



**WIKIPEDIA**  
The Free Encyclopedia

[Main Page](#)  
[Recent changes](#)  
[Random page](#)  
[Current events](#)

[Edit this page](#)  
[Discuss this page](#)  
[Page history](#)  
[What links here](#)  
[Related changes](#)

[Special pages](#)  
[Contact us](#)  
[Donations](#)

[Main Page](#) | [Recent changes](#) | [Edit this page](#) | [Page history](#)  
[Printable version](#)

Not logged in  
[Log in](#) | [Help](#)

[Go](#) [Search](#)

Other languages: [Deutsch](#) | [Español](#) | [Français](#) | [? ? ? \(Nihongo\)](#) | [Polski](#) | [? ? \(Zhongwen\)](#)

## Smalltalk programming language

(Redirected from [Smalltalk-80](#))

**Smalltalk** is a dynamically typed [object oriented programming language](#) designed at [Xerox PARC](#) by [Alan Kay](#), [Dan Ingalls](#), [Ted Kaehler](#), [Adele Goldberg](#), and others during the [1970s](#). The language was generally released as Smalltalk-80 and has been widely used since.

In spite of its 20-year history, it is widely believed that the overall programming experience and productivity of Smalltalk is still unsurpassed by other development environments. Smalltalk is in continuing active development, and has gathered a loyal community of users around it.

Smalltalk has been had a great influence on the development of many other computer languages, including: [Objective-C](#), Actor, [Java](#) and [Ruby](#). Many software development ideas of the [1990s](#) came from the Smalltalk community, such as [Design Patterns](#) (as applied to software), [Extreme Programming](#) and [Refactoring](#). Among Smalltalkers is [Ward Cunningham](#), the inventor of the [WikiWiki](#) concept.

Smalltalk's big ideas include:

- "Everything is an [object](#)." Strings, integers, booleans, class definitions, blocks of code, stack frames, memory are all represented as objects.
- Everything is available for modification. If you want to change the IDE, you can do it-- in a running system, without stopping to recompile and restart. If you want a new control construct in the language, you can add it. In some implementations, you can change even the syntax of the language, or the way the garbage collection works.
- Types are dynamic -- this means that you don't have to define types in the code which makes the language much more concise.
- [Garbage collection](#) is built in and invisible to the developer.
- Smalltalk programs are usually compiled to [bytecodes](#), run by a [virtual machine](#).
- [Dynamic translation](#): modern commercial virtual machines compile bytecodes to the native machine code for fast execution, a technique pioneered by Smalltalk-80 from ParcPlace Systems in mid-1980s. This idea was adopted by [Java](#) some ten years later and named "Just-in-time compilation", or JIT.

The following code example for finding the vowels in a string illustrates Smalltalk's style. ( | characters declare variables, : declares parameters, and think of [ and ] as { and } curly braces for the moment):

```
| aString vowels |
aString := 'This is a string'.
vowels := aString select: [:aCharacter | aCharacter isVowel].
```

In the last line, the string is sent a `select:` message with the code block following as an argument. Here's the code in the superclass `Collection` that does the work:

```
| newCollection |
newCollection := self species new.
self do: [:each |
    (aBlock value: each)
    ifTrue: [newCollection add: each]].
^newCollection
```

It responds to the message by iterating through its members (this is the `do:` method) evaluating a `Block` code once for each character; a `Block` (a `Character isVowel`) when evaluated creates a boolean, which is then sent `ifTrue:`. If the boolean is true, the character is added to a string to be returned. Because `select` is defined in the abstract class `Collection`, we can also use it like this:

```
| rectangles aPoint |
rectangles := OrderedCollection
    with: (Rectangle left: 0 right: 10 top: 100 bottom: 200)
    with: (Rectangle left: 10 right: 10 top: 110 bottom: 210).
aPoint := Point x: 20 y: 20.
collisions := rectangles select: [:aRect | aRect containsPoint:
```

## External Links

- "[Why Smalltalk?](#)" a community of Smalltalk developers.
- "[GoodStart Smalltalk Info](#)" a Smalltalk advocacy site.

## Implementations

- VisualWorks, see [Cincom Smalltalk website](#), [Wiki](#).
- IBM's VisualAge for Smalltalk, see <http://www-4.ibm.com/software/ad/smalltalk/>
- Dolphin Smalltalk, see <http://www.object-arts.com/Home.htm>
- Smalltalk/X, see [http://www.exept.de/sites/exept/english/Smalltalk/frame\\_uebersicht.html](http://www.exept.de/sites/exept/english/Smalltalk/frame_uebersicht.html)
- [Squeak](#), see <http://www.squeak.org/>
- Pocket Smalltalk which runs on a Palm Pilot, see <http://www.pocketsmalltalk.com/>
- Gnu Smalltalk, see <http://www.gnu.org/software/smalltalk/smalltalk.html>
- Smalltalk MT, see <http://www.objectconnect.com/>
- VisualWorks and Squeak are descendants of the original Smalltalk-80. [Here](#) is a screenshot of Smalltalk-80 running inside VisualWorks (that is, the virtual machine running Smalltalk-80 is itself a Smalltalk program running in VisualWorks).

## Tutorials

- [Free Smalltalk Books](#)
- There are [WikiWiki](#) implemented in Squeak Smalltalk (swiki) and VisualWorks (WikiWorks). You can download Swiki from <http://minnow.cc.gatech.edu/swiki> WikiWorks is included in VisualWorks distribution (available for free for non-commercial use, see the link above).

---

[Edit this page](#) | [Discuss this page](#) | [Page history](#) | [What links here](#) | [Related changes](#)

Other languages: [Deutsch](#) | [Español](#) | [Français](#) | [? ? ? \(Nihongo\)](#) | [Polski](#) | [? ? \(Zhongwen\)](#)

[Main Page](#) | [About Wikipedia](#) | [Recent changes](#) |

 Search:

This page was last modified 06:22, 20 Nov 2003. All text is available under the terms of the [GNU Free Documentation License](#).

