


```

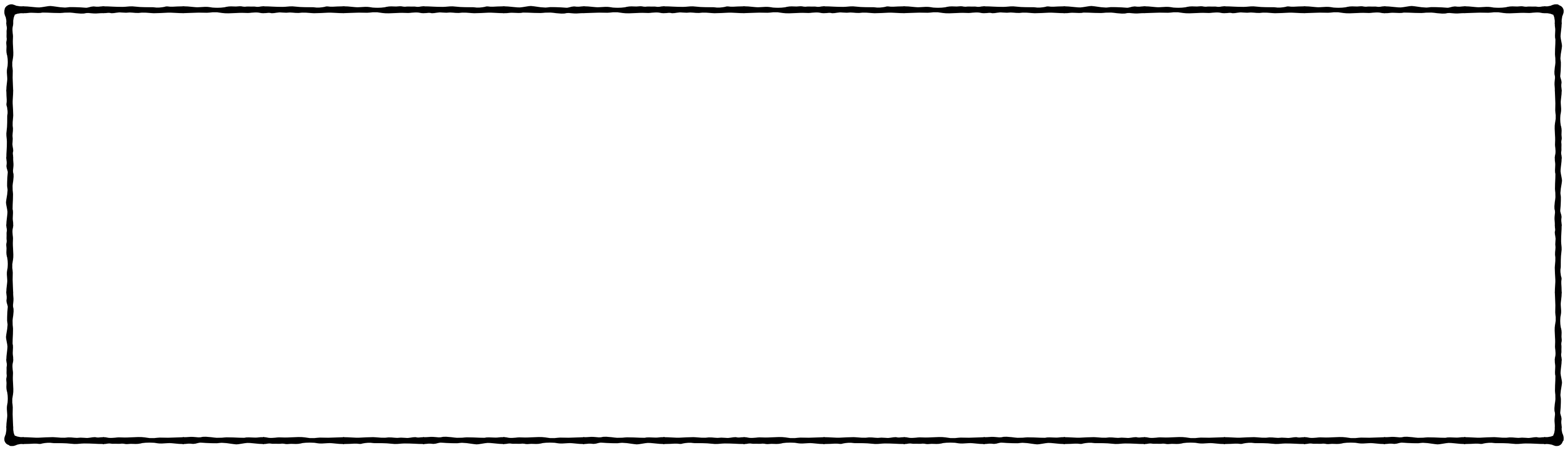
instance Generic (Tree a) where
  type Rep (Tree a) =
    D1 ('MetaData "Tree" "Main" "package-name" 'False)
      (C1 ('MetaCons "Leaf" 'PrefixI 'False)
        (S1 ('MetaSel 'Nothing
              'NoSourceUnpackedness
              'NoSourceStrictness
              'DecidedLazy)
          (Rec0 a)))

  ::+::
  C1 ('MetaCons "Node" 'PrefixI 'False)
    (S1 ('MetaSel 'Nothing
              'NoSourceUnpackedness
              'NoSourceStrictness
              'DecidedLazy)
        (Rec0 (Tree a)))

  ::*::
  S1 ('MetaSel 'Nothing
        'NoSourceUnpackedness
        'NoSourceStrictness
        'DecidedLazy)
    (Rec0 (Tree a)))

```

...



GHG.Generics

```
data Tree a = Leaf a | Node (Tree a) (Tree a)
```

```
deriving Generic
```

From H.C. Gerdunentation



GHC.Generics

From GHC.Generics documentation

```
data Tree a = Leaf a | Node (Tree a) (Tree a)  
    deriving Generic
```

GHC.Generics

From GHC.Generics documentation

```
data Tree a = Leaf a | Node (Tree a) (Tree a)
  deriving Generic
```

```
instance Generic (Tree a) where
  type Rep (Tree a) =
    Rec0 a
  :+:
  (Rec0 (Tree a) :* Rec0 (Tree a))
```