assert verifies the truth of a claim:

```
assert sex=="m" | sex=="f"
assert age>=18 & age<=65
22 contradictions in 2740 observations
assertion is false
r(9);
```

The best feature of `assert` is that, when the claim is false, it stops do-files and ado-files:

```
do my_data_prep
  use basedata, clear
  assert age>=18 & age<=64
  22 contradictions in 2740 observations
  assertion is false
  r(9);
end of do-file
r(9);
```

`assert` has two main uses:

1. It checks that claims made to you and suppositions you have made about the data you are about to process are true:

```
assert exp==. if age<18
assert exp<. if age>=18
```

2. It tests that, when you write complicated code, the code produces what you expect:

```
sort group
  by group: gen avg = sum(hours)/sum(hours<.)
  by group: assert avg!=. if _n==_N
  by group: gen relative = hours/avg[_N]
```

`assert` is especially useful following `merge`:

```
merge id
  sort id using demog
  assert _merge==3
  drop _merge
```

© 2003 StataCorp LP  dm0003