The main topic for today is identity.

Identity is a two place predicate, but it has some special features.
To reflect that, we write it using 'infix' notation.
So we write "a = b", rather than "=ab".
And we write "a ≠ b" as shorthand for "¬a = b".

Identity is part of logic.
This means that it has rules in tableau, and constraints on models.

The constraints on models are simple.
"a = b" is true in a model iff v(a) and v(b) are the same thing.
So if you are trying to build a model where "a=b" is true, make them the same.

There are two rules in tableau, both of them distinctive.
At any time, for any name 'a' in the tableau, you can write "a=a".
And if the branch includes "a=b" or "b=a", and an **atomic** sentence A(a), you can write A(b).
This is the first rule that has two different lines as input.
Note that from "Saa" and "a=b", the rule lets you infer three things:

- Sab
- Sba
- Sbb

$Pa \vdash \exists x(x = a \wedge Px)$

1. Pa
2. $\neg \exists x(x = a \wedge Px)$

$\exists x(Px \wedge \forall y(Py \supset x = y)), \exists x(Px \wedge Qx) \mid\!- \forall y(Py \supset Qy)$

1. $\exists x(Px \wedge \forall y(Py \supset x = y))$
2. $\exists x(Px \wedge Qx)$
3. $\neg \forall y(Py \supset Qy)$

Here is an invalid one to show how to build models

$\exists xPx, a=b \mid\!- \exists xQx$

1.	$\exists xPx$
2.	$a=b$
3.	$\neg\exists xQx$