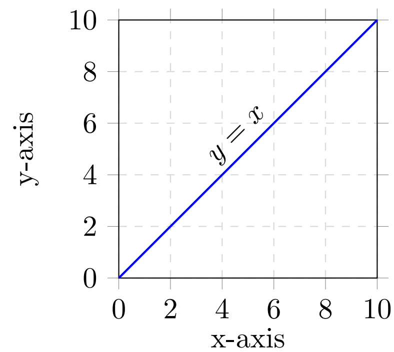
Lines

The *slope-intercept form* of a line in the xy-plane is given by the formula:

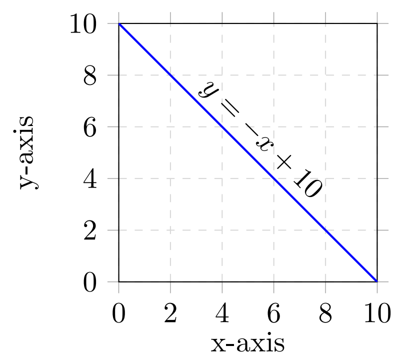
$$y = mx + b$$

where m denotes the *slope* of the line and b denotes the y-intercept. The slope of the line is the change in y divided by the change in x (which is sometimes referred to as “the rise over the run”). A slope of 2 means that the value of y increases at $2\times$ the rate of the value of x. The y-intercept is the value of y when the line crosses the y-axis.

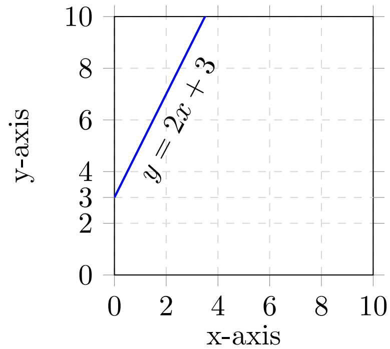
Example: $y = x$**.** In this example, $m = 1$ and $b = 0$.



Example: $y = -x + 10$**.** In this example, $m = -1$ and $b = 10$.



Example: $y = 2x + 3$



Deriving the slope-intercept formula for a line from two points

Let (x_1, y_1) and (x_2, y_2) denote two points in the xy-plane. The slope-intercept formula for the line passing through these two points can be computed as follows.

Step 1: Compute the value of m.

$$m = \frac{y_1 - y_2}{x_1 - x_2}$$

Step 2: Use x_1, y_1, and m to solve for b.

$$y_1 = mx_1 + b$$

This problem is more directly expressed as follows:

$$b = y_1 - mx_1$$

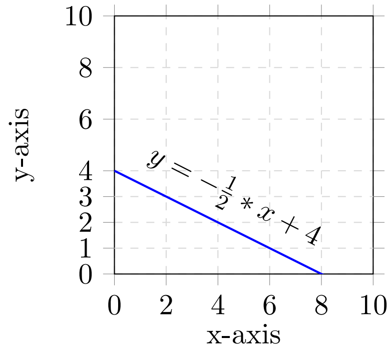
Example: Find the slope-intercept formula for the line passing through $(2, 3)$ and $(6, 1)$.

$$\begin{aligned} m &= \tfrac{3-1}{2-6} \\ &= \tfrac{2}{-4} \\ &= -\tfrac{1}{2} \end{aligned}$$

$$\begin{aligned} b &= 3 - (-\tfrac{1}{2}) * 2 \\ &= 3 + 1 \\ &= 4 \end{aligned}$$

So the slope-intercept formula for this line is:

$$y = -\frac{1}{2} * x + 4$$



Special Cases

There are two special cases of the slope-intercept formula. The first case arises when considering a horizontal line. If we try to derive the slope-intercept formula for two points belonging to a horizontal line we discover that $m = 0$. Thus, the slope-intercept for a horizontal line is of the form: $y = b$.

The second special case arises when considering a vertical line. If we try to derive the slope-intercept formula for two points belonging to a horizontal line we discover that the calculation of m involves a division by zero, which is undefined. Thus, there is no slope-intercept form for a vertical line.

Discrete Considerations: Precision and Accuracy

All cell locations in Bricklayer have integer coordinates. Furthermore, lines in Bricklayer must be modeled as discrete sequences of integer coordinates. This represents a departure from traditional paper and pencil mathematics which involves continuous sequences of real numbered coordinates. The issue at hand is that integer division, which in SML is denoted by *div*, represents an oftentimes coarse approximation to a corresponding division involving real numbers. For example, $1 \; div \; 3 = 0$ whereas $1/3 = 0.3\overline{33}$.

Example. Consider the following slop-intercept formula for a line.

$$y = \frac{1}{4}x + 0$$

Integer-based Calculation	Real-valued Calculation
$m = 1 \; div \; 4$	$m = 1.0/4.0$
$= 0$	$= 0.25$

x (input)	y (integer-value calculation)	y (real-value calculation)	**Rounding the real-valued y to the nearest integer**
0	0	0.0	0
1	0	0.25	0
2	0	0.5	1
3	0	0.75	1
4	1	1.0	1
5	1	1.25	1
6	1	1.5	2
7	1	1.75	2
8	2	2.0	2
9	2	2.25	2
10	2	2.5	3
11	2	2.75	3
12	3	3.0	3
13	3	3.25	3
14	3	3.5	4
15	3	3.75	4

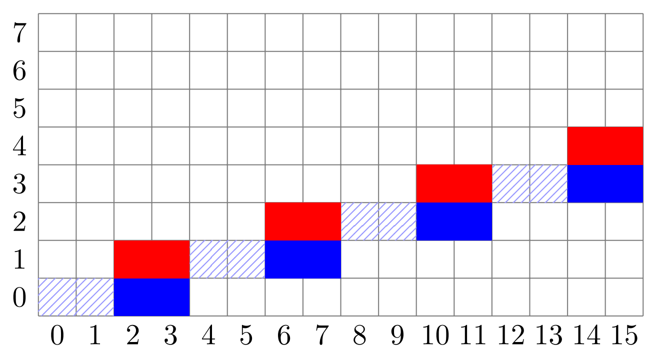


Figure 1: A discrete representation of the formula $y = \frac{1}{4}x + 0$. The blue squares result from integer-based division, red squares result from real number-based division, and textured squares are locations where integer and real number divisions yield the same results.

Roundoff errors in integer division can be compensated for in one of two ways. In the first approach the integer division operation is delayed for as long as mathematically possible in the calculation of the slope-intercept formula for a line.

Example. Using the formula shown below, suppose we want to find the value of y when $x = 8$.

$$y = \frac{1}{4}x + 0$$

The division operation can be mathematically delayed as follows:

$$
\begin{aligned}
y &= (1 \ div \ 4) * 8 + 0 \\
&= (1 * 8) \ div \ 4 + 0 \\
&= 8 \ div \ 4 + 0 \\
&= 2
\end{aligned}
$$

The other possibility is to (1) convert all integer values to real values, (2) perform a real-valued evaluation, and (3) convert the result back to an integer.

Conversion Function	**Description**
Real.fromInt	Converts an integer value to a real value.
Real.round	Rounds a real value to the nearest integer.

Function Call	Result of Evaluation
Real.fromInt 5	5
Real.round 5.4	5
Real.round 5.5	6

Example. Using the formula shown below, suppose we want to find the value of y when $x = 8$.

$$y = \frac{1}{4}x + 0$$

By using conversion, the integer division operation can be mathematically avoided as follows:

$$\begin{aligned} y &= (Real.fromInt\ 1 / Real.fromInt\ 4) * 8.0 + 0.0 \\ &= 0.25 * 8.0 + 0.0 \\ &= 2.0 \\ &= Real.round\ 2.0 \\ &= 2 \end{aligned}$$

Note that in SML, integer values and real values are syntactically distinct from one another. In particular, all real values must have a decimal point – even the value zero.