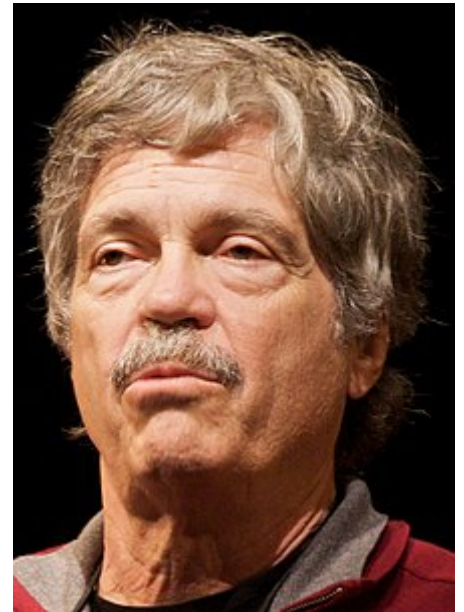


Alan Kay

Alan Curtis Kay (born 17 May 1940) is an American computer scientist best known for his pioneering work on object-oriented programming and windowing graphical user interface design.



Alan Kay (2008)

Contents

Quotes

1970s

1980s

1990s

2000s

A Conversation with Alan Kay, 2004–05

2010s

External links

Quotes

1970s

- **The best way to predict the future is to invent it.**
 - Alan Kay (1971) at a 1971 meeting of PARC (<http://quoteinvestigator.com/2012/09/27/invent-the-future/>)
 - Similar remarks are attributed to Peter Drucker and Dandridge M. Cole.
 - Cf. Dennis Gabor, *Inventing the Future* (1963): "The future cannot be predicted, but futures can be invented."
 - Nigel Calder reviewed Gabor's book and wrote, "we cannot predict the future, but we can invent it..."
- [Computing] is just a fabulous place for that, because it's a place where you don't have to be a Ph.D. or anything else. It's a place where you can still be an artisan. People are willing to pay you if you're any good at all, and you have plenty of time for screwing around.
 - Alan Kay (1972) in 1972 Rolling Stone article (http://www.wheels.org/spacewar/stone/rolling_stone.html)

1980s

- People who are really serious about software should make their own hardware.
 - talk at Creative Think seminar, 20 July 1982 (https://www.folklore.org/StoryView.py?project=Macintosh&story=Creative_Think.txt)

- **A change in perspective is worth 80 IQ points.**
 - Perspective is worth 80 IQ points.
 - Point of view is worth 80 IQ points
 - Talk at Creative Think seminar, 20 July 1982 (https://www.folklore.org/StoryView.py?project=Macintosh&story=Creative_Think.txt)
- Technology is anything that wasn't around when you were born.
 - *Hong Kong press conference in the late 1980s*
- **The future is not laid out on a track. It is something that we can decide, and to the extent that we do not violate any known laws of the universe, we can probably make it work the way that we want to.**
 - 1984 in Alan Kay's paper *Inventing the Future* which appears in *The AI Business: The Commercial Uses of Artificial Intelligence*, edited by Patrick Henry Winston and Karen Prendergast. As quoted by Eugene Wallingford in a post entitled *ALAN KAY'S TALKS AT OOPSLA* (http://www.cs.uni.edu/~wallingf/blog/archives/monthly/2004-11.html#e2004-11-06T21_03_42.htm) on November 06, 2004 9:03 PM at the website of the Computer Science section of the University of Northern Iowa.

1990s

- I don't know how many of you have ever met Dijkstra, but you probably know that arrogance in computer science is measured in nano-Dijkstras.
 - The Computer Revolution hasn't happened yet — 1997 OOPSLA Keynote (<https://www.youtube.com/watch?v=oKg1hTOQXoY>)
- Actually I made up the term "object-oriented", and I can tell you I did not have C++ in mind.
 - The Computer Revolution hasn't happend yet — 1997 OOPSLA Keynote (<https://www.youtube.com/watch?v=oKg1hTOQXoY>)
 - *Alternative*: I invented the term Object-Oriented, and I can tell you I did not have C++ in mind.
 - Attributed to Alan Kay in: Peter Seibel (2005) *Practical Common Lisp*. p.189

2000s

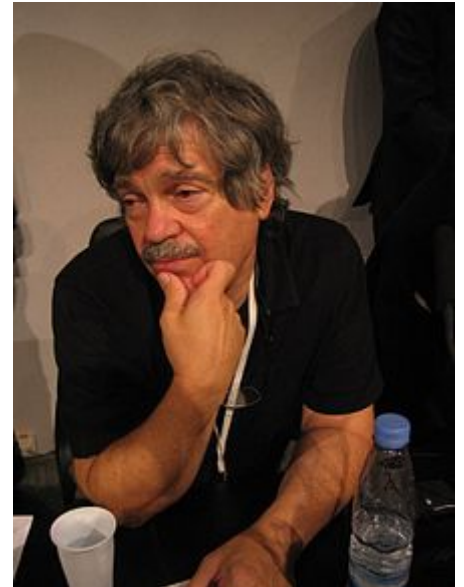
- ... greatest single programming language ever designed.
(About the Lisp programming language.)
 - 2003. Daddy, Are We There Yet? A Discussion with Alan Kay (http://www.openp2p.com/pub/a/p2p/2003/04/03/alan_kay.html)
- I finally understood that the half page of code on the bottom of page 13 of the Lisp 1.5 manual was **Lisp** in itself. These were "Maxwell's Equations of Software!"
 - ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 (<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=273&page=4>)



If you don't fail at least 90 percent of the time, you're not aiming high enough.

A Conversation with Alan Kay, 2004–05

- Most software today is very much like an Egyptian pyramid with millions of bricks piled on top of each other, with no structural integrity, but just done by brute force and thousands of slaves.
 - ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 (<http://queue.acm.org/detail.cfm?id=1039523>)
- **Perl** is another example of filling a tiny, short-term need, and then being a real problem in the longer term. Basically, a lot of the problems that computing has had in the last 25 years comes from systems where the designers were trying to fix some short-term thing and didn't think about whether the idea would scale if it were adopted. There should be a half-life on software so old software just melts away over 10 or 15 years.
 - ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 (<http://queue.acm.org/detail.cfm?id=1039523>)
- **Basic** would never have surfaced because there was always a language better than Basic for that purpose. That language was Joss, which predated Basic and was beautiful. But Basic happened to be on a GE timesharing system that was done by Dartmouth, and when GE decided to franchise that, it started spreading Basic around just because it was there, not because it had any intrinsic merits whatsoever.
 - ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 (<http://queue.acm.org/detail.cfm?id=1039523>)
- Computing spread out much, much faster than educating unsophisticated people can happen. In the last 25 years or so, we actually got something like a pop culture, similar to what happened when television came on the scene and some of its inventors thought it would be a way of getting Shakespeare to the masses. But they forgot that you have to be more sophisticated and have more perspective to understand Shakespeare. What television was able to do was to capture people as they were. So I think the lack of a real computer science today, and the lack of real software engineering today, is partly due to this pop culture.
 - ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 (<http://queue.acm.org/detail.cfm?id=1039523>)
- Sun Microsystems had the right people to make Java into a first-class language, and I believe it was the Sun marketing people who rushed the thing out before it should have gotten out.
 - ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 (<http://queue.acm.org/detail.cfm?id=1039523>)
- If the pros at Sun had had a chance to fix Java, the world would be a much more pleasant place. This is not secret knowledge. It's just secret to this pop culture.



By the time I got to school, I had already read a couple hundred books. I knew in the first grade that they were lying to me because I had already been exposed to other points of view. School is basically about one point of view — the one the teacher has or the textbooks have.



Simple things should be simple, complex things should be possible.

- [ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 \(http://queue.acm.org/detail.cfm?id=1039523\)](http://queue.acm.org/detail.cfm?id=1039523)
- I fear —as far as I can tell— that most undergraduate degrees in computer science these days are basically Java vocational training. I've heard complaints from even mighty Stanford University with its illustrious faculty that basically the undergraduate computer science program is little more than Java certification.
 - [ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 \(http://queue.acm.org/detail.cfm?id=1039523\)](http://queue.acm.org/detail.cfm?id=1039523)
- **Most creativity is a transition from one context into another where things are more surprising.** There's an element of surprise, and especially in science, there is often laughter that goes along with the "Aha." Art also has this element. **Our job is to remind us that there are more contexts than the one that we're in — the one that we think is reality.**
 - [ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 \(http://queue.acm.org/detail.cfm?id=1039523\)](http://queue.acm.org/detail.cfm?id=1039523)
- I hired finishers because I'm a good starter and a poor finisher.
 - [ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 \(http://queue.acm.org/detail.cfm?id=1039523\)](http://queue.acm.org/detail.cfm?id=1039523)
- The flip side of the coin was that even good programmers and language designers tended to do terrible extensions when they were in the heat of programming, because design is something that is best done slowly and carefully.
 - [ACM Queue A Conversation with Alan Kay Vol. 2, No. 9 - Dec/Jan 2004-2005 \(http://queue.acm.org/detail.cfm?id=1039523\)](http://queue.acm.org/detail.cfm?id=1039523)

2010s

- However, I am no big fan of Smalltalk either, even though it compares very favourably with most programming systems today (I don't like any of them, and I don't think any of them are suitable for the real programming problems of today, whether for systems or for end-users).
 - [2010 for Computerworld Australia \(http://www.computerworld.com.au/article/352182/z_programming_languages_smalltalk-80/\)](http://www.computerworld.com.au/article/352182/z_programming_languages_smalltalk-80/)
- Possibly the only real object-oriented system in working order. (About Internet)
 - [2010 for Computerworld Australia \(http://www.computerworld.com.au/article/352182/z_programming_languages_smalltalk-80/\)](http://www.computerworld.com.au/article/352182/z_programming_languages_smalltalk-80/)
- The Internet was done so well that most people think of it as a natural resource like the Pacific Ocean, rather than something that was man-made. When was the last time a technology with a scale like that was so error-free? The Web, in comparison, is a joke. The Web was done by amateurs.
 - [2012 Dr. Dobb's Interview with Alan Kay \(http://www.drdobbs.com/architecture-and-design/interview-with-alan-kay/240003442\)](http://www.drdobbs.com/architecture-and-design/interview-with-alan-kay/240003442)
- Object-oriented [programming] never made it outside of Xerox PARC; only the term did.
 - [Alan Kay - Rethinking Design, Risk, and Software \(https://www.youtube.com/watch?v=Qbol1WJUIM\)](https://www.youtube.com/watch?v=Qbol1WJUIM)

External links

Retrieved from "https://en.wikiquote.org/w/index.php?title=Alan_Kay&oldid=2815093"

This page was last edited on 25 June 2020, at 17:27.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.