`\ifx` ⟨*token*$_1$⟩ ⟨*token*$_2$⟩

This command tests if ⟨*token*$_1$⟩ and ⟨*token*$_2$⟩ agree. Unlike `\if` and `\ifcat`, `\ifx` does *not* expand the tokens following `\ifx`, so ⟨*token*$_1$⟩ and ⟨*token*$_2$⟩ are the two tokens immediately after `\ifx`. There are three cases:

1) If one token is a macro and the other one isn't, the tokens don't agree.
2) If neither token is a macro, the tokens agree if:
   a) both tokens are characters (or control sequences denoting characters) and their character codes and category codes agree, or
   b) both tokens refer to the same TEX command, font, etc.
3) If both tokens are macros, the tokens agree if:
   a) their "first level" expansions, i.e., their replacement texts, are identical, and
   b) they have the same status with respect to `\long` (p. '`\long`') and `\outer` (p. '`\outer`').

Note in particular that *any two undefined control sequences agree.*

This test is generally more useful than `\if`.

*Example:*

```
\ifx\alice\rabbit true\else false\fi;
% true since neither \rabbit nor \alice is defined
\def\a{a}%
\ifx a\a true\else false\fi;
% false since one token is a macro and the other isn't
\def\first{\a}\def\second{\aa}\def\aa{a}%
\ifx \first\second true\else false\fi;
% false since top level expansions aren't the same
\def\third#1:{(#1)}\def\fourth#1?{(#1)}%
\ifx\third\fourth true\else false\fi
% false since parameter texts differ
```

*produces:*

true; false; false; false