## type PackageType = Infer<typeof schema>

```
/*
const schema: Fuji<"shape", {
    name: Fuji<"string" | "required", string>;
    version: Fuji<"pattern", string>;
    workspaces: Fuji<"array-of", string[]>;
    repository: Fuji<"shape", {
        type: Fuji<...>;
        url: Fuji<...>;
    }>;
}>
```

```
/*
{
  name: string,
  version?: string,
  workspaces?: string[],
  repository?: {
    type: string | null,
    url?: string
  }
}
*/
```

```
export type Infer<FujiSchema> = FujiSchema extends Fuji<
  infer $RuleType,
  infer Value
  ? Value extends AnyShapeSchema
    ? InferRecord<Value>
    : Value extends Array<AnyShapeSchema>
    ? InferArrayOfRecords<Value>[]
    : RequiredType extends $RuleType
    ? NullableType extends $RuleType
      ? Value | null
      : Value
    : DefaultToType extends $RuleType
    ? Value
    : NullableType extends $RuleType
    ? Value | null
    : Value | undefined
  : never
type InferArrayOfRecords<Value extends Array<AnyShapeSchema>> = ...
type InferRecord<Shape extends Record<string, Fuji<any, any>>> = ...
type RequiredKeys<T extends AnyShapeSchema> = ...
type OptionalKeys<T extends AnyShapeSchema> = ...
```