register.   A *register* is a named location for storing a value. It is much like a variable in a programming language. T<small>E</small>X has five kinds of registers, as shown in the following table:

Register type	Contents
box	a box
count	a number
dimen	a dimension
muskip	muglue
skip	glue
toks	a token list

The registers of each type are numbered from 0 to 255. You can access register n of category c by using the form '\cn', e.g., \muskip192. You can use a register anywhere that information of the appropriate type is called for. For instance, you can use \count12 in any context calling for a number or \skip0 in any context calling for glue.

You put information into a register by assigning something to it:

```
\setbox3 = \hbox{lagomorphs are not mesomorphs}
\count255 = -1
```

The first assignment constructs an hbox and assigns it to box register 3. You can subsequently use '\box3' wherever a box is called for, and you will get just that hbox.[1] The second assignment assigns -1 to count register 255.

A register of a given type, e.g., a glue register, behaves just like a parameter of that type. You retrieve its value or assign to it just as you would with a parameter. Some T<small>E</small>X parameters, e.g., \pageno, are implemented as registers, in fact.

Plain T<small>E</small>X uses many registers for its own purposes, so you should not just pick an arbitrary register number when you need a register. Instead you should ask T<small>E</small>X to reserve a register by using one of the commands \newbox, \newcount, \newdimen, \newmuskip, \newskip, or \newtoks (p. '\@newbox'). These commands are outer, so you can't use them in a macro definition. If you could, you'd use up a register every time the macro was called and probably run out of registers before long.

Nonetheless you can with some caution use any register temporarily within a group, even one that T<small>E</small>X is using for something else. After T<small>E</small>X finishes executing the commands in a group, it restores the contents of every register to what they were before it started executing the group. When you use an explicitly numbered register inside a group, you must be sure that the register isn't modified by any macro that you might call

---

[1] But note carefully: using a box register also empties it so that its contents become void. The other kinds of registers don't behave that way. You can use the \copy command (p. '\copy') to retrieve the contents of a box register without emptying it.

2   \   §*0*

within the group. Be especially careful about using arbitrary registers in
a group that calls macros that you didn't write yourself.

T_EX reserves certain registers for special purposes: `\count0` through
`\count9` for page numbering information and `\box255` for the contents of
a page just before it is offered to the output routine. Registers `\dimen0`–
`\dimen9`, `\skip0`–`\skip9`, `\muskip0`–`\muskip9`, `\box0`–`\box9`, and the
255 registers other than `\box255` are generally available as "scratch" reg-
isters. Thus plain T_EX provides only one scratch register, `\count255`, for
counts. See pages 122 and 346 of *The T_EXbook* for conventions to follow
in choosing register numbers.

You can examine the contents of registers during a T_EX run with the
`\showthe` command (p. '`\showthe`'), e.g., with '`\showthe\dimen0`'.