

Name-directed

The field named "a"

↓

```
data T = T { a :: Int  
            , b :: String  
            , c :: Bool }
```

```
field @"a" :: Lens T Int
```

↙

```
Int
```

↘

```
String × Bool
```

Implementation: `field`

```
class HasField (field :: Symbol) a s | s field → a where  
  field :: Lens s a
```

```
instance  
  (Generic s  
   ,ErrorUnless field s (HasTotalFieldP field (Rep s))  
   ,GHasField field (Rep s) a  
  ) ⇒ HasField field a s where  
  field = ... gfield ...
```