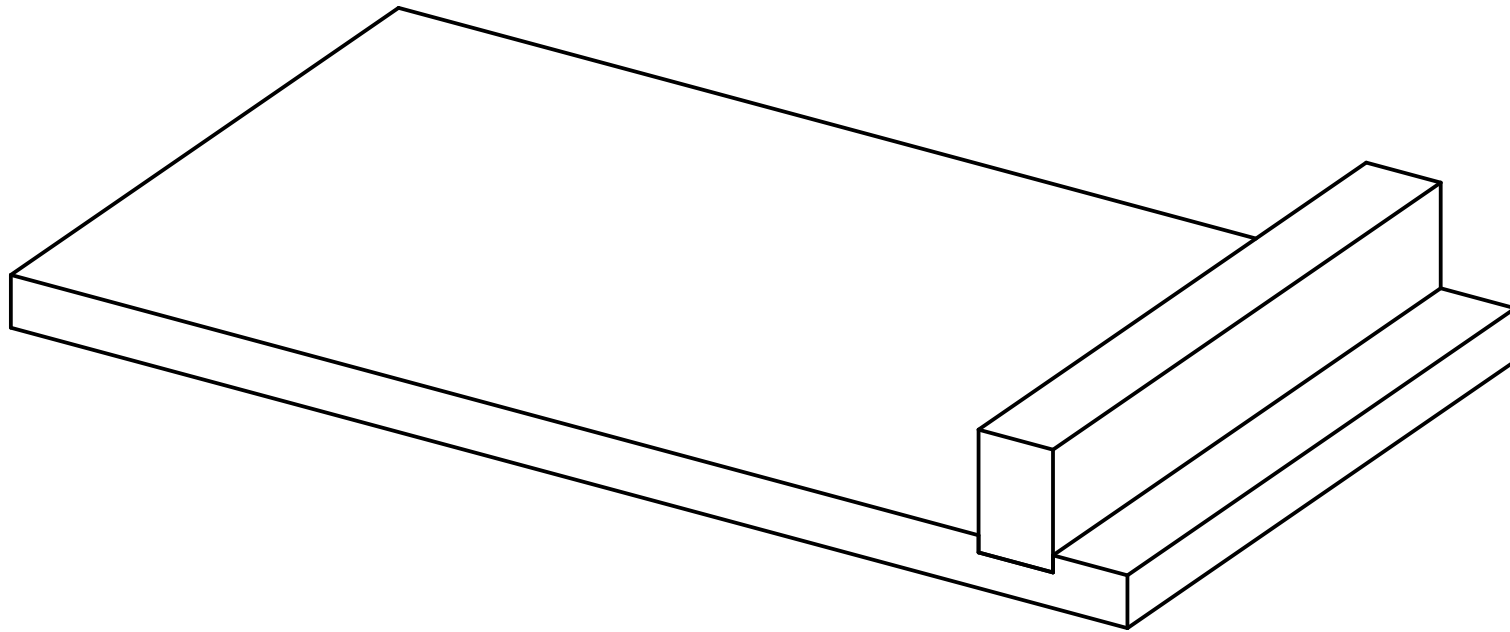


Shooting board



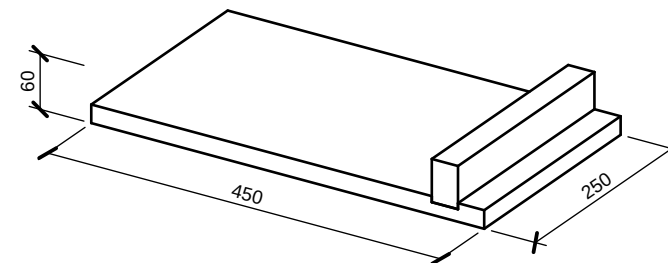
parts	2
overall	450 x 250 x 60 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

Contents

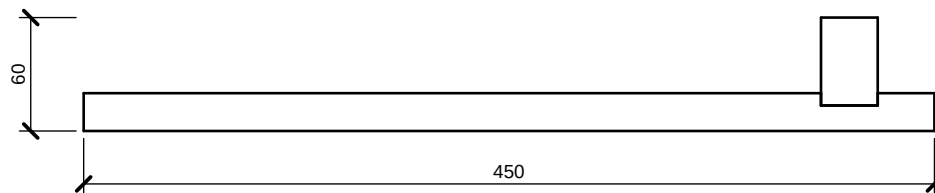
Drawings	3
Assembly	3
Parts	4
1 base	5
2 stop	6
Cut list	7
Appendix: source	8



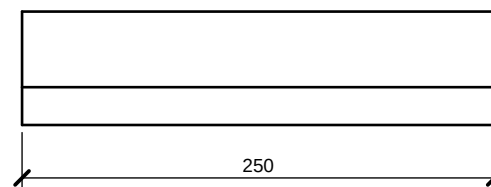
TOP



ISOMETRIC



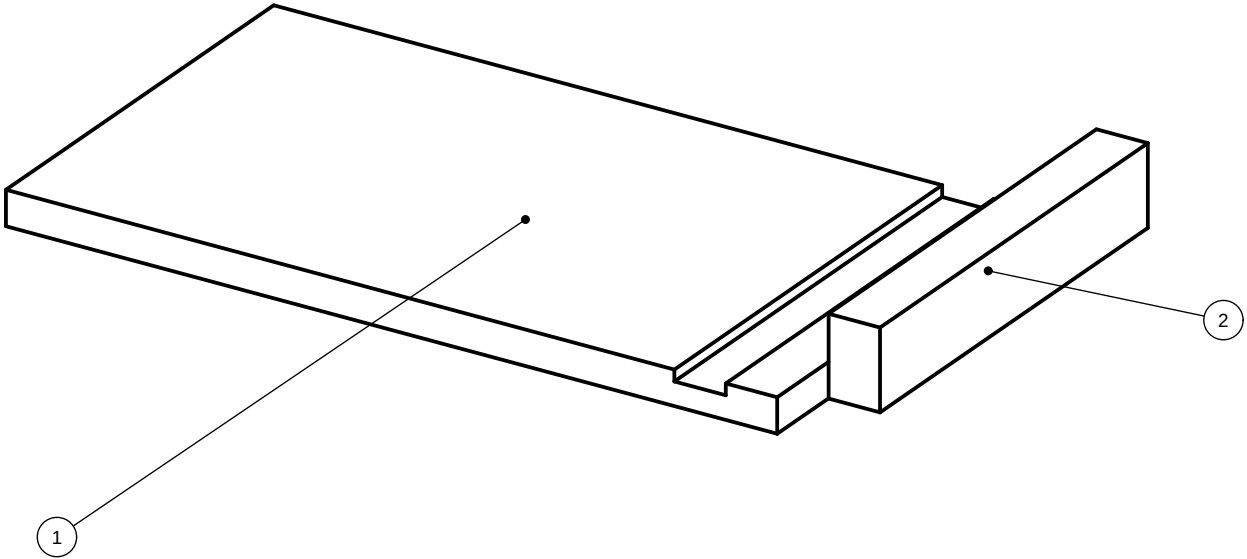
FRONT



RIGHT SIDE

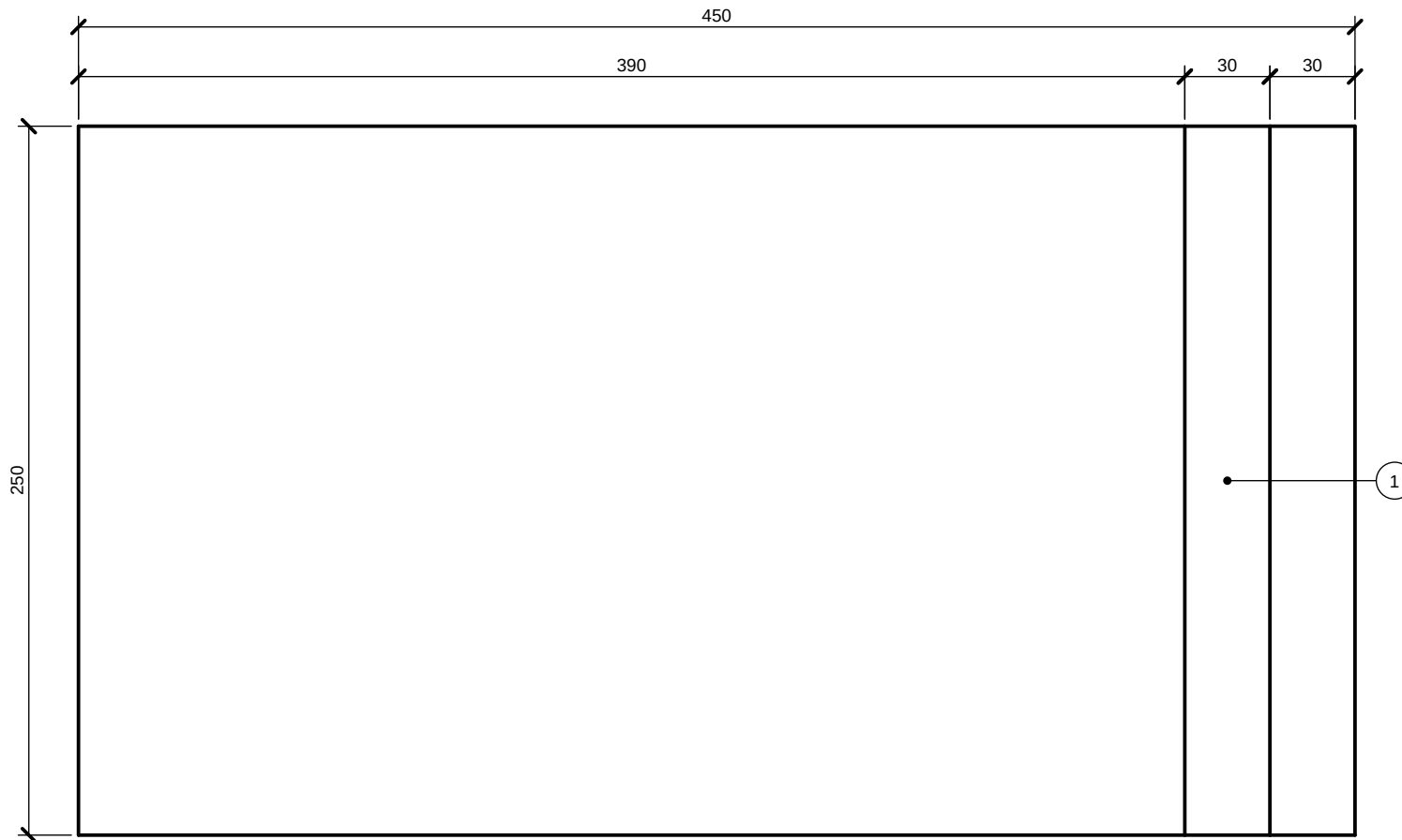
assembly	shooting_board
parts	2
overall	450 x 250 x 60 (W x D x H)
material	oak
scale	1:4, mm

no.	part	qty	cut size (L x W x T)	material
1	base	1	450 x 250 x 20	oak
2	stop	1	250 x 46.5 x 30	oak



EXPLODED

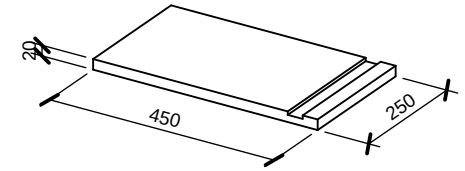
assembly	shooting_board
parts	2
overall	450 x 250 x 60 (W x D x H)
material	oak
scale	not to scale, mm



TOP FACE



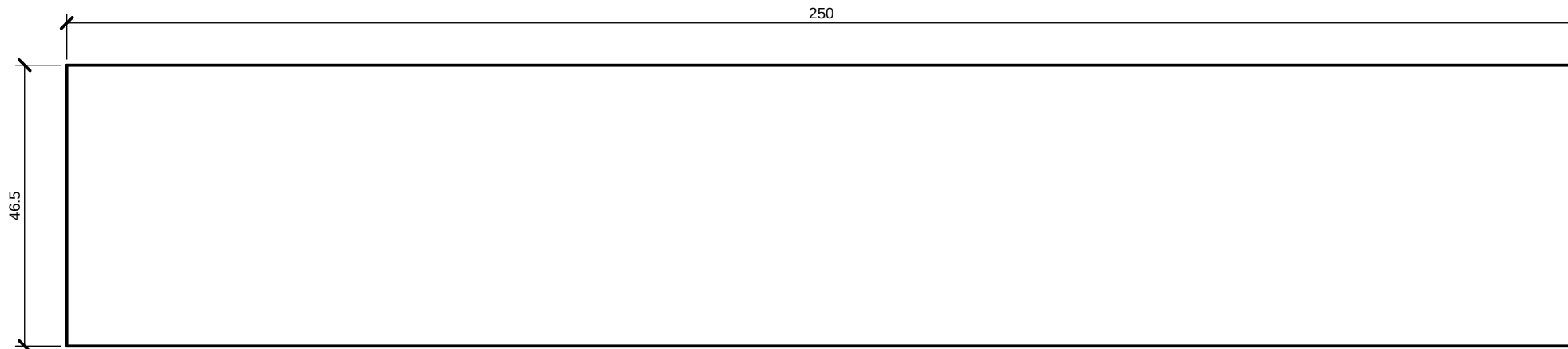
FRONT EDGE



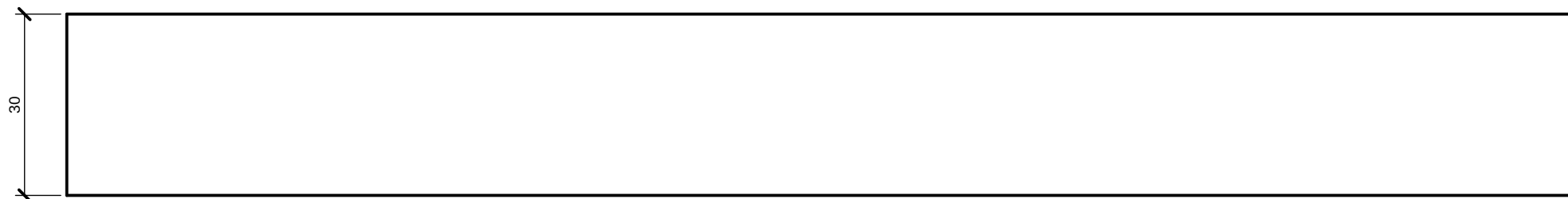
ISOMETRIC

1 dado 30 mm, 6.5 deep, in top_face

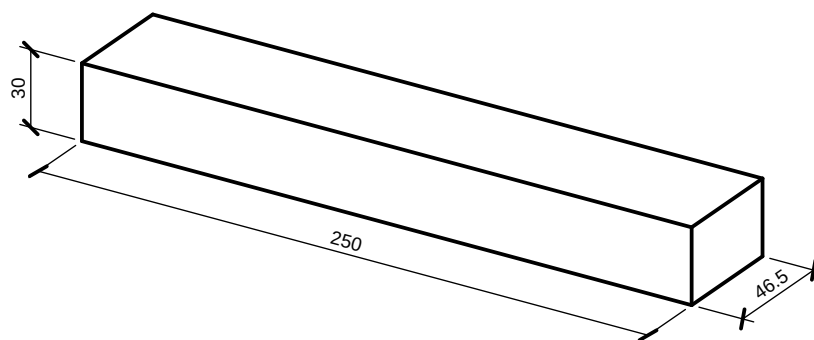
part	base
of	shooting_board
make	1
cut size	450 x 250 x 20
material	oak
scale	1:2.5, mm



TOP FACE



FRONT EDGE



ISOMETRIC

part	stop
of	shooting_board
make	1
cut size	250 x 46.5 x 30
material	oak
scale	1:1, mm

Cut list

base 450 x 250 x 20 mm oak
- dado 30 mm, 6.5 deep, in top_face
stop 250 x 46.5 x 30 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.shooting_board
```

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

```
def shooting_board(
    length: float = 450,
    width: float = 250,
    stop_width: float = 40,
    stop_from_end: float = 30,
    stock: Stock = OAK,
    stop_stock: Stock = THICK_OAK,
) -> Assembly:
    """A base board with a stop block across it, housed in a dado near one end."""
    asm = Assembly("shooting_board")
    base = asm.add(Board("base", stock, length=length, width=width))
    stop = asm.add(Board("stop", stop_stock, length=width, width=stop_width))
    asm.join(
        DadoJoint(
            dado=base.top_face,
            end=stop.front_edge,
            position=offset(stop_from_end, base.right_end),
            # The stop lies across the base: its end flush with the base's front edge.
            flush=(base.front_edge, stop.left_end),
        )
    )
    return asm
```