Structure

FloatingPointParseStrategy

A parse strategy for creating floating-point values from formatted strings.

iOS 15.0+ | iPadOS 15.0+ | Mac Catalyst 15.0+ | macOS 12.0+ | tvOS 15.0+ | visionOS 1.0+ | watchOS 8.0+

```
struct FloatingPointParseStrategy<Format> where Format : Format
Style, Format.FormatInput : BinaryFloatingPoint
```

Overview

Create an explicit `FloatingPointParseStrategy` to parse multiple strings according to the same parse strategy. In the following example, `usCurrencyStrategy` is a `FloatingPoint ParseStrategy` that uses US dollars and the `en_US` locale's conventions for number formatting. The example then uses this strategy to parse an array of strings, some of which represent valid US currency values.

```
let usCurrencyStrategy: FloatingPointParseStrategy =
FloatingPointFormatStyle<Double>.Currency(code: "USD",
                                          locale: Locale(identifier: "en_US"))
    .parseStrategy
let currencyValues = ["$100.11", "$1,000.22", "$10,000.33", "€100.44"]
let parsedValues = currencyValues.map { try? usCurrencyStrategy.parse($0) } // [Opti
```

You don't need to instantiate a parse strategy variable to parse a single string. Instead, use the `BinaryFloatingPoint` initializers that take a source `String` and a `format` parameter to parse the string according to the provided `FormatStyle`. The following example parses a string that represents a currency value in US dollars.

```
let formattedUSDollars = "$1,234.56"
let parsedUSDollars = try? Double(formattedUSDollars, format: .currency(code: "USD")
    .locale(Locale(identifier: "en_US"))) // 1234.56
```

Topics

Creating a floating-point parse strategy

`init<Value>(format: Format, lenient: Bool)`

Creates a parse strategy instance using the specified floating-point format style.

`init<Value>(format: Format, lenient: Bool)`

Creates a parse strategy instance using the specified floating-point currency format style.

`init<Value>(format: Format, lenient: Bool)`

Creates a parse strategy instance using the specified floating-point percentage format style.

Accessing strategy properties

`var formatStyle: Format`

The format style this strategy uses when parsing strings.

`var lenient: Bool`

A Boolean value that indicates whether parsing allows any discrepencies in the expected format.

Relationships

Conforms To

`Copyable`
`Decodable`
`Encodable`
`Equatable`
`Hashable`
`ParseStrategy`

Conforms when `Format` conforms to `FormatStyle` and `Format.FormatInput` conforms to `BinaryFloatingPoint`.

`Sendable`

`SendableMetatype`

See Also

Data parsing in Swift

`protocol ParseableFormatStyle`

A type that can convert a given input data type into a representation in an output type.

`protocol ParseStrategy`

A type that parses an input representation, such as a formatted string, into a provided data type.

`struct IntegerParseStrategy`

A parse strategy for creating integer values from formatted strings.

`struct ParseStrategy`

A parse strategy for creating decimal values from formatted strings.