

THE POWER OF PREDICATES

$(\text{YES} == \text{NO}) == \text{NO}$

OVERVIEW

- ✻ Predicates

- ✻ Foundation

 - ✻ structure, expressions, examples

- ✻ AppKit

 - ✻ predicate editors, row templates,
localization

PREDICATES

PREDICATES

- ☀ Predicate: A statement that evaluates to true or false

- ☀ Boolean Algebra

- ☀ $\neg(A \vee B) \wedge (C \wedge A)$

- ☀ 3SAT: At the heart of $P \stackrel{?}{=} NP$

PREDICATES

$$\odot 1 + 1 = 2 \Rightarrow \text{true}$$

$$\odot \sin(\pi) = 0 \Rightarrow \text{true}$$

$$\odot e^{i\pi} + 1 = 0 \Rightarrow \text{true}$$

PREDICATES

☀ The sky is blue \Rightarrow ?

☀ $6 \times 9 = 42 \Rightarrow$?

☀ You are paying attention \Rightarrow ?

SO WHAT...?

- ✱ Predicates are everywhere in code
- ✱ `if(predicate statement)...`
- ✱ `while(predicate statement)...`
- ✱ But we know about these

PREDICATES & FOUNDATION

FOUNDATION

- ✱ NSPredicate: your new best friend
- ✱ Represents predicates in a syntax tree
- ✱ Parses strings into a tree predicate objects
 - ✱ NSComparisonPredicate
 - ✱ NSCompoundPredicate

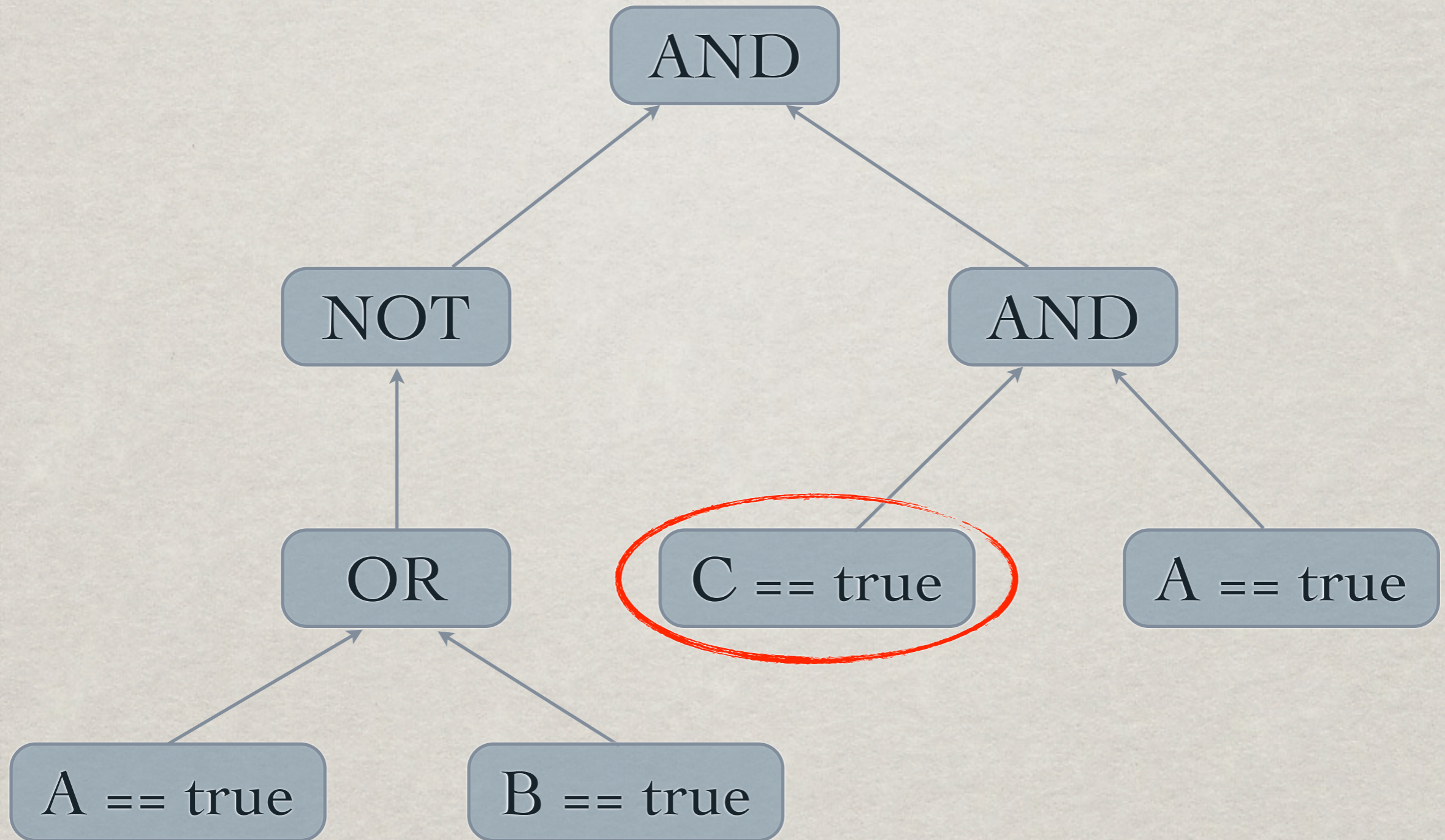
NSCOMPOUNDPREDICATE

- ✻ Any parent node in a predicate tree
- ✻ Represents logical AND, OR, or NOT
- ✻ Has an array of subpredicates

NSCOMPARISONPREDICATE

- ✱ Any leaf node in a predicate tree
- ✱ Has 3 main readonly properties
 - ✱ leftExpression (NSExpression)
 - ✱ predicateOperatorType (enum)
 - ✱ rightExpression (NSExpression)

$$\neg(A \vee B) \wedge (C \wedge A)$$



NSEXPRESSION

- ✻ Represents a single (evaluated) value
- ✻ NSExpressions can represent:
 - ✻ self, blocks, constant values, functions (with arguments), key paths, set operations, variables, filtered collections

CREATION SYNTAX

- ✻ + [NSPredicate predicateWithFormat:]
- ✻ + [NSComparisonPredicate
predicateWithLeftExpression:
rightExpression:...]
- ✻ + [NSCompoundPredicate (or|and|not)
PredicateWithSubpredicates:]

PREDICATE GOTCHAS

- ✱ Format strings support basic syntax
 - ✱ Also, “%K”
- ✱ Equal format strings \neq equal predicates
- ✱ Not all predicates are serializable
- ✱ -hash is usually not useful

EXAMPLES

BASIC

```
NSArray * objects = ...
```

```
NSArray * filtered = [objects  
    filteredArrayUsingPredicate:[NSPredicate  
    predicateWithFormat:@"property = 42"]];
```

VARIABLES

```
NSPredicate * template = [NSPredicate  
    predicateWithFormat:@"property = $SEARCH OR  
    relationship.property = $SEARCH"];
```

```
NSDictionary * substitute = [NSDictionary  
    dictionaryWithObject:@"foo" forKey:@"SEARCH"];
```

```
NSPredicate * filter = [template  
    predicateWithSubstitutionVariables:substitute];
```

```
NSArray * objects = ...
```

```
NSArray * filtered = [objects  
    filteredArrayUsingPredicate:filter];
```

COMPOUND

```
NSPredicate * template = [NSPredicate predicateWithFormat:@"property =
    $SEARCH OR relationship.property = $SEARCH"];
NSArray * searchTerms = ...
NSMutableArray * subpredicates = [NSMutableArray array];

for (NSString * searchTerm in searchTerms) {
    [subpredicates addObject:[template
        predicateWithSubstitutionVariables:[NSDictionary
            dictionaryWithObject:searchTerm forKey:@"SEARCH"]]]
}

NSPredicate * filter = [NSCompoundPredicate
    andPredicateWithSubpredicates:subpredicates];

NSArray * objects = ...
NSArray * filtered = [objects filteredArrayUsingPredicate:filter];
```

FUNCTIONS

```
NSPredicate * p = [NSPredicate  
    predicateWithFormat:@"42 = 6 * 9"];
```

```
p = [NSPredicate predicateWithFormat:@"MAX(1, 5, 10) =  
10];
```

MORE FUNCTIONS

```
NSPredicate * p = [NSPredicate  
    predicateWithFormat:@"sin(%f) = 1", M_PI/2];
```

```
//Crash!
```

MORE FUNCTIONS

```
@interface NSNumber (TrigFunctions)

- (NSNumber *) sinValue

@end

@implementation NSNumber (TrigFunctions)

- (NSNumber *) sinValue {
    return [NSNumber numberWithDouble:sin([self
doubleValue])];
}

@end

NSPredicate * p = [NSPredicate predicateWithFormat:@"FUNCTION(%f, %@) =
1", M_PI/2, @"sinValue"];
```

COLLECTIONS

zip	city	state	phone
02871	Portsmouth	RI	401.....
55303	Anoka	MN	763.....
92833	Fullerton	CA	701.....

```
NSArray * csv = ... //array of arrays of strings
```

```
NSPredicate * p = [NSPredicate predicateWithFormat:@"SELF[2] = %@",  
    @"RI"];
```

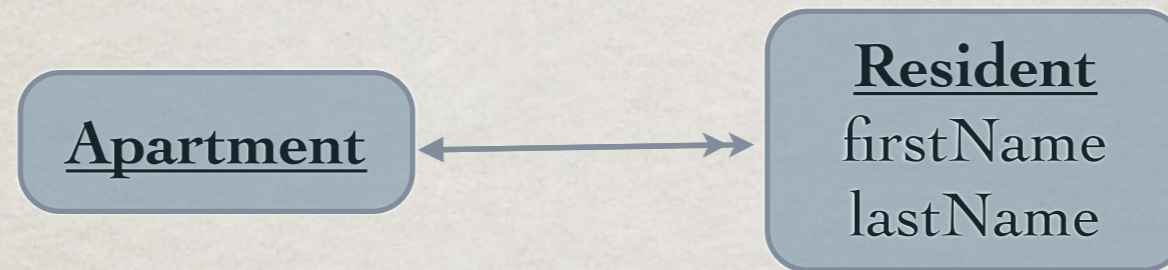
```
NSArray * filtered = [csv filteredArrayUsingPredicate:p];
```

```
csv = ... //array of dictionaries
```

```
p = [NSPredicate predicateWithFormat:@"SELF[state] = %@", @"RI"];  
//p = [NSPredicate predicateWithFormat:@"state = %@", @"RI"];
```

```
NSArray * filtered = [csv filteredArrayUsingPredicate:p];
```

SUBQUERY



```
NSArray * apartments = ...
```

```
NSPredicate * p = [NSPredicate predicateWithFormat:@"SUBQUERY(residents,  
    $r, $r.firstName = %@ AND $r.lastName = %@).@count > 0", @"Steve",  
    @"Jobs"];
```

```
NSArray * filtered = [apartments filteredArrayUsingPredicate:p];
```


PREDICATES & APPKIT

NSPREDICATEEDITOR

- ✱ A visual representation of a predicate
- ✱ Fairly common in productivity apps
 - ✱ iTunes smart playlists
 - ✱ Mail.app rules
 - ✱ Spotlight
- ✱ Relatively straightforward, until you want to customize

NSPREDICATEEDITOR

- ✱ Predicate set and retrieved via “objectValue”
- ✱ Based on “row templates”
- ✱ Each template can display certain predicates
 - ✱ Somewhat limited in visual representation
- ✱ Quite customizable through code

NSPREDICATEEDITORROWTEMPLATE

- ✻ Key methods to override

- ✻ -templateViews

- ✻ -predicateWithSubpredicates:

- ✻ -setPredicate:

- ✻ -matchForPredicate:

LOCALIZATION

- ✱ Localization is (essentially) undocumented
- ✱ Uses a special syntax of NSLocalizedString
- ✱ Requires setting some options in code
- ✱ NSLocalizedString(@"%[left]@ %[is]@
%@" , ...)

DEMO

OVERVIEW

- ✱ Predicates are everywhere
- ✱ NSPredicate is awesome
- ✱ NSPredicateEditor is awesome*

QUESTIONS?