Class

NSPredicate

A definition of logical conditions for constraining a search for a fetch or for in-memory filtering.

iOS 3.0+ | iPadOS 3.0+ | Mac Catalyst 13.1+ | macOS 10.4+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
class NSPredicate
```

Overview

Predicates represent logical conditions, which you can use to filter collections of objects. Although it's common to create predicates directly from instances of `NSComparisonPredicate`, `NSCompoundPredicate`, and `NSExpression`, you often create predicates from a format string that the class methods parse on `NSPredicate`. Examples of predicate format strings include:

- Simple comparisons, such as `grade == "7"` or `firstName like "Juan"`
- Case- and diacritic-insensitive lookups, such as `name contains[cd] "stein"`
- Logical operations, such as `(firstName like "Mei") OR (lastName like "Chen")`
- Temporal range constraints, such as `date between {$YESTERDAY, $TOMORROW}`
- Relational conditions, such as `group.name like "work*"`
- Aggregate operations, such as `@sum.items.price < 1000`

For a complete syntax reference, refer to the Predicate Programming Guide.

You can also create predicates that include variables using the `evaluate(with:substitutionVariables:)` method so that you can predefine the predicate before substituting concrete values at runtime.

Topics

Creating a Predicate

`init(format: String, argumentArray: [Any]?)`

Creates a predicate by substituting the values in a specified array into a format string and parsing the result.

`init(format: String, arguments: CVaListPointer)`

Creates a predicate by substituting the values in an argument list into a format string and parsing the result.

`convenience init(format: String, any CVarArg...)`

Creates a predicate by substituting the values in an argument list into a format string and parsing the result.

`convenience init?<Input>(Predicate<Input>)`

Creates a predicate by converting an existing predicate.

`func withSubstitutionVariables([String : Any]) -> Self`

Returns a copy of the predicate and substitutes the predicates variables with specified values from a specified substitution variables dictionary.

`init(value: Bool)`

Creates and returns a predicate that always evaluates to a specified Boolean value.

`init(block: (Any?, [String : Any]?) -> Bool)`

Creates a predicate that evaluates using a specified block object and bindings dictionary.

`init?(fromMetadataQueryString: String)`

Creates a predicate with a metadata query string.

Evaluating a Predicate

`func evaluate(with: Any?) -> Bool`

Returns a Boolean value that indicates whether the specified object matches the conditions that the predicate specifies.

```
func evaluate(with: Any?, substitutionVariables: [String : Any]?) ->
Bool
```

Returns a Boolean value that indicates whether the specified object matches the conditions that the predicate specifies after substituting in the values from a specified variables dictionary.

```
func allowEvaluation()
```

Forces a securely decoded predicate to allow evaluation.

Getting a String Representation

```
var predicateFormat: String
```

The predicate's format string.

Relationships

Inherits From

`NSObject`

Inherited By

`NSComparisonPredicate`, `NSCompoundPredicate`

Conforms To

```
CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSCoding
NSCopying
NSObjectProtocol
NSSecureCoding
```

See Also

Filltering

`struct Predicate`

A logical condition used to test a set of input values for searching or filtering.

`struct PredicateError`

An error thrown while evaluating a predicate.

`struct PredicateCodableConfiguration`

A specification of the expected types and key paths found in an archived predicate.

`protocol PredicateCodableKeyPathProviding`

A type that provides the expected key paths found in an archived predicate.

`protocol PredicateExpression`

A component expression that makes up part of a predicate.

`protocol StandardPredicateExpression`

A component expression that makes up part of a predicate, and that's supported by the standard predicate type.

`enum PredicateExpressions`

The expressions that make up a predicate.

`struct PredicateBindings`

A mapping from a predicates's input variables to their values.

`class NSExpression`

An expression for use in a comparison predicate.

`class NSComparisonPredicate`

A specialized predicate for comparing expressions.

`class NSCompoundPredicate`

A specialized predicate that evaluates logical combinations of other predicates.