

3.10.1 Simple Rectangular Plate

Just about the simplest possible example, a rectangular box

```
result = cadquery.Workplane("front").box(2.0, 2.0, 0.5)
```

Api References

- [Workplane\(\)](#) !
- [Workplane.box\(\)](#) !

3.10.2 Plate with Hole

A rectangular box, but with a hole added.

“>Z” selects the top most face of the resulting box. The hole is located in the center because the default origin of a working plane is the projected origin of the last Workplane, the last Workplane having origin at (0,0,0) the projection is at the center of the face. The default hole depth is through the entire part.

```
# The dimensions of the box. These can be modified rather than changing the
# object's code directly.
length = 80.0
height = 60.0
thickness = 10.0
center_hole_dia = 22.0

# Create a box based on the dimensions above and add a 22mm center hole
result = (
    cq.Workplane("XY")
    .box(length, height, thickness)
    .faces(">Z")
    .workplane()
    .hole(center_hole_dia)
)
```

Api References

- [Workplane.hole\(\)](#) !
- [Workplane.box\(\)](#)
- [Workplane.box\(\)](#)

3.10.3 An extruded prismatic solid

Build a prismatic solid using extrusion. After a drawing operation, the center of the previous object is placed on the stack, and is the reference for the next operation. So in this case, the rect() is drawn centered on the previously draw circle.

By default, rectangles and circles are centered around the previous working point.

```
result = cq.Workplane("front").circle(2.0).rect(0.5, 0.75).extrude(0.5)
```

Api References

- `Workplane.circle()` !
- `Workplane.rect()` !
- `Workplane.extrude()` !
- `Workplane()`

3.10.4 Building Profiles using lines and arcs

Sometimes you need to build complex profiles using lines and arcs. This example builds a prismatic solid from 2D operations.

2D operations maintain a current point, which is initially at the origin. Use `close()` to finish a closed curve.

```
result = (
    cq.Workplane("front")
        .lineTo(2.0, 0)
        .lineTo(2.0, 1.0)
        .threePointArc((1.0, 1.5), (0.0, 1.0))
        .close()
        .extrude(0.25)
)
```

Api References

- `Workplane.threePointArc()` !
- `Workplane.lineTo()` !
- `Workplane.extrude()`
- `Workplane()`

3.10.5 Moving The Current working point

In this example, a closed profile is required, with some interior features as well.

This example also demonstrates using multiple lines of code instead of longer chained commands, though of course in this case it was possible to do it in one long line as well.

A new work plane center can be established at any point.

```
result = cq.Workplane("front").circle(
    3.0
) # current point is the center of the circle, at (0, 0)
result = result.center(1.5, 0.0).rect(0.5, 0.5) # new work center is (1.5, 0.0)

result = result.center(-1.5, 1.5).circle(0.25) # new work center is (0.0, 1.5).
# The new center is specified relative to the previous center, not global coordinates!

result = result.extrude(0.25)
```

Api References

- `Workplane.center()` !
- `Workplane()`
- `Workplane.circle()`
- `Workplane.rect()`
- `Workplane.extrude()`

3.10.6 Using Point Lists

Sometimes you need to create a number of features at various locations, and using `Workplane.center()` is too cumbersome.

You can use a list of points to construct multiple objects at once. Most construction methods, like `Workplane.circle()` and `Workplane.rect()`, will operate on multiple points if they are on the stack

```
r = cq.Workplane("front").circle(2.0) # make base
r = r.pushPoints(
    [(1.5, 0), (0, 1.5), (-1.5, 0), (0, -1.5)]
) # now four points are on the stack
r = r.circle(0.25) # circle will operate on all four points
result = r.extrude(0.125) # make prism
```

Api References

- `Workplane.pushPoints()` !
- `Workplane()`
- `Workplane.circle()`
- `Workplane.extrude()`

3.10.7 Polygons

You can create polygons for each stack point if you would like. Useful in 3d printers whose firmware does not correct for small hole sizes.

```
result = (
    cq.Workplane("front")
    .box(3.0, 4.0, 0.25)
    .pushPoints([(0, 0.75), (0, -0.75)])
    .polygon(6, 1.0)
    .cutThruAll()
)
```

Api References

- `Workplane.polygon()` !
- `Workplane.pushPoints()`
- `Workplane.box()`