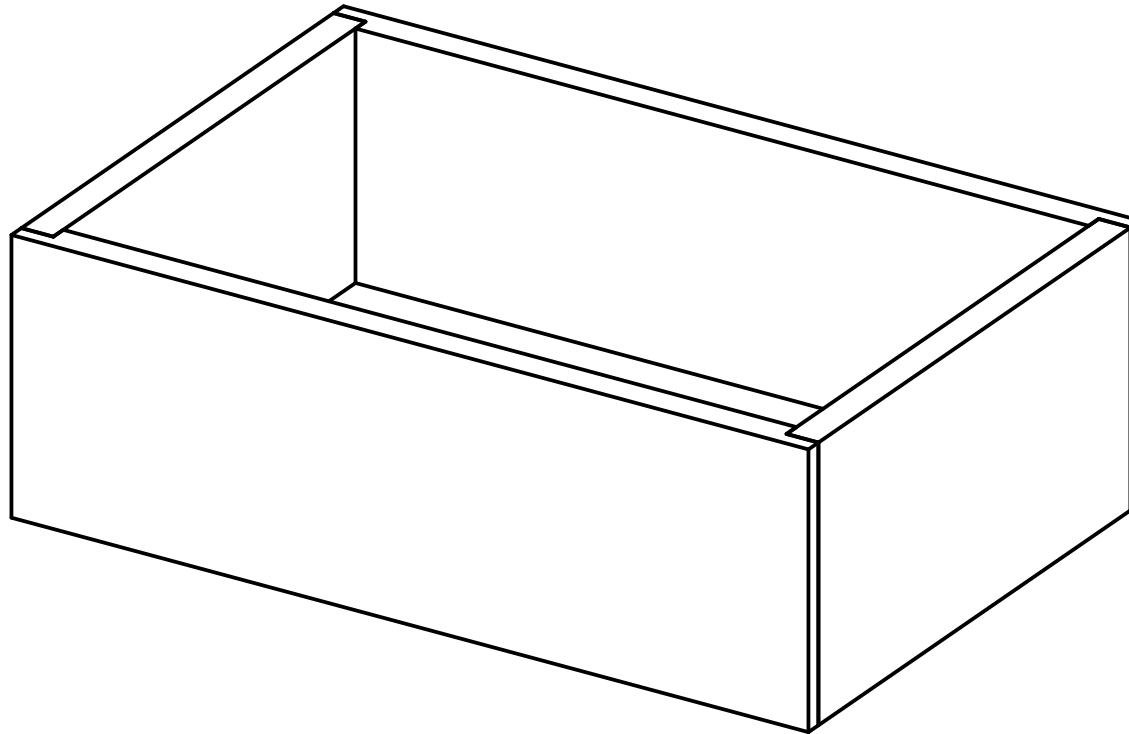


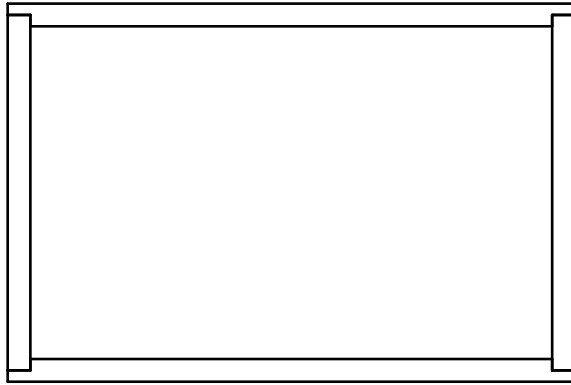
Rabbeted box



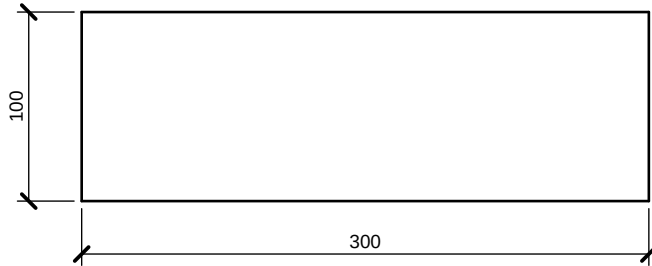
parts	5 (3 different)
overall	300 x 200 x 100 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

Contents

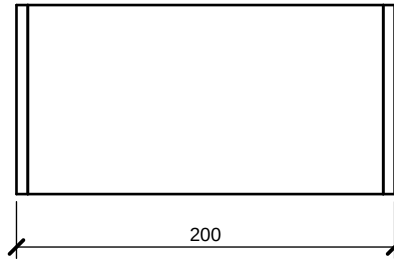
Drawings	3
Assembly	3
Parts	4
1 front_back	5
2 left_right	6
3 bottom	7
Cut list	8
Appendix: source	9



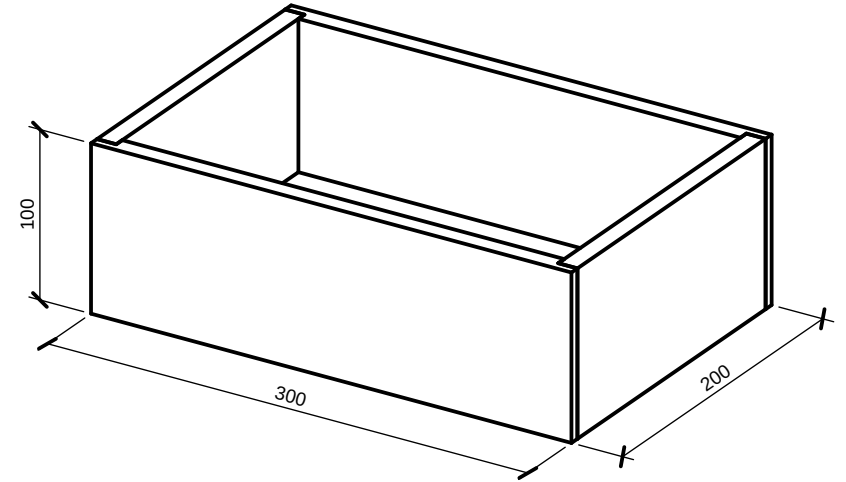
TOP



FRONT

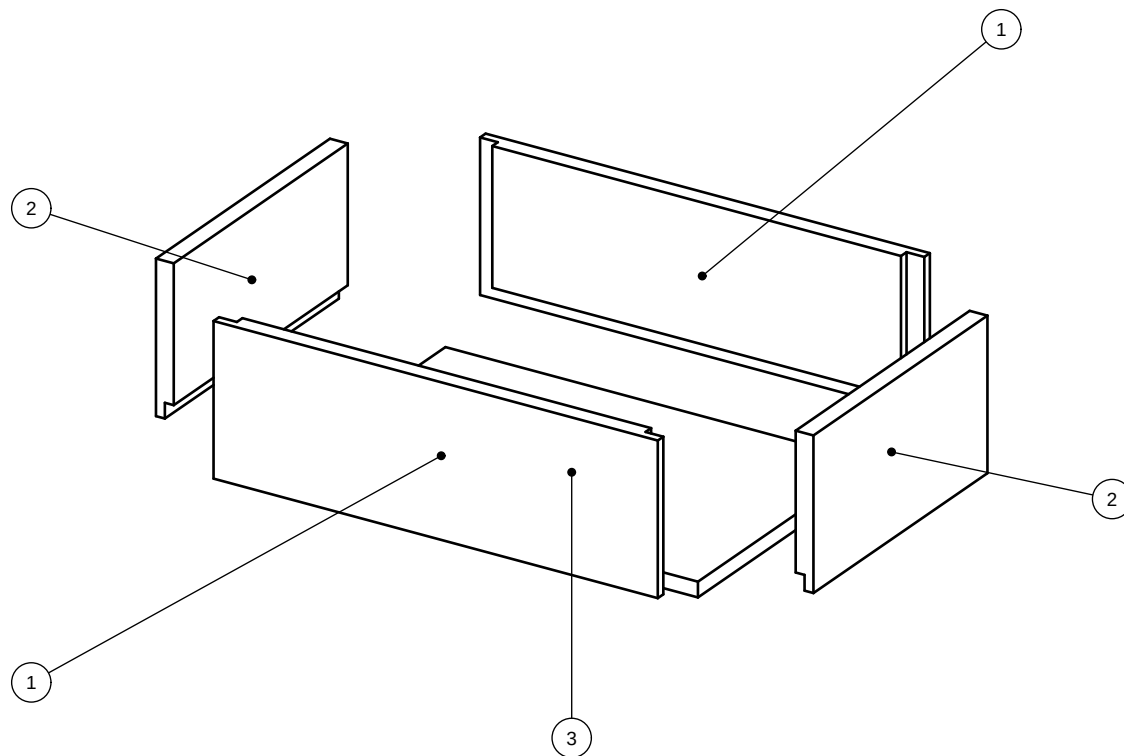


RIGHT SIDE



ISOMETRIC

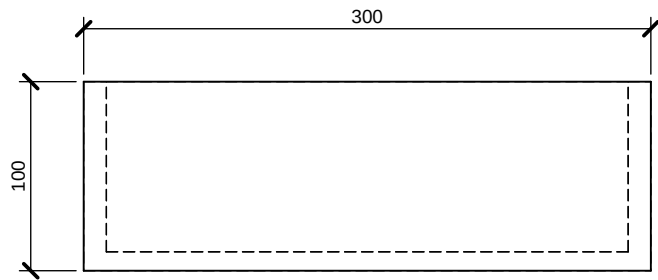
assembly	rabbeted_box
parts	5 (3 different)
overall	300 x 200 x 100 (W x D x H)
material	oak
scale	1:4, mm



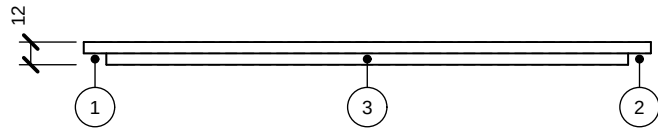
EXPLODED

no.	part	qty	cut size (L x W x T)	material	
1	front_back	2	300 x 100 x 12	oak	
2	left_right	2	188 x 100 x 12	oak	
3	bottom	1	288 x 188 x 10	oak	mirror image

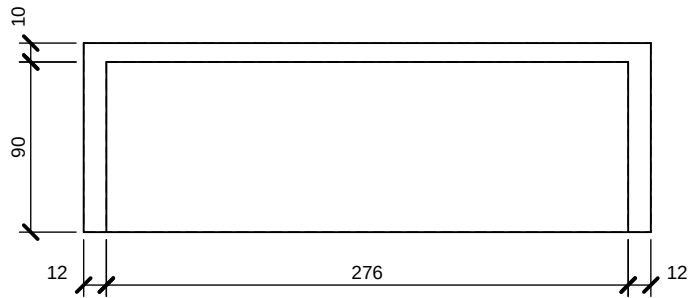
assembly	rabbeted_box
parts	5 (3 different)
overall	300 x 200 x 100 (W x D x H)
material	oak
scale	not to scale, mm



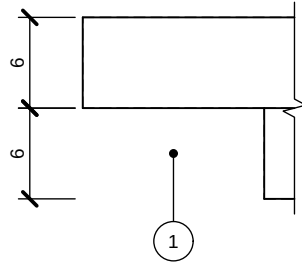
TOP FACE



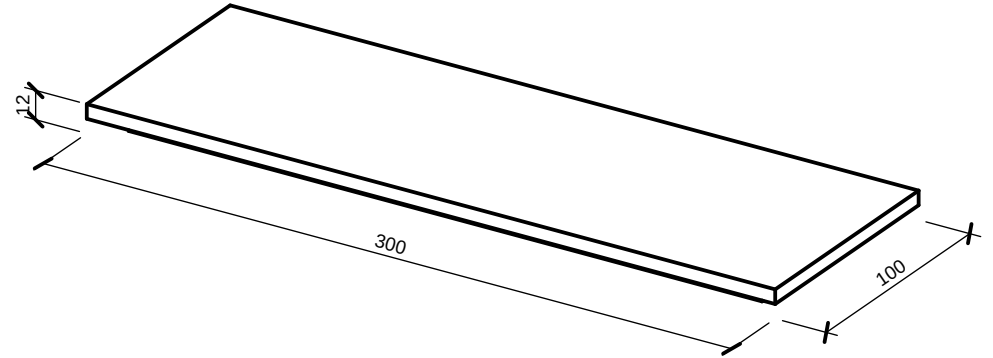
FRONT EDGE



BOTTOM FACE



FRONT EDGE: 1-3 2:1

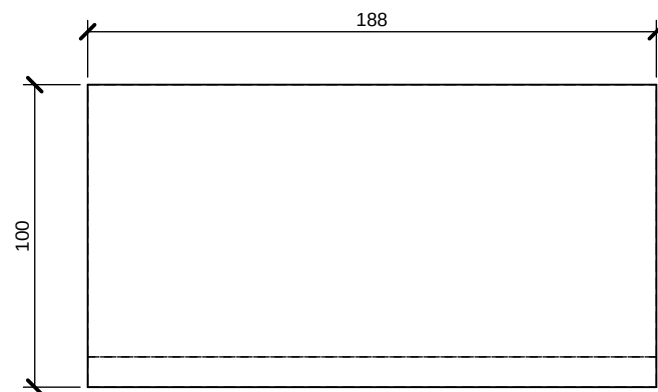


ISOMETRIC

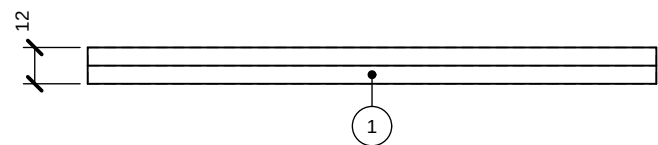
2 pieces: front, back

- 1 end rabbet 12 mm, 6 deep, in bottom_face at left_end
- 2 end rabbet 12 mm, 6 deep, in bottom_face at right_end
- 3 rabbet 10 mm, 6 deep, in bottom_face at front_edge

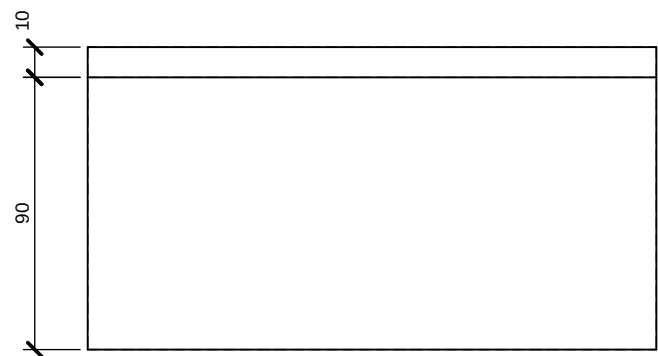
part	front_back
of	rabbeted_box
make	2
cut size	300 x 100 x 12
material	oak
scale	1:4, mm



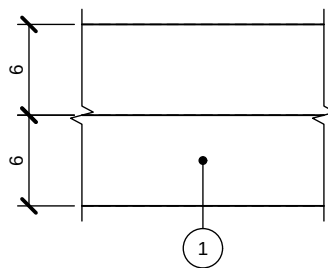
TOP FACE



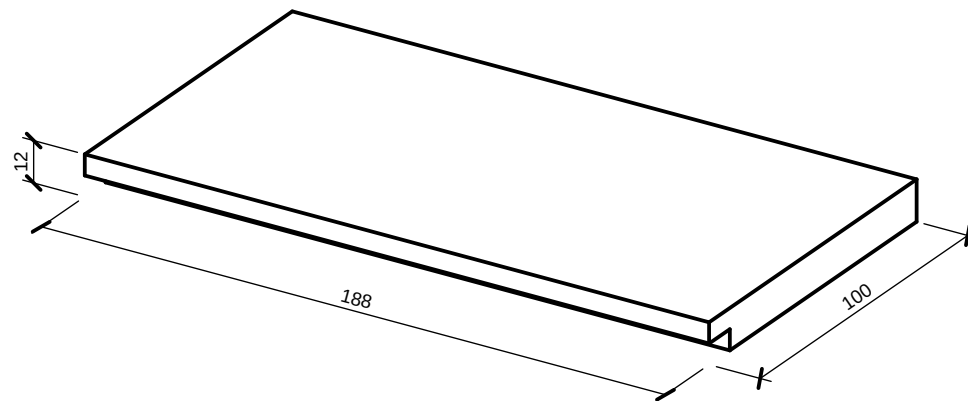
FRONT EDGE



BOTTOM FACE



FRONT EDGE: 1 2:1

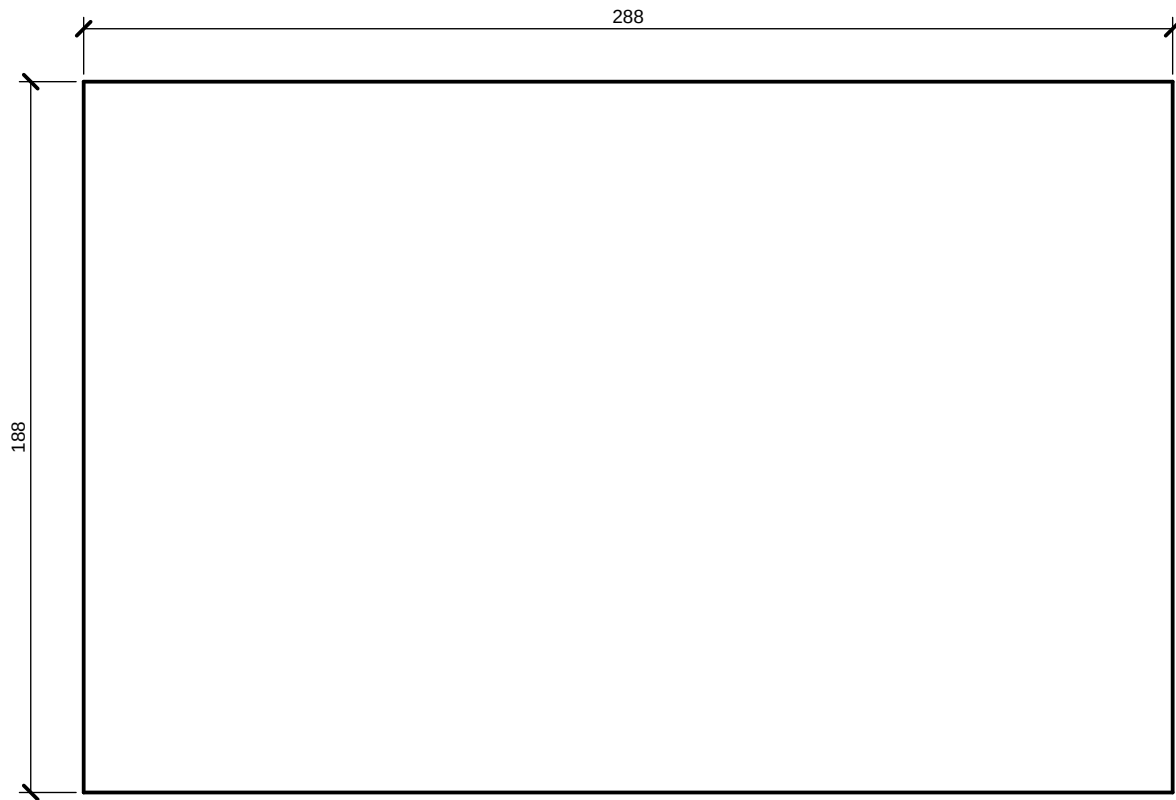


ISOMETRIC

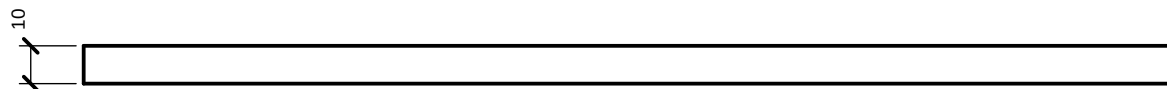
2 pieces: left, right

1 rabbet 10 mm, 6 deep, in bottom_face at front_edge

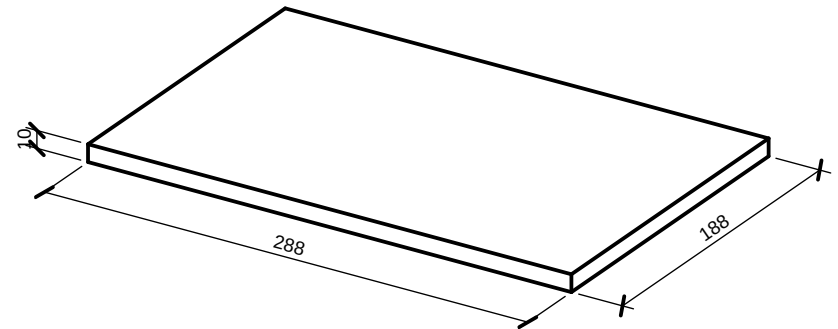
part	left_right
of	rabbeted_box
make	2
cut size	188 x 100 x 12
material	oak
scale	1:2.5, mm



TOP FACE



FRONT EDGE



ISOMETRIC

part	bottom
of	rabbeted_box
make	1
cut size	288 x 188 x 10
material	oak, mirror image
scale	1:2, mm

Cut list

front_back 300 x 100 x 12 mm oak, 2 pieces: front, back
- end rabbet 12 mm, 6 deep, in bottom_face at left_end
- end rabbet 12 mm, 6 deep, in bottom_face at right_end
- rabbet 10 mm, 6 deep, in bottom_face at front_edge
left_right 188 x 100 x 12 mm oak, 2 pieces: left, right
- rabbet 10 mm, 6 deep, in bottom_face at front_edge
bottom 288 x 188 x 10 mm oak, mirror image

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.rabbeted_box
```

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

```
def rabbeted_box(
    length: float = 300,
    depth: float = 200,
    height: float = 100,
    stock: Stock = SIDE_OAK,
    bottom_stock: Stock = THIN_OAK,
) -> Assembly:
    """An open box: the sides' ends in rabbets at the ends of the front and
    the back, the bottom in rabbets along the inside of the walls' bottom
    edges, flush with them; outside sizes.

    The sides and the bottom are as long (and wide) as the inside of the
    box; the rabbets add to them. The back and the right side are mirror
    images of the front and the left side, all with their reference faces
    outside.
    """
    t = stock.thickness
    asm = Assembly("rabbeted_box", root_location=BOX_FRONT)
    front = asm.add(Board("front", stock, length=length, width=height))
    back = asm.add(Board("back", stock, length=length, width=height, mirror=True))
    left = asm.add(Board("left", stock, length=depth - 2 * t, width=height))
    right = asm.add(Board("right", stock, length=depth - 2 * t, width=height, mirror=True))
    # The bottom lies with its reference face down; mirrored, so that its
    # left end is still at the left.
    bottom = asm.add(
        Board("bottom", bottom_stock, length=length - 2 * t, width=depth - 2 * t, mirror=True)
    )
    # The sides' right ends are at the front, so that their front edges are
    # at the bottom, as the other walls' are.
    for side, front_end, back_end in [
```

```

        (left, front.left_end, back.left_end),
        (right, front.right_end, back.right_end),
]:
    asm.join(RabbitJoint(rabbit=front.bottom_face, end=side.right_end, at=front_end))
    asm.join(RabbitJoint(rabbit=back.bottom_face, end=side.left_end, at=back_end))
for wall, edge in [
    (front, bottom.front_edge),
    (back, bottom.back_edge),
    (left, bottom.left_end),
    (right, bottom.right_end),
]:
    # Along the front and the back, the rabbit runs into those at the corners.
    through = wall in (front, back)
    asm.join(
        RabbitJoint(rabbit=wall.bottom_face, end=edge, at=wall.front_edge, through=through)
    )
return asm

```