JAVASCRIPT & EDUCATION

The idea of teaching students emerging technology has always been a hot button topic, among educators.  Depending on the philosophy an educator takes, Marc Prensky could be compared to a modern day Moses, with his work on education reform.   Prensky"s most famous work, “Digital Natives, Digital Immigrants” postulates that educational material needs to evolve with the emerging technologies.  (Prensky, 2001)If you take his general thesis that was created in 2001, that the digital native needs proper localized stimulation and generate a continued forward thinking hypothesis in a direction of continued technological growth, the concept that emerges is the need to learn how to program in 2014.  One such individual is Brian MacDonald, he’s even taken it a step further, and in his current opinion Javascript is the ideal language to include into a forward thinking academic curriculum. (Macdonald, 2013)

“An instructor cannot teach a student how to think. He or she can however, teach a student how to organize thought, and teach them tools and methods for defining and simplifying solutions and problems. If a student does not have this ability, then teaching them a language will be a wash. Spend some time on this first before any code hits the screen. “(Naillon, 2011)

Edward Naillon gave any educator the hardest issue with teaching Javascript or programming in general to high school students. It is impossible to teach a programming language without the ability to constructively critically think. (Naillon, 2011) This is the largest issue that many educators cannot overcome and think that teaching programming to students is a beneficial tool.  Honestly this is the most important reason to teach students programming in a well organized detailed curriculum. You can take MacDonald’s reasoning for Javascript as introductory language to programming as a solution to this problem, because it has a forgiving syntax, instantly gratifying results, and lightweight IDE. (Macdonald, 2013)

Taking MacDonald’s list of why Javascript is ideal, first being the forgiving syntax.  This allows the educator and student to see results without a 100% mastery of the debugging process, and this allows students to focus more on the problem solving tasks and exercise of programming. (Macdonald, 2013) This allows the student, a personalized approach to complex thinking, professional resolve and confidence not just in but also outside of a traditional programming/academic environment.   This will also congruently grow and expand the student’s academic growth if properly coupled with a rigorous mathematic education.  Taking MacDonald’s second reason for Javascript in the classroom, the student will have immediate hands on results.  Anyone who’s experienced the joy of programming Javascript can attest it’s a great experience, and interest generating to see code display out be it console.log(“Hello World!”); or a multi-lined  behemoth that calculates the probability that Tony Gwynn Jr. will have as good of a career as his father.  (Macdonald, 2013)  Finally the lightweight IDE, makes it radically simple to teach anyone, as it doesn’t get in the way of teaching and experiencing the language. (Macdonald, 2013)

Everyone knows that education budgets are not what they used to be, but also our expectation of students are naturally growing and expanding with every generation.  Prensky forecasted the growth of education nearly fifteen years ago, and we’re at the next evolutionary step of his classic Digital Immigrants vs Natives argument. (Prensky, 2001) So we must continue to grow or fall back even further in the education process, Javascript is the perfect gateway programming language to teach students, because MacDonald doesn’t have a farm anymore, he has himself a laptop instead. (Macdonald, 2013) So it’s time to get students to put their thinking hats back on and lets get them to think constructively with a new way to think instead of the classic mathematics approach, and I promise we’ll see new horizons on our sails.

References

Macdonald, B. (2013, November). *Which Language Should You Learn First?*Retrieved May 2014, from http://programming.oreilly.com/2013/11/which-language-should-you-learn-first.html

Naillon, E. (2011). *Teaching Computer Programming to High School Students*. Retrieved May 2014, from http://oroville.wednet.edu/~Ed/resources/programming/Teaching\_Computer\_Programming\_To\_High\_School\_Students.pdf

Prensky, M. (2001, October). *Digital Natives, Digital Immigrant*. Retrieved May 2014, from http://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf