

Re: Programming languages for the very young

Source: <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2004-01/0723.html>

From: Anton van Straaten (anton_at_appsolutions.com)

Date: 01/14/04

Date: Wed, 14 Jan 2004 20:51:20 GMT

Joe Marshall wrote:

> "Anton van Straaten" <anton@appsolutions.com> writes:
>> 3. *Balancing the above point, it has those sloppy scripting language
>> features that tend to make life easier for beginners who haven't yet
learned
>> to be detail-oriented
>
> I hate those. I think there is an advantage to understanding that
> certain objects have certain behaviors --- you simply cannot add a
> number and a sentence.*

I think some of those kinds of features can be helpful, others can be the opposite. Still, we're talking about teaching young children. We don't expect that a kid's model of a house to conform to the local municipal building codes. They use cardboard instead of drywall, etc. I think some of the things that we tend to assume are essential might need to be let go of.

>> 5. *It has a prototype-based object system.
>
> Minus.*

Of course, you don't have to use it. If you want objects at all, you can build them out of closures, just like in Scheme, or mix the approaches, using the prototype system to add public methods to an object but using lexical variables to achieve private state. You can also simulate classes, if you want them.

Again, I think the friendliness of being able to simply say "obj.foo = ..." and have it work, without requiring prior setup such as creating a class, may still be a positive for kids, regardless of what poor software engineering it might be.

>> 6. *It has a good track record with neophytes in the form of web designers,
>
> I don't want my kids to be code monkeys, I want them to *learn*.*

comp.lang.lisp: Re: Programming languages for the very young

My point was just Javascript has a demonstrated low barrier to entry – that people with minimal coding aptitude have been able to use it, therefore kids with no prior experience should also be able to.

- > > *The biggest disadvantage of Javascript is probably its syntax,*
- >
- > *Yes. I would prefer parentheses.*

Well, the next step would be to write a small interpreter hosted in the web browser. You could quite easily build a teaching-oriented Scheme-like subset on top of Javascript, especially if you didn't care about tail recursion & continuations and so on. Someone posted here about their Scheme-in-Javascript implementation a while ago (which might be more than you need, although I haven't looked at it). You'd still get the integration with HTML and the "deployability" benefits, but the nature of the language would be under your control.

Anton