classroom/Content?oid=2171969

New Hampshire Information and Communications Technology Toolkit
http://www.nheon.org/ictliteracy/ictstandards.html

Massachusetts Digital Literacy and Computer Science Standards Panel
http://www.doe.mass.edu/stem/standards.html

Exploring CS
http://www.exploringcs.org/

EcoMUVE
http://ecolearn.gse.harvard.edu/ecoMUVE/overview.php

We want to include more resources here! Please contact us if you'd like to suggest a CS resources/tools/platforms for us to include.

CONTACT
Jackie Gonzalez, Jobs for the Future
jgonzalez@jff.org
617.728.4446