

dart_pretext

Obstacle-Aware Text Layout · Pure Dart · Any Backend

v1.0.0

dart_pretext is a pure-Dart text layout engine that mathematically computes how text flows around arbitrary geometric obstacles. Unlike Flutter-only solutions, it works in CLI tools, PDF renderers, game engines, and server-side generators - any environment that can run Dart. Bring your own font metrics; dart_pretext handles the rest.

obstacle

Backend Agnostic

PDF, CLI, game engines, servers.

RTL Support

Arabic, Urdu, Hebrew - bidirectional.

Shape Obstacles

Rects, circles, or any bounding box.

Pure Dart

Zero native deps. Runs everywhere.

SHAPE OBSTACLE SHOWCASE

Circle Obstacle

Imagine flowing body copy around a circular author photo or logo.

dart_pretext

computes exact per-line offsets so every line kisses the curve - no manual tweaking required. The filled circle here is a live obstacle bounding box.

Logo

LTR · Centre-Top

Left-Edge Obstacle

LTR · Left Indent

BAR

A tall left-edge block - like a decorative sidebar, chapter mark, or pull-quote bar - forces dart_pretext to indent every overlapping line by the exact width of the obstacle. The result is a clean ragged-left margin that hugs the shape naturally.

RIGHT-TO-LEFT LANGUAGE SUPPORT

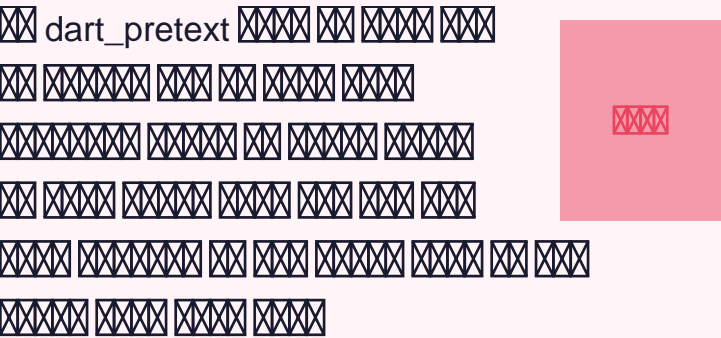
Arabic

RTL - Circle Obstacle



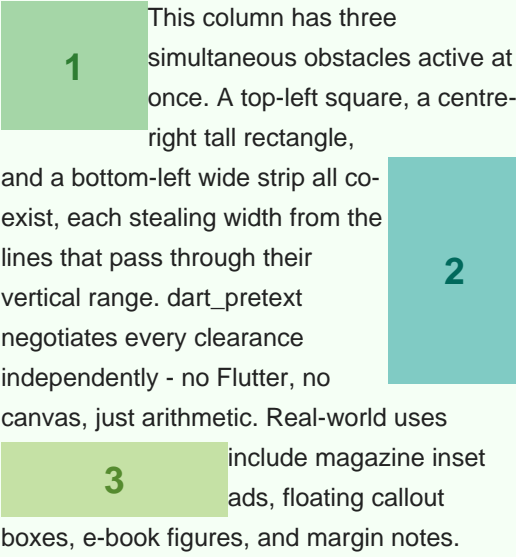
Urdu

RTL - Right-Edge Rect



3 Simultaneous Obstacles

LTR - Multi-Shape



Obstacle Layout: How It Works

```
// 1. Implement Font for your backend
class MyFont extends pfont.Font {
  double measureWidth(String seg) =>
    myFontLib.measure(seg, fontSize);
}
// 2. Prepare once, layout many times
final prepared = prepare(text, MyFont());
// 3. Walk lines with per-line width
LayoutCursor cursor = LayoutCursor(
  segmentIndex: 0, graphemeIndex: 0);
double y = 0;
while (cursor.segmentIndex < prepared.widths.length) {
  final w = computeAvailableWidth(y, obstacles);
  final line = layoutNextLine(prepared, cursor, w);
  if (line == null) break;
  render(line.text, x: indent(y), y: y);
  cursor = line.end; y += lineHeight;
}
```