Tactic	Description	Example
`rfl`	Prove equalities that hold *by definition*.	`example : 1 = 1 := by` `  rfl`
`simp`	Simplifies trivial expressions such as `a + 0`, `1 * a` or `0 * a`.	`example (a : ℤ) : 1 * a + 0 = a := by` `  simp`
`numbers`	Prove (in)equalities between purely numerical expressions.	`example : 5 ^ 3 < 2 ^ 7 := by` `  numbers`
`algebra`	Prove algebraic identities.	`example (x y : ℝ) : (x + y) * (x - y) = x ^ 2 - y ^ 2 := by` `  algebra`
`rewrite [h]`, `rewrite [←h]`	If hypothesis `h` is of the form `a = b`, replace `a` with `b` in the goal. The version with ←h replaces `b` with `a`.	`example (x : ℚ) (h : x = 2) : x ^ 2 = 4 := by` `  rewrite [h]; numbers`
`positivity`	Prove goals of the form `a > 0` or `a ≥ 0`.	`example (x : ℝ) : x ^ 2 ≥ 0 := by` `  positivity`
`extra`	Prove inequalities of the form `a + e > a` or `a + e ≥ a`, where `e` is (strictly) positive.	`example (a b : ℤ) (h : a ≥ 0) : a + b ≥ b := by` `  extra`
`rel [h]`	Use inequality `h` to prove a directly related inequality.	`example (a b : ℝ) (h : a ≥ b) : a + 1 ≥ b + 1 := by` `  rel [h]`
`linarith`	A powerful tactic to automatically prove linear inequalities from linear (in)equalities in the hypotheses.	`example (a b : ℝ) (h1 : a > 2) (h2 : a + b < 3) : b < 1 := by` `  linarith`

`calc`	Chain (in)equalities together to prove an (in)equality.	`example (x y : ℝ) : x ^ 2 + y ^ 2 - 2 * x * y ≥ 0 := by`
		`  calc`
		`    x ^ 2 + y ^ 2 - 2 * x * y = (x - y) ^ 2 := by algebra`
		`                              _ ≥ 0         := by positivity`

`use`	Start proving an existential statement (∃) by providing the value that you will use.	`example : ∃ n : ℕ, 81 * n = 2025 := by`
		`  use 25`
		`  numbers`

`intro`	Start proving a universal statement (∀) by introducing an arbitrary variable.	`example : ∀ x : ℝ, x ^ 2 ≥ 0 := by`
		`  intro x`
		`  positivity`

`have`	Add another hypothesis (with proof). This becomes available to use in the rest of the proof.	`example (x : ℝ) (h : x + 1 > 2) : x * ` x^{-1} ` = 1 := by`
		`  have h2 : x ≠ 0 := by linarith`
		`  algebra`

`simple_induction n with k IH`	Prove something for all natural numbers `n` by induction. Creates a base case (`n=0`) and an inductive step (`n=k+1` with induction hypothesis `IH` for `n=k`).	`example (n : ℕ) : n ≥ 0 := by`
		`  simple_induction n with k IH`
		`  · numbers`
		`  · positivity`

`induction_from_starting_point n, h with k hk IH`	Prove something for all natural numbers `n ≥ C` by induction, given hypothesis `h : n ≥ C`. Creates a base case for `n=C` and an inductive step.	`example (n : ℕ) (h : n ≥ 3) : n ≥ 3 := by`
		`  induction_from_starting_point n, h with k hk IH`
		`  · numbers`
		`  · linarith`

Symbol	Type	Description
$\mathbb{N}$	`\N`	natural numbers
$\mathbb{Z}$	`\Z`	integers
$\mathbb{Q}$	`\Q`	rational numbers
$\mathbb{R}$	`\R`	real numbers
$\le$	`\le`	less than or equal to
$\ge$	`\ge`	greater than or equal to
π	`\pi`	pi
a^{-1}	`\inv`	inverse
$\forall$	`\forall`	for all
$\exists$	`\exists`	exists
$\leftarrow$	`\l`	used in `rewrite [←h]`
$\cdot$	`\.`	bullet (used for induction cases)