

Hollow leg



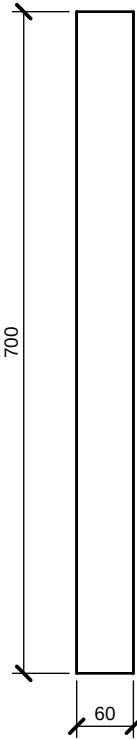
parts	4 (1 different)
overall	60 x 60 x 700 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

Contents

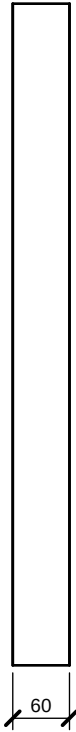
Drawings	3
Assembly	3
Parts	4
1 front	5
Cut list	6
Appendix: source	7



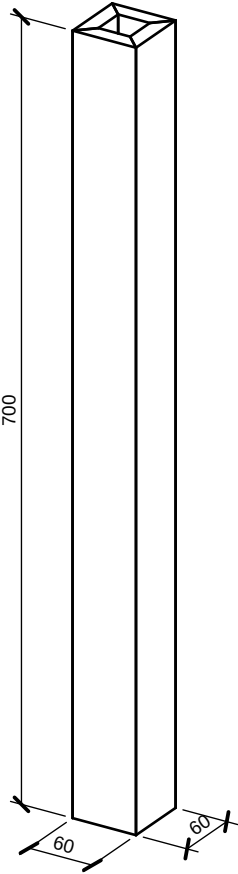
TOP



FRONT

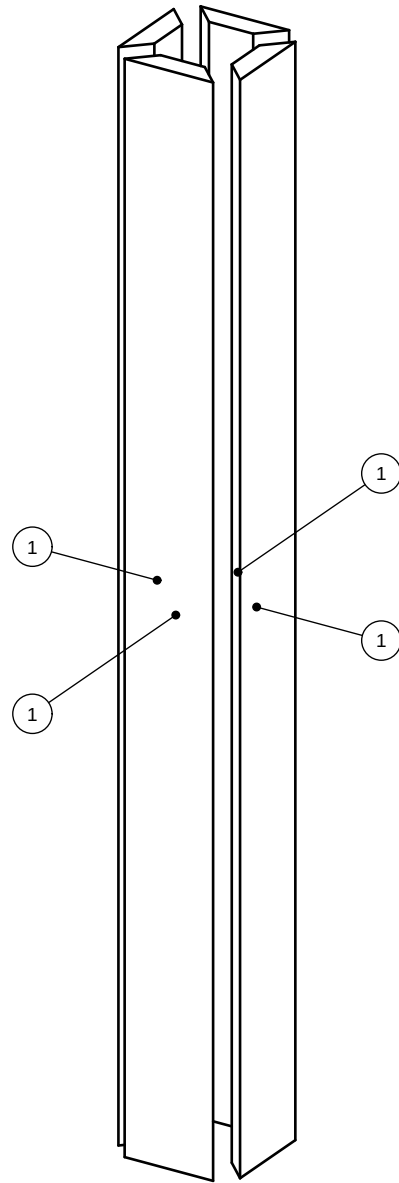


RIGHT SIDE



ISOMETRIC

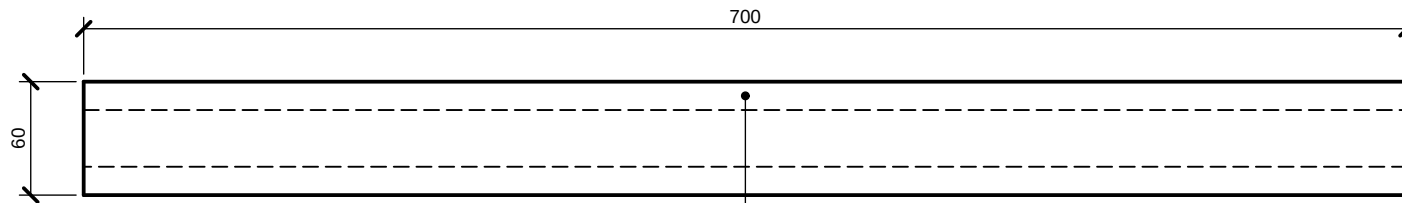
assembly	hollow_leg
parts	4 (1 different)
overall	60 x 60 x 700 (W x D x H)
material	oak
scale	1:8, mm



EXPLODED

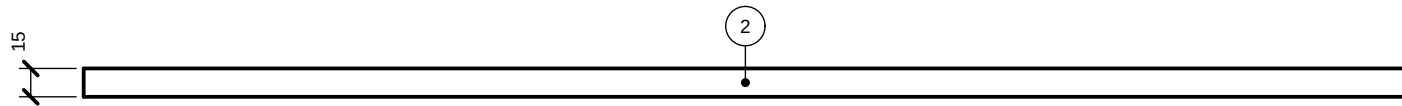
no.	part	qty	cut size (L x W x T)	material
1	front	4	700 x 60 x 15	oak

assembly	hollow_leg
parts	4 (1 different)
overall	60 x 60 x 700 (W x D x H)
material	oak
scale	not to scale, mm



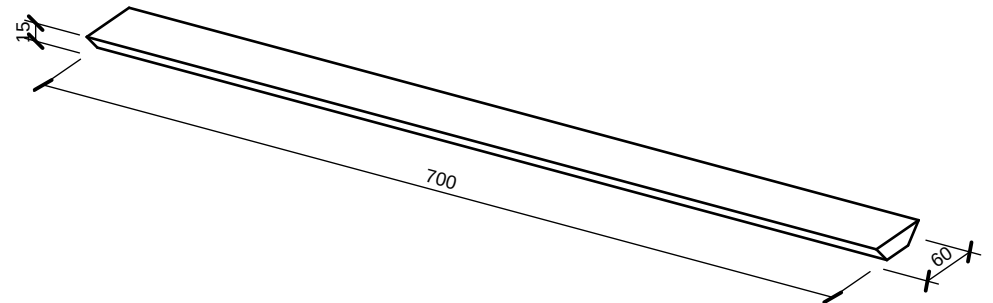
1

TOP FACE



2

FRONT EDGE



ISOMETRIC

4 pieces: front, left, back, right

1 mitre 45 deg on back_edge, bottom_face inside

2 mitre 45 deg on front_edge, bottom_face inside

part	front
of	hollow_leg
make	4
cut size	700 x 60 x 15
material	oak
scale	1:4, mm

Cut list

front 700 x 60 x 15 mm oak, 4 pieces: front, left, back, right
- mitre 45 deg on back_edge, bottom_face inside
- mitre 45 deg on front_edge, bottom_face inside

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 joints.hollow_leg
```

joints.hollow_leg, in woodwright/examples/joints.py

```
def hollow_leg(length: float = 700, width: float = 60, stock: Stock = HOLLOW_LEG_OAK) -> Assembly:
    """A square hollow leg: four boards glued along their edges, each edge
    mitred at 45 degrees along the grain, so they meet on the diagonals of
    the corners and no edge grain shows; outside sizes.

    Lighter than a solid leg and showing face grain on all four sides, with
    a square hole through its length (here 30 mm for 15 mm boards). The
    reference faces are outside; the boards stand with their left ends at
    the bottom.
    """
    asm = Assembly("hollow_leg", root_location=orient(right_end=UP, ref_face=FRONT))
    names = ("front", "left", "back", "right")
    boards = [asm.add(Board(name, stock, length=length, width=width)) for name in names]
    # Going round, each board's back edge meets the next one's front edge.
    for a, b in zip(boards, boards[1:] + boards[:1], strict=True):
        asm.join(MitreCorner(a.back_edge, b.front_edge, a.bottom_face, b.bottom_face))
    return asm
```