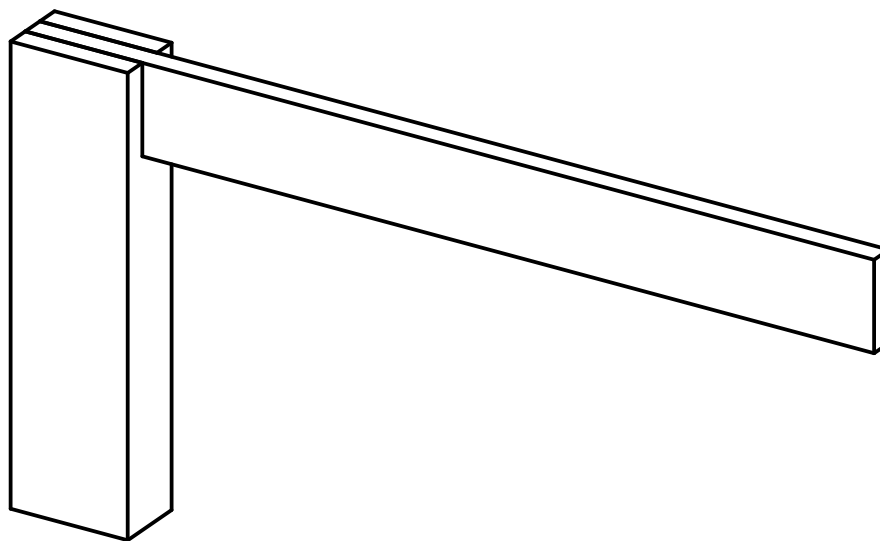


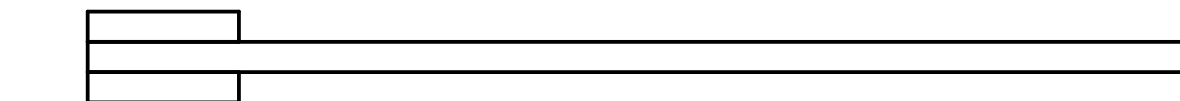
Try square



parts	2
overall	290 x 24 x 150 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

Contents

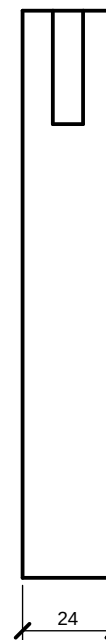
Drawings	3
Assembly	3
Parts	4
1 stock	5
2 blade	6
Cut list	7
Appendix: source	8



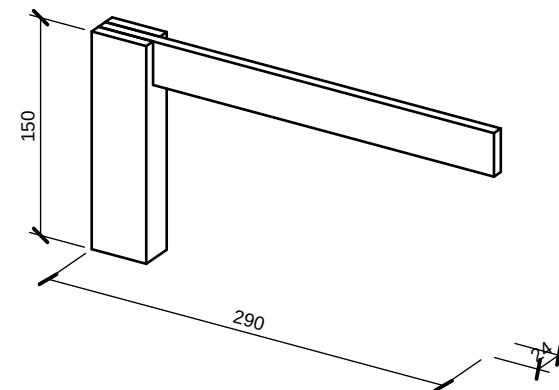
TOP



FRONT

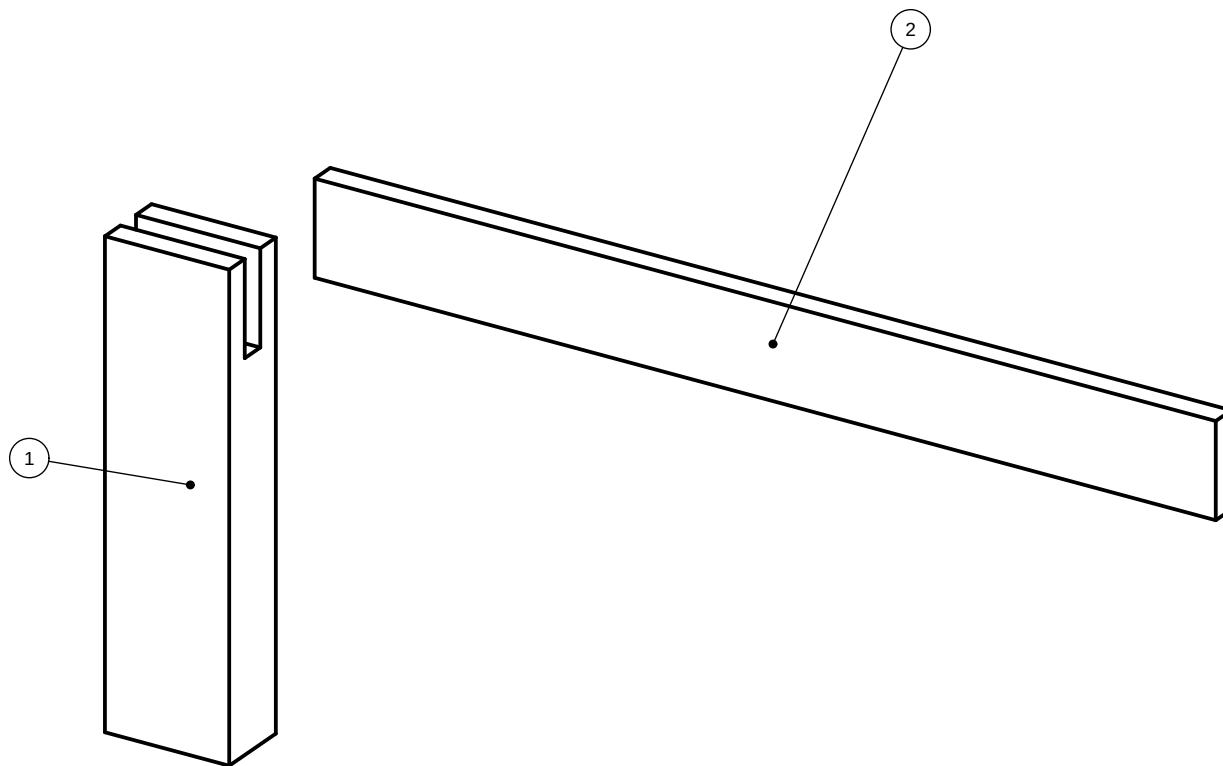


RIGHT SIDE



ISOMETRIC

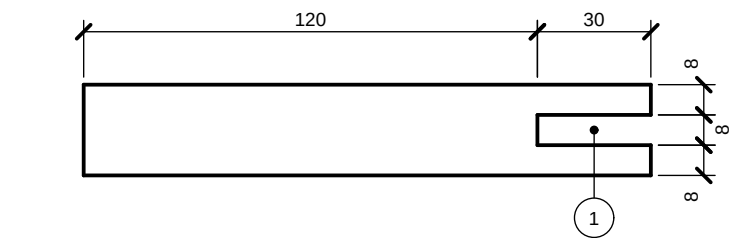
assembly	try_square
parts	2
overall	290 x 24 x 150 (W x D x H)
material	oak
scale	1:2, mm



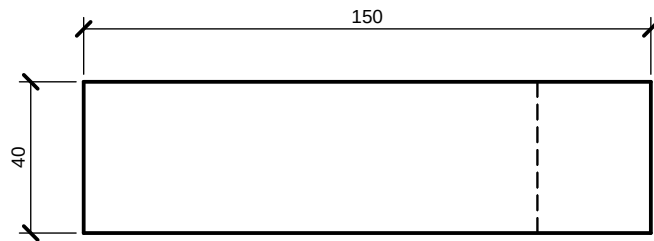
EXPLODED

no.	part	qty	cut size (L x W x T)	material
1	stock	1	150 x 40 x 24	oak
2	blade	1	290 x 30 x 8	oak

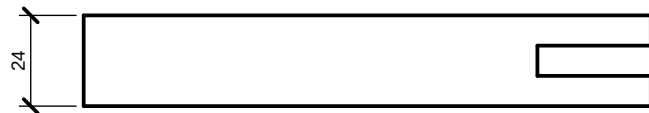
assembly	try_square
parts	2
overall	290 x 24 x 150 (W x D x H)
material	oak
scale	not to scale, mm



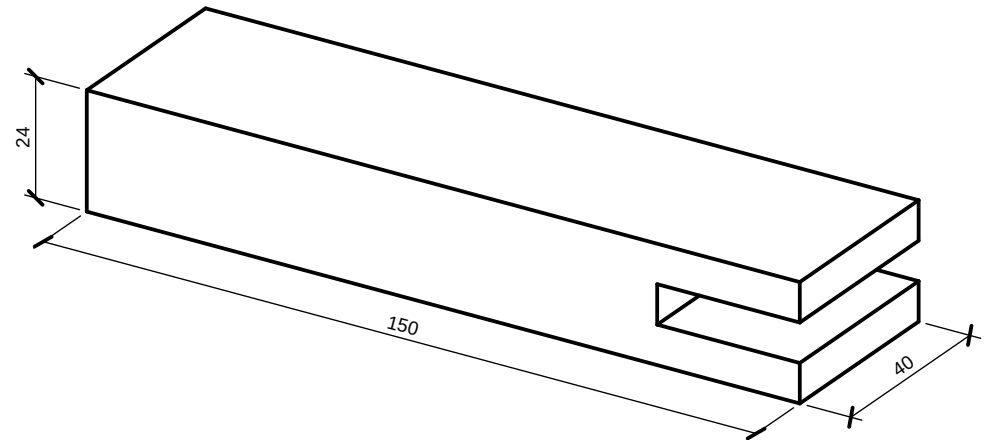
BACK EDGE



TOP FACE



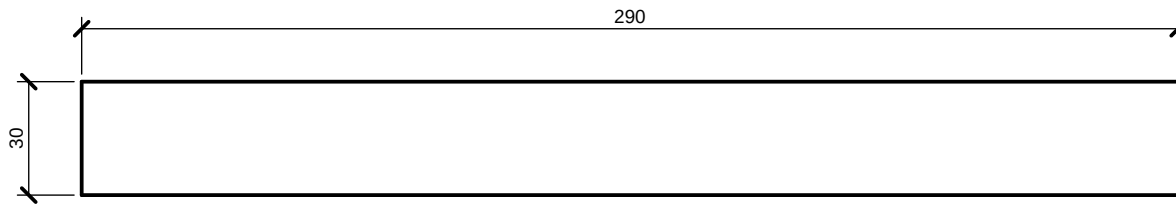
FRONT EDGE



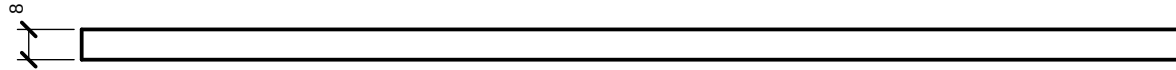
ISOMETRIC

1 slot mortise 8 mm, 30 deep, in right_end

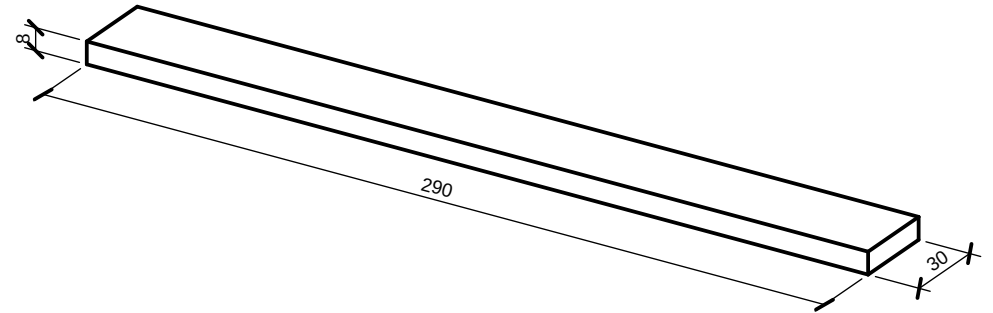
part	stock
of	try_square
make	1
cut size	150 x 40 x 24
material	oak
scale	1:2, mm



TOP FACE



FRONT EDGE



ISOMETRIC

part	blade
of	try_square
make	1
cut size	290 x 30 x 8
material	oak
scale	1:2, mm

Cut list

stock 150 x 40 x 24 mm oak
- slot mortise 8 mm, 30 deep, in right_end
blade 290 x 30 x 8 mm oak

Appendix: source

Made with woodwright 0.9.0, build123d 0.13.0, Python 3.14.4.

To make these plans again:

```
uvx --python 3.14 woodwright==0.9.0 export projects.try_square
```

projects.try_square, in woodwright/examples/projects.py

```
def try_square(
    stock_length: float = 150,
    stock_width: float = 40,
    blade_length: float = 250,
    blade_width: float = 30,
    stock: Stock = SQUARE_STOCK,
) -> Assembly:
    """A try square: a thin blade in a bridle slot at the end of a thick stock.

    The blade is as thin as the slot (a third of the stock's thickness), so
    it goes in whole, without cheeks, centered in the stock's thickness.
    """
    asm = Assembly("try_square", root_location=STANDING)
    handle = asm.add(Board("stock", stock, length=stock_length, width=stock_width))
    blade_stock = Stock.board(thickness=stock.thickness / 3, species=stock.species)
    blade = asm.add(Board("blade", blade_stock, length=blade_length, width=blade_width))
    asm.join(
        Bridle(
            tenon=blade.left_end,
            slot=handle.back_edge,
            position=flush_with(handle.right_end),
            tenon_thickness=blade_stock.thickness,
            across=CENTERED,
        )
    )
    return asm
```