

```
import { f, string, array, required, number, arrayOf, run } from '@winexy/fuji'

function test_fuji(value: unknown) {
  const plugin = f(string(), required())

  const schema = f.shape({
    name: f(string(), required()),
    version: f(number(), required()),
    optional: f(string()),
    plugins: f(array(), arrayOf(plugin), required()),
  })

  const result = run(schema, value)

  if (result.invalid) {
    return
  }

  result.value // { name: string, version: number, optional?: string, plugins: string[] }
}
```

```
import { Rule } from '../types'

// string.ts
export type StringType = 'string'

export const string = (msg?: string): Rule<StringType, string> => ({
  type: 'string',
  ...,
  func(ctx) {...}
})

// required.ts
export type RequiredType = 'required'

export const required = <Value extends any = any>(
  msg?: string
): Rule<RequiredType, Value> => ({
  type: 'required',
  ...,
  func(ctx) {...}
})
```