Title: `sample_code`

John Doe

December 13, 2025

Some text and some `code`.

- item 1
- item 2
- item 3

```
x = 1
# a comment
y = x+1
## this is another comment
```

$$\begin{aligned} a &= b + c \\ &\quad + d + e \\ f &= g - h \end{aligned}$$

As shown in Equation, Euler's identity is a fundamental mathematical relationship.

1 Einstein's Equation `E=m c^2`

This is markdown. Consider:

$$E = mc^2 \tag{1}$$

Where Equation (1) demonstrates the relationship between energy `e`, mass `m`, and the speed of light `c`.

```
m = 1

c = 3e8

E = m*c^2
```

Another multiline equation is as follows:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \tag{2}$$

Here, Equation (2) represents a system of linear equations with non-negativity constraints.

Here is a table:

Algorithm	$f(x^*)$	Time (s)
PDHG	1.234	0.07
B&B	1.230	3.12

Another table:

Field	Type	Mathematics	Implementation detail	Role in the package
`_id`	`Int`	Pure identifier; no direct math meaning	Incremented from global `NEXT_ID[]`	Stable identity for hashing, comparisons, dictionary keys
`_is_leaf`	`Bool`	Indicates whether this is a fundamental vector in the Gram basis	`true` for leaf points, `false` for linear combinations	Determines Gram dimensioning and whether `.counter` is set
`decomposition_dict`	`OrderedDict{Point,Float64}`	Stores coefficients of the linear form $X = \sum_i \alpha_i P_i$	For a leaf, set to `{self ⇒ 1.0}`; for a composite, the sparse coefficient map	Drives conversion of inner products to linear forms over G
`counter`	`Union{Int,Nothing}`	Index i of the leaf in the Gram basis (only for leaves)	Set to `Point_counter[]` at leaf creation, `nothing` otherwise	Used to size/index the Gram matrix and build evaluation vectors
`_value`	`Union{Vector{Float64},Nothing}`	Numeric value of the vector after solving the PEP	`nothing` until `solve!` writes results back	Enables `eval` to return a concrete vector after solve