

```
1  # derivate.ex
2  defmodule Derivate do
3      @type constant() :: {:const, number()} | {:const, atom()}
4      @type literal() :: {:const, number()} | {:const, atom()}
5      • | {:var, atom()}
6      @type expr() :: {:add, expr(), expr()} | {:mul, expr(),
7      • expr()} | literal()
8
9      def deriv({:const, _}, _) do {:const, 0} end
10     def deriv({:var, v}, v) do {:const, 1} end
11     def deriv({:var, y}, _) do {:var, y} end
12     def deriv({:mul, e1, e2}, v) do
13         • {:add, {:mul, deriv(e1, v), e2}, {:mul, e1, deriv(e2,
14         v)}}
15     end
16     def deriv({:exp, {:var, v}, {:const, c}}, v) do
17         • {:mul, {:const, c}, {:exp, {:var, v}, {:const, c - 1}}}
18     end
19     def deriv({:add, e1, e2}, v) do
20         • {:add, deriv(e1, v), deriv(e2, v)}
21     end
22 end
```