1 Introduction
The amsopn package provides a command \DeclareMathOperator for defining
new ‘math operator names’ similar to the standard function names \sin, \lim,
\max, etc.

Standard file identification.
\NeedsTeXFormat{LaTeX2e}% LaTeX 2.09 can’t be used (nor non-LaTeX)
[1994/12/01]% LaTeX date must December 1994 or later
\ProvidesPackage{amsopn}[2022/04/08 v2.04 operator names]

What \nolimits@ does is keep a \limits typed by the user from having
any effect. This is used for operator names whose standard usage is never to
have limits.
\def\nolimits@{\@ifnextchar\limits{\nolimits\@gobble}{\nolimits}}

In operator names, it is sometimes desired to have text-mode punctuation
characters such as *-/:. Because the body of an operator name is set in math
mode, these few punctuation characters will not come out right (wrong symbol
and/or wrong spacing). The purpose of \newmcodes@ is to make them act like
their normal text versions.

Where practical, we use decimal numbers to cut down main mem usage ("not needed). Use \Umathcodedefnum with xetex and LuaTeX to avoid problems
using \mathcode if - has already been defined using \Umathcode.
\begingroup \catcode"=12
\ifx\Umathcode\@undefined
\gdef\newmcodes@{\mathcode\'39\mathcode\"42\mathcode\."613A
\Define\std@minus\relbar use; otherwise there are problems with arrows
constructed with relbar.
\ifnum\mathcode\'=-45 \else
\mathchardef\std@minus\mathcode\'\relax
\fi
\mathcode\'=-45\mathcode\'/47\mathcode\":603A\relax
\endgroup

1
The command \texttt{\operatorname} prints its argument as a ‘math operator’ like \texttt{\sin} or \texttt{\det}, with proper font and spacing.

In the interior of the \texttt{\mathop} we need a null object (we choose a zero kern for minimum waste of main mem) in order to guard against the case where \texttt{\#3} is a single letter; \TeX\ will seize it and center it on the math axis if there is nothing else inside the \texttt{\mathop} atom.

The command \texttt{\DeclareMathOperator} defines the first argument to be an operator name whose text is the second argument. The star form means that the operator name should take limits (like \texttt{\max} or \texttt{\lim}).

In the basic set of operator names (below) we did not use \texttt{\DeclareRobustCommand} because of the hash table cost. But we use it here to minimize the chances of trouble, since we are producing a user-defined command.
This command is provided to allow the document styles to decide in which way math operators like ‘max’ or ‘log’ are typeset. The default is to set them in \langle math group \rangle zero of the current math version. The original name was \operatorfont, retained for compatibility; the second name was added to make it more accessible so that users can call this font for use in special constructs that are not ordinary operator names but conceptually related.

\operatorfont is also declared by the LATEX kernel (for at least 14 years), thus defining it here effectively means “resetting it” to the kernel value, which is counterproductive in situations where the user (or a class) has altered its definition and at a later point amsopn got added.

\%\def\operatorfont{\mathgroup\symoperators} % commented out in 2.03
\def\operatorfont{\operatorfont}

For backwards compatibility we keep this old command name for the time being:

\def\operatornamewithlimits{\operatorname*}

These macros use \mathpalette so that they will change size in script and scriptscript styles, though it’s hard to imagine they will ever be used there (the arrows, particularly, look bad in subscript sizes). Notice that the use of \exp means that the vertical spacing may not be optimal in script and scriptscript sizes. Unfortunately \LaTeX provides no easy way to do math mode vertical spacing that varies with current math style like mu units.
If we don’t load the \texttt{amsgen} package then the use of \texttt{ex@} in \texttt{\varlim@} will lead to trouble.

\texttt{\RequirePackage{amsgen}}

The usual \texttt{\endinput} to ensure that random garbage at the end of the file doesn’t get copied by \texttt{docstrip}.

\texttt{\endinput}