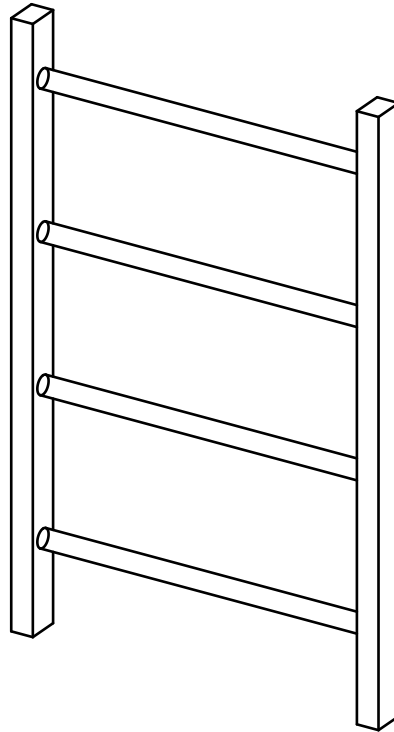


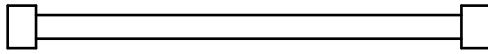
Ladder



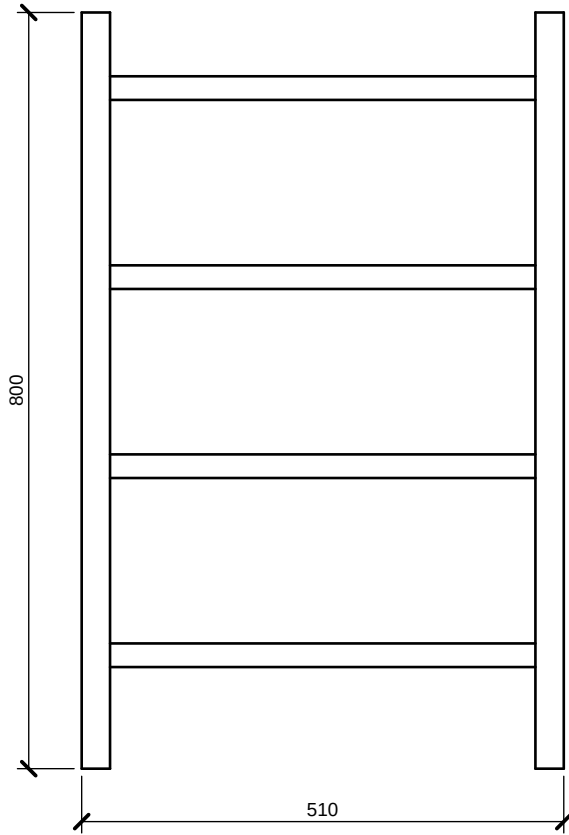
parts	6 (2 different)
overall	510 x 45 x 800 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

Contents

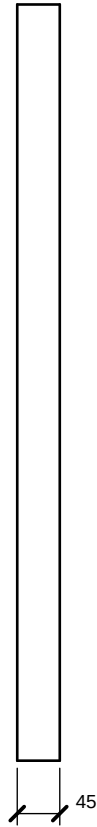
Drawings	3
Assembly	3
Parts	4
1 stile	5
2 rung	6
Cut list	7
Appendix: source	8



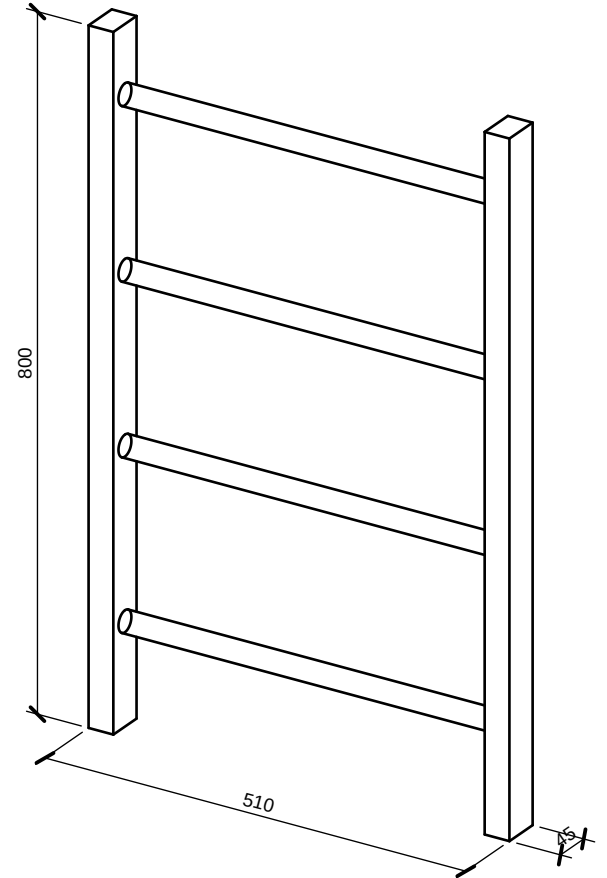
TOP



FRONT

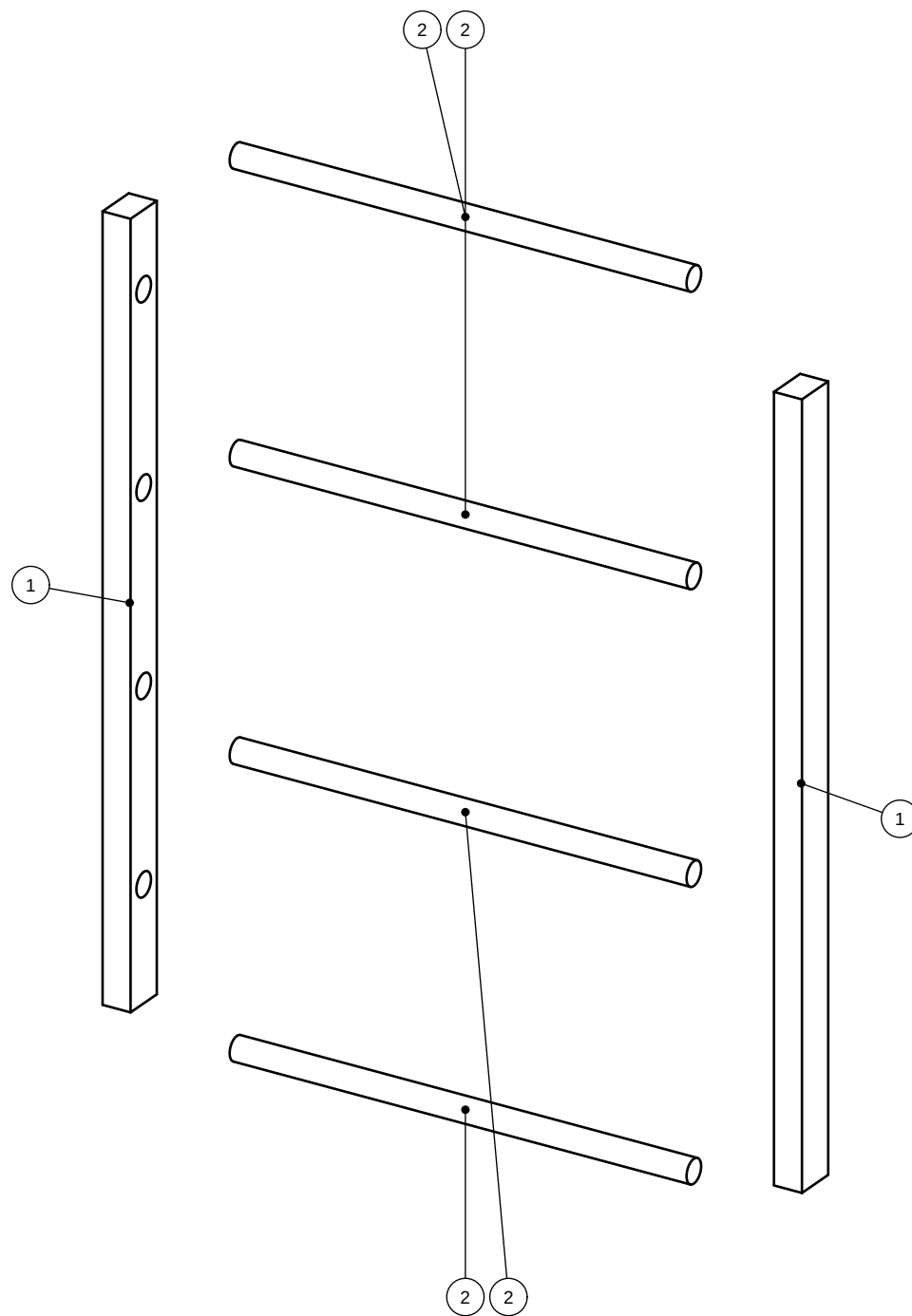


RIGHT SIDE



ISOMETRIC

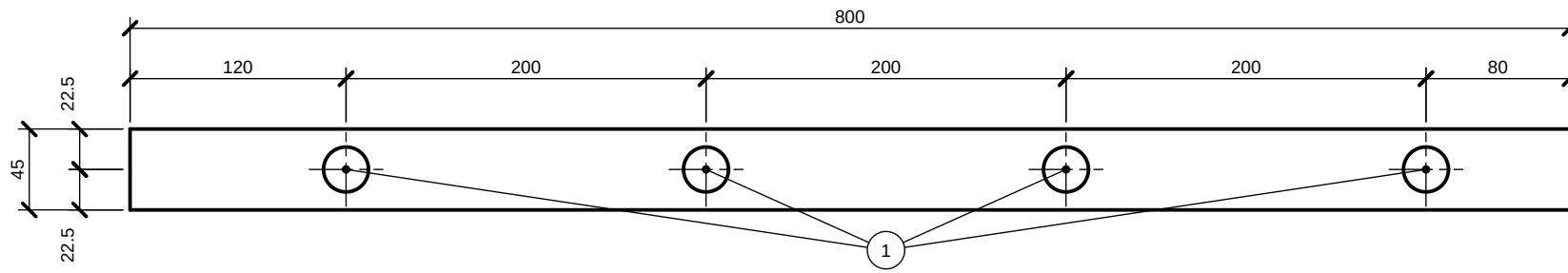
assembly	ladder
parts	6 (2 different)
overall	510 x 45 x 800 (W x D x H)
material	oak
scale	1:8, mm



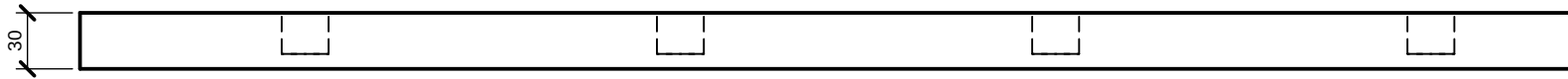
EXPLODED

no.	part	qty	cut size (L x W x T)	material
1	stile	2	800 x 45 x 30	oak
2	rung	4	ø25 x 490	oak

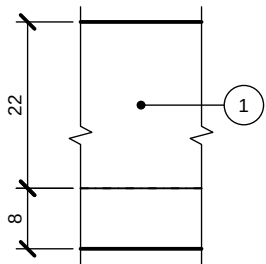
assembly	ladder
parts	6 (2 different)
overall	510 x 45 x 800 (W x D x H)
material	oak
scale	not to scale, mm



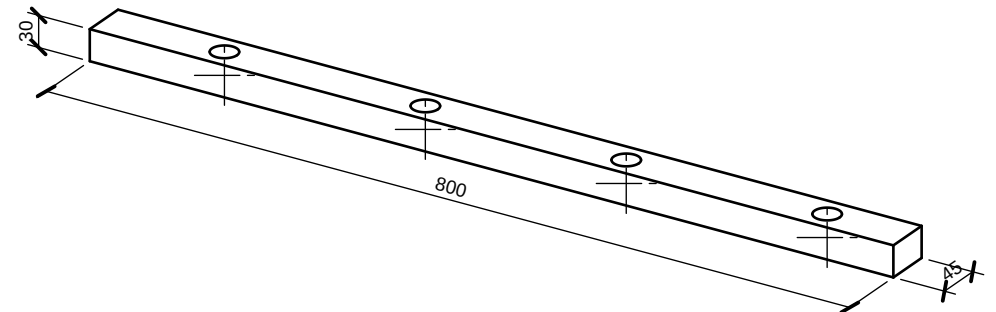
TOP FACE



FRONT EDGE



FRONT EDGE: 1 1:1

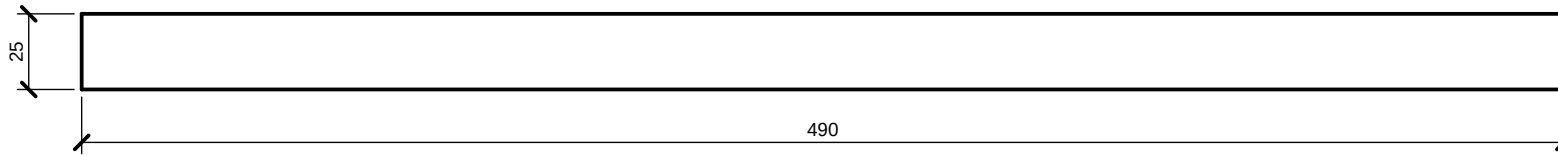


ISOMETRIC

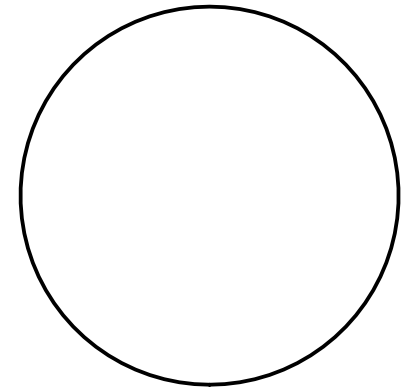
2 pieces: stile_left, stile_right

1 4 x hole $\varnothing 25$, 22 deep, in top_face

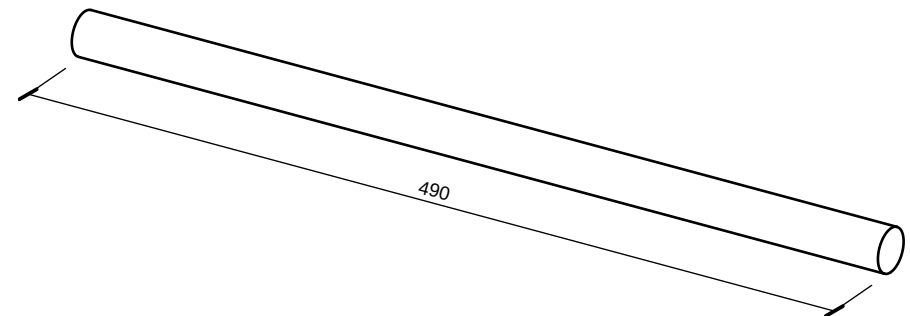
part	stile
of	ladder
make	2
cut size	800 x 45 x 30
material	oak
scale	1:4, mm



SIDE



RIGHT END 2:1



ISOMETRIC

4 pieces: rung_1, rung_2, rung_3, rung_4

part	rung
of	ladder
make	4
cut size	ø25 x 490
material	oak
scale	1:2.5, mm

Cut list

stile 800 x 45 x 30 mm oak, 2 pieces: stile_left, stile_right
- 4 x hole ø25, 22 deep, in top_face, 120 to 720 every 200 from left_end; centered
rung ø25 x 490 mm oak, 4 pieces: rung_1, rung_2, rung_3, rung_4

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 frames.ladder
```

frames.ladder, in woodwright/examples/frames.py

```
def ladder(
    height: float = 800,
    width: float = 450,
    rungs: int = 4,
    stile_width: float = 45,
    stock: Stock = LADDER_OAK,
) -> Assembly:
    """A towel rail (a short ladder): rods in blind holes in the inside faces
    of two stiles, from 120 above their feet, evenly up to 80 below their tops.

    `width` is between the stiles; each rod goes 20 mm into either.
    """
    asm = Assembly("ladder", root_location=orient(right_end=UP, ref_face=(1, 0, 0)))
    left = asm.add(Board("stile_left", stock, length=height, width=stile_width))
    # The mirror image of the left one, its ref face inside too, its left end at the foot.
    right = asm.add(Board("stile_right", stock, length=height, width=stile_width, mirror=True))
    rod_stock = Stock.rod(diameter=25, species="oak")
    pitch = (height - 120 - 80) / (rungs - 1)
    for i in range(rungs):
        rung = asm.add(Rod(f"rung_{i + 1}", rod_stock, length=width))
        at = offset(120 - rod_stock.diameter / 2 + i * pitch, ref=left.left_end)
        asm.join(RoundMortiseTenon(rung.left_end, left.top_face, length=20, position=at))
        at = offset(120 - rod_stock.diameter / 2 + i * pitch, ref=right.left_end)
        asm.join(RoundMortiseTenon(rung.right_end, right.top_face, length=20, position=at))
    return asm
```