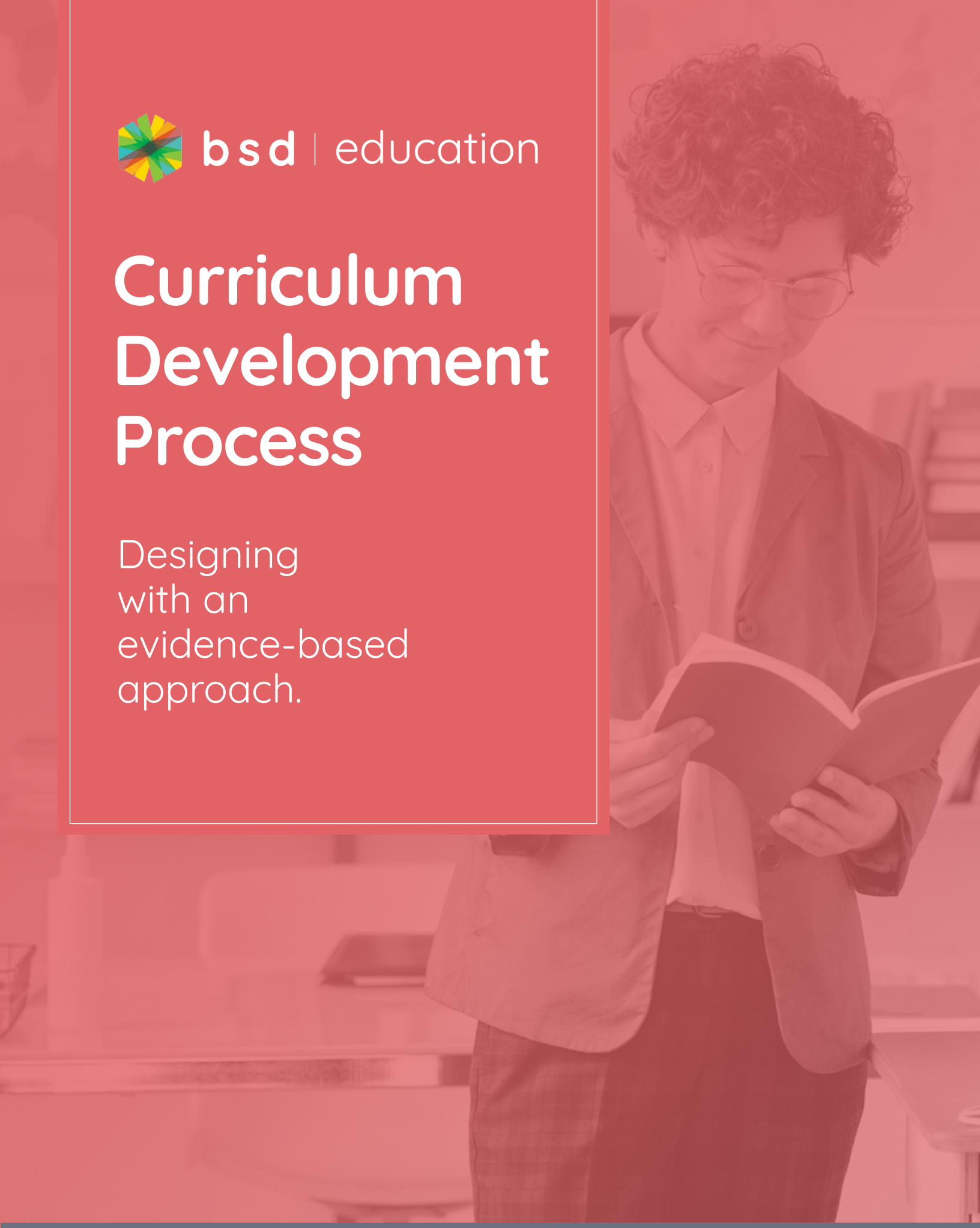
bsd | education

Curriculum Development Process

Designing with an evidence-based approach.



Our pedagogical foundation and curriculum development process is built from studying educational research and from decades of practical experience in classrooms all over the world. Our team includes technology experts, educators, and education researchers to co-create technology solutions for schools that are rooted in evidence-based pedagogical practices.

Meeting the Challenges of the Digital Learning Gap

In an effort to address the Digital Learning Gap, BSD Education has designed a comprehensive technology education product that can be used by teachers of any skill level or background. Cator (2019) describes the Digital Learning Gap by saying "many teachers and students of all ages do not yet have the knowledge and skills to use technology in personally productive and powerful ways."

At BSD Education, we take "Powerful Ideas" very seriously, by leaning on the pedagogical research in the field of constructionism and learning sciences. Papert (1984-2020) describes Powerful Ideas as "seeds of cultural change, (that) can help people form new relationships with knowledge." We echo this description of Powerful Ideas and intentionally design curriculum that exposes these Powerful Ideas, allowing students to form their own personal relationships with technology. Our approach to addressing the Digital Learning Gap is to provide digital skills curriculum that uses professional tools, is scaffolded for ease of learning, is rooted in personally relevant topics, allows for personalization and customization, and is supported through professional development and ongoing support.

To help us address the Digital Learning Gap, we are driven by 4 main theories of action, presented below.

1. BSD Pedagogical Foundation

At BSD Education, we design our projects, lessons, and technology solutions from a strong pedagogical foundation, believing that: **1)** teachers need high quality support for digital learning, **2)** learning by doing is the most practical approach in education, and **3)** a mixture

of scaffolded learning and open-ended projects provide students with the support they need to be successful.

1) **Teachers need high quality support for digital learning**
   "*The most effective teachers have deep knowledge of the subjects they teach, and when teachers' knowledge falls below a certain level it is a significant impediment to students' learning.*" (Coe, et al. 2014) With so many advances in digital technology, it is often difficult for teachers to stay abreast of the best practices in digital skills. At BSD Education, we provide continual professional development, and individualized coaching and support to teachers to help them build the confidence and skills needed.

2) **Learning by doing is the most practical way to approach education**
   Psychologists and cognitive scientists have been preaching the benefits of hands-on learning for decades. From Jean Piaget who said "*to understand is to invent*" and Seymour Papert who said "*knowledge is built by the learner, not supplied by the teacher*", we at BSD Education wholeheartedly agree and design our learning materials in the form of projects that students engage with and experiment with.

3) **A mixture of scaffolded learning and open-ended projects provide students with the support they need to be successful**
   In a scaffolded approach, learners start with small bits of information and slowly build and add on to that knowledge in manageable loads. We provide worked-examples and guided instructions for students that slowly build up to a level where they can explore, tinker and play in an unobstructed environment called Sandbox mode.

2. The BSD Curriculum

BSD Education Learning Cycle

All of our lessons, projects and curriculum follow a learning cycle that is designed to match the pedagogical framework that we have described above, called **Explore, Learn, Create**.

Explore

In this phase, we introduce topics and subjects in an engaging way to help students elicit previous knowledge, and also to discuss and explore new ideas and concepts.

Learn

In this phase, we go into more detail and teach digital skills and technology concepts that are needed to complete the Create phase successfully. In this phase, key concepts are introduced, but to fully grasp the knowledge, students must be immersed in the **doing of learning** that comes in the next phase.

Create

In the final phase of our curriculum, students will be working on a BSD code project or another creative endeavour that allows them to synthesize previously learned concepts by building digital artifacts.

Assessment

Traditional grading practices and formal assessment in schools tend to rely on punitive measures, leading to possible poor self-image and a variety of other unintended negative effects. To support non-punitive measures, we approach assessment in two different ways. **1)** we assess the technical learning of digital skills on our platform with multi-attempt quizzes at the end of projects **2)** Teachers will provide written feedback on performance, creativity, and overall effort using rubrics. Both of these approaches focus on the intended nature of assessment, to provide feedback for improvement, and celebrate success.

1) **Quizzes**

   When assessing technical concepts such as how well students have gained skills in HTML, CSS, and JavaScript, teachers trust us to provide the appropriate level of assessment. Our platform provides multiple choice questions at the end of each project to assess technical ability.

2) **Rubrics**

   When it comes to assessing the students individual performance, creativity and effort, we trust teachers to provide written feedback through the use of a rubric. This type of assessment provides an opportunity for teachers to have a real conversation with students about their progress.

3. Best Practices in Curriculum Design at BSD

Along with our foundational pedagogy and learning cycle, we also follow evidence-based practices for curriculum design.

Teaching through real-world connections

Our lessons are designed to create authentic learning experiences where students can experience connections to real world issues and problems, exposing them to key global issues and themes.

Teaching Social Emotional Learning

Our curriculum designers use a social emotional learning lens when designing lessons that connect to key issues in social justice, critical race pedagogy, and culturally relevant practices.

Best Practices in Computer Science Education

Our team consults best practices in computer science education in our approach to designing and developing coding projects that align to rigorous standards such as CSTA and ISTE.

User Experience and Interface Design

We design our platform so that students and teachers can easily operate and navigate through it in an intuitive and logical manner. Continual improvement on the design of our product relies on feedback from customers and prioritisation for new developments and features, using the RISE methodology for continuous improvement.

4. Digital Skills for the Workforce of Tomorrow

To prepare students for undefined futures where artificial intelligence, augmented reality and data privacy are all emerging topics with tremendous impacts on society, we embed 4 approaches into our curriculum that have been identified as future proof and fundamental: **1)** Computational Thinking, **2)** Design Thinking, **3)** coding/programming, and **4)** Digital Citizenship.

1) Computational Thinking

A method that helps students to break down complex problems into smaller manageable parts (decomposition) and develop solutions for the individual components by identifying patterns and similarities between all the parts (pattern recognition), solving each problem on its own (abstraction) and then developing an unified solution for the entire problem that can be reused multiple times (algorithm).

2) Design Thinking

A method for innovating, based on the needs of a particular user or group of users. Students as designers first seek to understand the problems that a particular group faces (empathize). Once the problem has been determined, (define) then student designers will make a planned solution (ideate) that can be rapidly made into a minimal viable product (prototype). To be sure the design functions as intended, the users should be able to test the design (test). Student designers will often make and test several prototypes and refine parts of it before a final solution is ready to publish or scale.

3) Coding and Programing

At BSD Education, we teach HTML, CSS, and JavaScript, which are used in the real world to produce the content that you interact with everyday online through websites and mobile apps. We see coding and programming as another form of digital literacy that prepares students for interacting in the digital world as creators instead of just users.

- **HTML** provides the structure of web pages, like headings and sections.
- **CSS** provides the style and look of the web page, like colors and fonts.
- **JavaScript** adds functionality, like responding to clicks or playing games

4) Digital Citizenship

To prepare students for the workplace of tomorrow, a comprehensive understanding of digital citizenship is vital. All of our curriculum is infused with learning opportunities in digital citizenship so that students see how citizenship is a natural part of interacting online, and in virtual communities instead of an isolated practice.

Using a logic model to describe our evidence-based approach

A logic model is a tool that researchers and practitioners use to align the objectives of a product or service to measured outputs that are proven by peer-reviewed research. The research that best supports the **Outcomes** in our logic model are divided into 3 categories: Teacher Ability, Technical Skills, and Project-Based Curriculum.

Teacher Ability

Learning to program and adopting new digital skills can be a difficult process for both teachers and students. Our product is designed to be an effective tool for teachers, which means that teachers will be provided with an opportunity to expand their own digital skills. Ying-Shao et al. (2007) have found that when teachers have ongoing support and trust in the product, their ability and effectiveness to teach technical skills increases. At BSD, we help to build trust in the product by providing onboarding support, professional development, and continued support during all stages of use.

Teachers all over the world have found our approach to support and professional development to be very effective. Kathy Smith, a teacher in the US says, “the professional development was phenomenal, they included hands-on parts so that we could actually get the experience of doing it just as a student would do it.”

Technical Skills

At BSD Education, we teach students how to build websites, make web games, analyze data, tinker with AI algorithms, and design virtual reality worlds. These projects require learning real programming languages like HTML, CSS, and JavaScript. While we don't expect every student to become professional programmers, we do see value in the benefits of learning to code. Schrer and Siddiq (2018) have found that learning to code has a strong correlation to transfer-skills and has far-reaching cognitive benefits. Transfer-skills are skills that are learned through programming, but help in other areas, such as creative thinking, mathematical skills, and metacognition. Creative thinking and metacognitive ability are vital skills for any field of work.

We take learning to code seriously and provide an easy-to-navigate structure for new learners. Rum and Ismail (2017) have found that metacognitive reflection improves performance for novice programmers. In order to facilitate metacognitive reflection and support novice learners, we have embedded opportunities throughout our projects to ask questions in quizzes and re-introduce introductory topics to help with retention, in a process called metacognitive scaffolding. This process allows learners to slowly learn new skills and self-reflect on the process, which is important for learning to code.

Project-Based

To best serve our students, BSD Education curriculum is designed around projects like "design a multi-page website for a community organization" or "make an interactive digital story for a unique target audience". These projects weave together personal interest, cultural relevance, and acquisition of new coding skills. Hsu and Hwang (2017) have found that when learning to code, structured projects are better than open-ended activities. Our structured projects introduce students to the topics of the projects through lessons, videos and other interactive activities. When students start the coding aspect of the project, we apply a scaffolded approach to teach coding in a way that breaks down complex content into steps that build in difficulty and sequence. Belland et al (2013) have found that scaffolding the knowledge of new content can increase student motivation and engagement. In a 2021 survey of students learning from BSD Education courses, 99% found our courses to be engaging.

A project-based approach has other benefits, Drake and Long (2009) have found that students show an increase in problem-solving ability when learning through projects. The nature of a project-based approach means that students are prompted with a challenge, problem, or driving question that requires critical thinking and problem-solving to complete. In every BSD Education project, students will have the opportunity to flex and increase their problem-solving skills.

In summary, at BSD Education, we know that in order to provide high quality products to teachers and students, it must start with a strong foundation that comes from research, proven methods, and from best practices within the fields of education and the workforce.

References for the Logic Model

Want to read the full papers that we referenced in the Logic model?

Belland, B., Kim, C. & Hannafin, M. (2013) A Framework for Designing Scaffolds That Improve Motivation and Cognition, Educational Psychologist, 48:4, 243-270, DOI: 10.1080/00461520.2013.838920

Drake, K., & Long, D. (2009). Rebecca's in the Dark: A Comparative Study of Problem-Based Learning and Direct Instruction/Experiential Learning in Two 4th-Grade Classrooms. Journal of Elementary Science Education, 21(1), 1-16. Retrieved April 30, 2021, from http://www.jstor.org/stable/43155840

Hsu, T., & Hwang, G. (2017). Effects of a Structured Resource-based Web Issue-Quest Approach on Students' Learning Performances in Computer Programming Courses. Journal of Educational Technology & Society, 20(3), 82-94. Retrieved April 30, 2021, from http://www.jstor.org/stable/26196121

Rum, S., & Ismail, M. (2017). Metacognitive Support Accelerates Computer Assisted Learning for Novice Programmers. Journal of Educational Technology & Society, 20(3), 170-181. Retrieved April 30, 2021, from http://www.jstor.org/stable/26196128

Scherer, Ronny & Siddiq, Fazilat & Viveros, Bárbara. (2018). The Cognitive Benefits of Learning Computer Programming: A Meta-Analysis of Transfer Effects. Journal of Educational Psychology. 111. 10.1037/edu0000314.

Ying-Shao Hsu, Hsin-Kai Wu, & Fu-Kwun Hwang. (2007). Factors Influencing Junior High School Teachers' Computer-Based Instructional Practices Regarding Their Instructional Evolution Stages. Journal of Educational Technology & Society, 10(4), 118-130. Retrieved April 30, 2021, from http://www.jstor.org/stable/jeductechsoci.10.4.118