|  |  |
| --- | --- |
| class {base} | R Documentation |

**Object Classes**

**Description**

**R** possesses a simple generic function mechanism which can be used for an object-oriented style of programming. Method dispatch takes place based on the class of the first argument to the generic function.

**Usage**

class(x)

class(x) <- value

unclass(x)

inherits(x, what, which = FALSE)

oldClass(x)

oldClass(x) <- value

**Arguments**

|  |  |
| --- | --- |
| x | a **R** object |
| what, value | a character vector naming classes. value can also be NULL. |
| which | logical affecting return value: see ‘Details’. |

**Details**

Here, we describe the so called “S3” classes (and methods). For “S4” classes (and methods), see ‘Formal classes’ below.

Many **R** objects have a class attribute, a character vector giving the names of the classes from which the object *inherits*. If the object does not have a class attribute, it has an implicit class, "matrix", "array" or the result of [mode](http://127.0.0.1:14695/library/base/help/mode)(x) (except that integer vectors have implicit class "integer"). (Functions oldClass and oldClass<- get and set the attribute, which can also be done directly.)

When a generic function fun is applied to an object with class attribute c("first", "second"), the system searches for a function called fun.first and, if it finds it, applies it to the object. If no such function is found, a function called fun.second is tried. If no class name produces a suitable function, the function fun.default is used (if it exists). If there is no class attribute, the implicit class is tried, then the default method.

The function class prints the vector of names of classes an object inherits from. Correspondingly, class<- sets the classes an object inherits from. Assigning NULL removes the class attribute.

unclass returns (a copy of) its argument with its class attribute removed. (It is not allowed for objects which cannot be copied, namely environments and external pointers.)

inherits indicates whether its first argument inherits from any of the classes specified in the what argument. If which is TRUE then an integer vector of the same length as what is returned. Each element indicates the position in the class(x) matched by the element of what; zero indicates no match. If which is FALSE then TRUE is returned by inherits if any of the names in what match with any class.

All but inherits are [primitive](http://127.0.0.1:14695/library/base/help/primitive) functions.

**Formal classes**

An additional mechanism of *formal* classes, nicknamed “S4”, is available in package **methods** which is attached by default. For objects which have a formal class, its name is returned by class as a character vector of length one and method dispatch can happen on *several* arguments, instead of only the first. However, S3 method selection attempts to treat objects from an S4 class as if they had the appropriate S3 class attribute, as does inherits. Therefore, S3 methods can be defined for S4 classes. See the ‘[Introduction](http://127.0.0.1:14695/library/methods/html/Introduction.html)’ and ‘[Methods\_for\_S3](http://127.0.0.1:14695/library/base/help/Methods_for_S3)’ help pages for basic information on S4 methods and for the relation between these and S3 methods.

The replacement version of the function sets the class to the value provided. For classes that have a formal definition, directly replacing the class this way is strongly deprecated. The expression [as](http://127.0.0.1:14695/library/base/help/as)(object, value) is the way to coerce an object to a particular class.

The analogue of inherits for formal classes is [is](http://127.0.0.1:14695/library/base/help/is). The two functions behave consistently with one exception: S4 classes can have conditional inheritance, with an explicit test. In this case, is will test the condition, but inherits ignores all conditional superclasses.

**Note**

Functions oldClass and oldClass<- behave in the same way as functions of those names in S-PLUS 5/6, *but* in **R** [UseMethod](http://127.0.0.1:14695/library/base/help/UseMethod) dispatches on the class as returned by class (with some interpolated classes: see the link) rather than oldClass. *However*, [group generic](http://127.0.0.1:14695/library/base/help/group%20generic)s dispatch on the oldClass for efficiency, and [internal generic](http://127.0.0.1:14695/library/base/help/internal%20generic)s only dispatch on objects for which [is.object](http://127.0.0.1:14695/library/base/help/is.object) is true.

In some versions of **R**, assigning a zero-length vector with class removes the class: in others it is an error (whereas it works for oldClass. It is clearer to always assign NULL to remove the class.

**See Also**

[UseMethod](http://127.0.0.1:14695/library/base/help/UseMethod), [NextMethod](http://127.0.0.1:14695/library/base/help/NextMethod), ‘[group generic](http://127.0.0.1:14695/library/base/help/group%20generic)’, ‘[internal generic](http://127.0.0.1:14695/library/base/help/internal%20generic)’

**Examples**

x <- 10

class(x) # "numeric"

oldClass(x) # NULL

inherits(x, "a") #FALSE

class(x) <- c("a", "b")

inherits(x,"a") #TRUE

inherits(x, "a", TRUE) # 1

inherits(x, c("a", "b", "c"), TRUE) # 1 2 0