

# Objective C Tutorial

컨텐츠검색플랫폼팀 조영일

# 목차

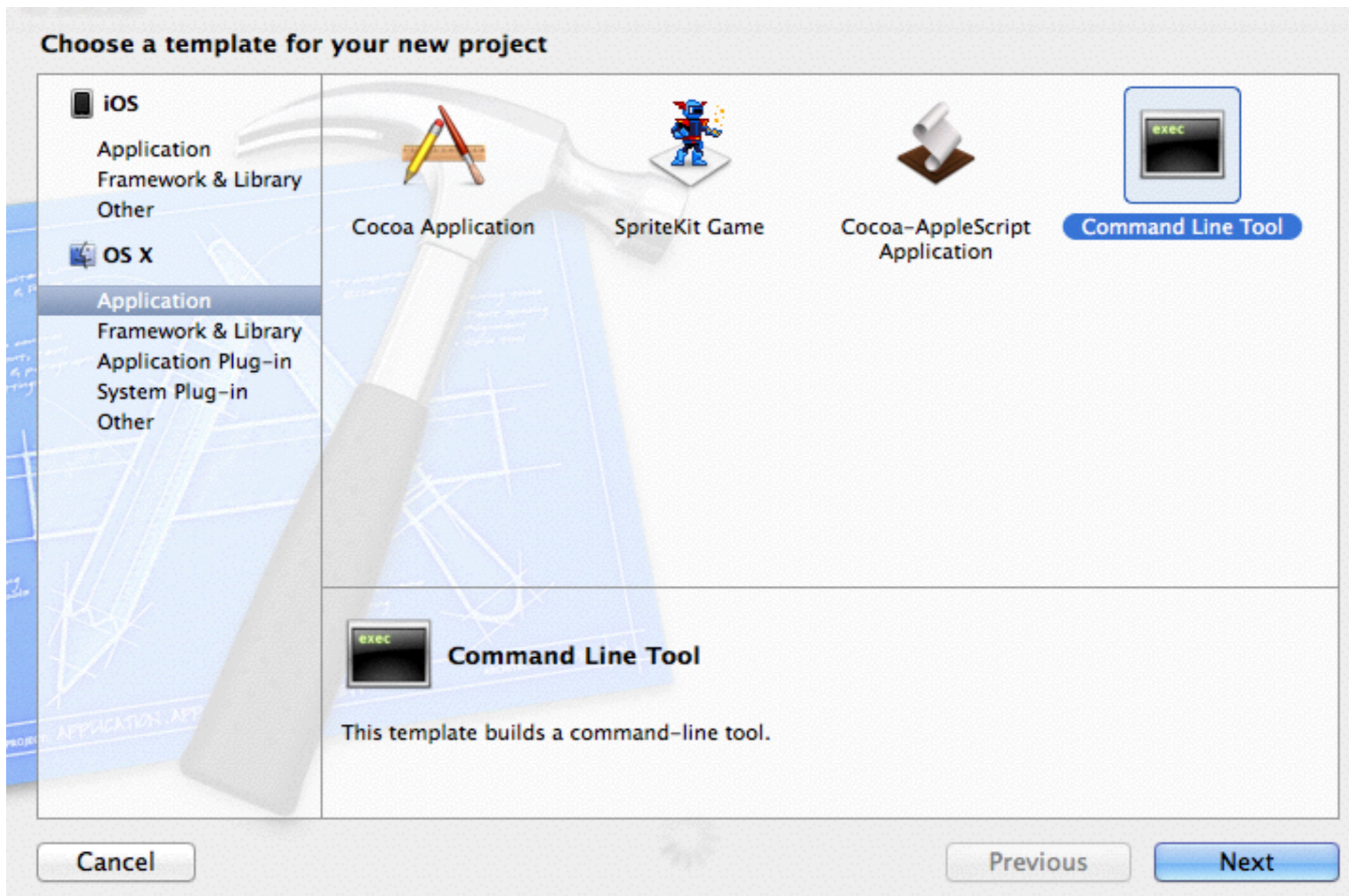
- 소개
- 기초 문법
- 문자열
- 제어문
- Message & Method
- Class
- 예외

# 소개

- 역사
  - 1983, Tom Love & Brad Cox, StepStone
  - Smalltalk의 영향
  - 1988, NeXT가 StepStone에서 Object C 라이선스 확보, NeXTSTEP GUI 개발 언어로 사용됨
  - 1996, Apple이 NeXT 합병, OpenStep을 기반으로 한 Mac OS X 출시
- 언어적 특징
  - C의 확장 (추가 라이브러리)
  - 메시지 전달
  - dynamic typing, class like prototype in Javascript
  - interface+implementation: class in Java
  - protocol: interface in Java

# 실습준비

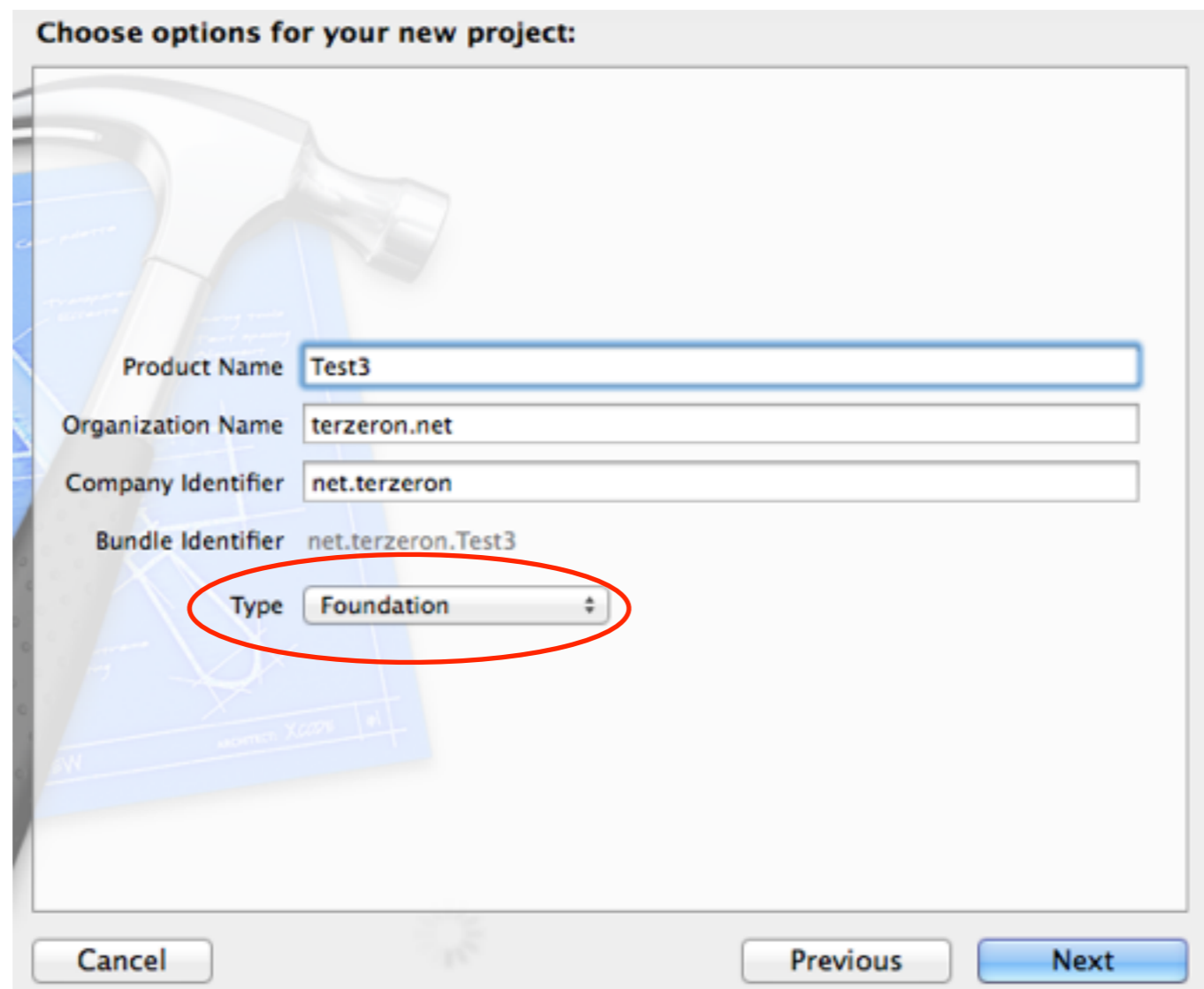
- Xcode를 실행하고 신규 프로젝트를 생성



# 실습준비

- 프로젝트 이름을 정하고 Foundation type으로 지정
- C로 할 수도 있지만 컴파일을 따로 해줘야 함

Choose options for your new project:

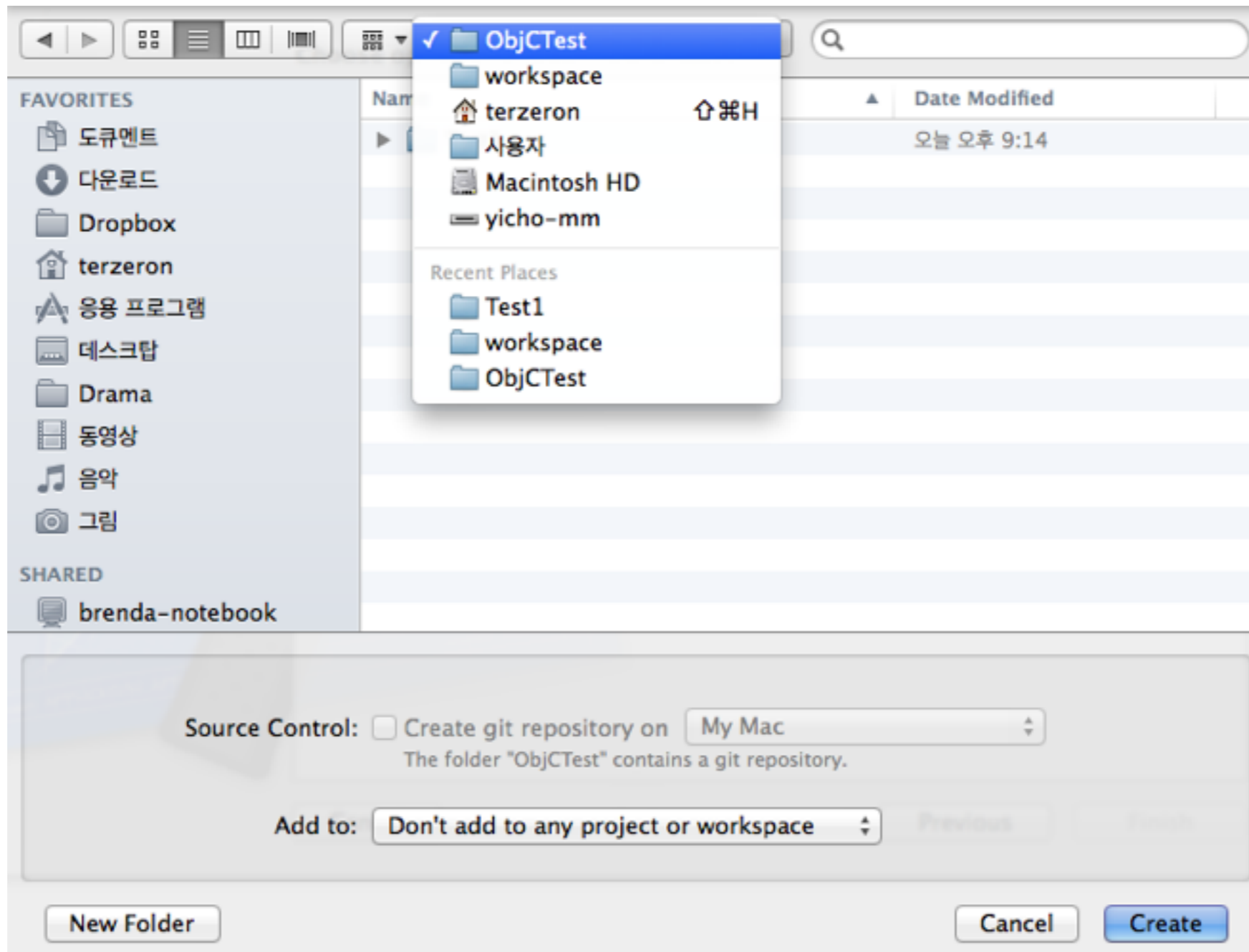


Product Name	Test3
Organization Name	terzeron.net
Company Identifier	net.terzeron
Bundle Identifier	net.terzeron.Test3
Type	Foundation

Cancel Previous Next

# 실습준비

- 프로젝트를 저장할 위치 지정



# 기초 문법

- Types

Java	ObjC
boolean (1bit)	BOOL (8bit)
byte	char
char	unichar
short	short
int	int
long	long long
float	float
double	double

# 기초문법

- Contants

Java	ObjC
true	YES
false	NO
null	nil (reference), NULL (pointer)

# 기초문법

- enum
  - enum {
    - Zero = 0,
    - One,
    - Two
  - } v;

# 기초문법

- typedef
  - typedef int PartCode;
  - PartCode aCode;
  
- typedef enum {
  - Strawberry,
  - Rhubarb,
  - Apricot
- } Flavor;

# 기초문법

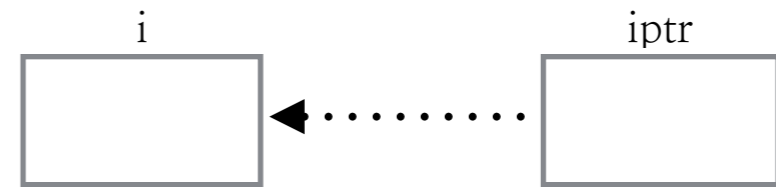
- pointer

- `int i = 1;`

- `int* iptr;`

- `iptr = &i;`

- `*iptr = 2;`



# 기초문법

- structure
  - typedef struct {
    - int key;
    - int count;
  - } KeyCount;
  - KeyCount counter;
  - counter.key = 1;
  - counter.count = 0;

# 기초문법

- array
  - `int numbers[10];`
  - `int j = number[3];`

# 기초문법

- function
  - NSDate\* earlierDate(NSDate\* a, NSDate\* b)
  - {
    - if (a == nil)
      - return b;
    - if (b == nil || [a compare:b] == NSOrderedAscending)
      - return a;
    - return b;
  - }

# 기초문법

- `static`
  - `int count = 1; // accessible from external modules`
  - `static int count = 1; // accessible only from local`
- `extern`
  - `extern int count;`
  - `extern void increment(int c);`

# 기초문법

- define
  - #define MAX\_COUNT 999
  - #ifdef DEVELOPER\_VERSION
    - NSAssert(poolSize < 256, @"pool overflow");
  - #else
    - if (poolSize >= 256)
      - return;
- #endif

# 기초문법

- include/import
  - include가 기계적인 insertion을 수행하는 반면,
  - import는 이미 insertion이 되어있는지 확인함
    - C header file에 관습적으로 넣던 다음과 같은 코드가 필요없음
      - #ifndef \_\_MY\_HEADER\_H\_\_
      - #define \_\_MY\_HEADER\_H\_\_
      - ...
      - #endif // \_\_MY\_HEADER\_H\_\_

# 제어문

- 예제

- int segment[100][100];
- int optimalLength = 200000;
- int lineLength;
- int bestFit = 0;
- for (int i=0; i<100; i++) {
  - lineLength = 0;
  - for (int j=0; j<100; j++) {
    - int s = segment[i][j];
    - if (s==0)
      - break;
    - lineLength += segment[i][j];

- if (lineLength > optimalLength)
  - goto nextLine;
- if (lineLength == optimalLength)
  - goto stop;
- if (lineLength > bestFit)
  - bestFit = lineLength;
- }
- nextLine:
  - ;
  - }
- stop:
  - return (lineLength);

# Message & Method

- Java
  - `object.method();`
  - `object.method(param);`
- Objective C
  - `[object method];`
  - `[object methodWithParam: param];`

# Message & Method

- Java

- `char c[] = new char[20];`
- `String s = "The time has come";`
- `s.getChars(3, 9, c, 1);`

- Objective C

- `unichar c[20];`
- `NSString* s = @"The time has come";`
- `[s getCharacters:&c[1] range:NSMakeRange(3, 6)];`

# Message & Method

- Multiline Message

- NSString\* new =

- [s stringByReplacingOccurrencesOfString:@"Walrus"

- withString:@"Carpenter"

- options:NSLiteralSearch

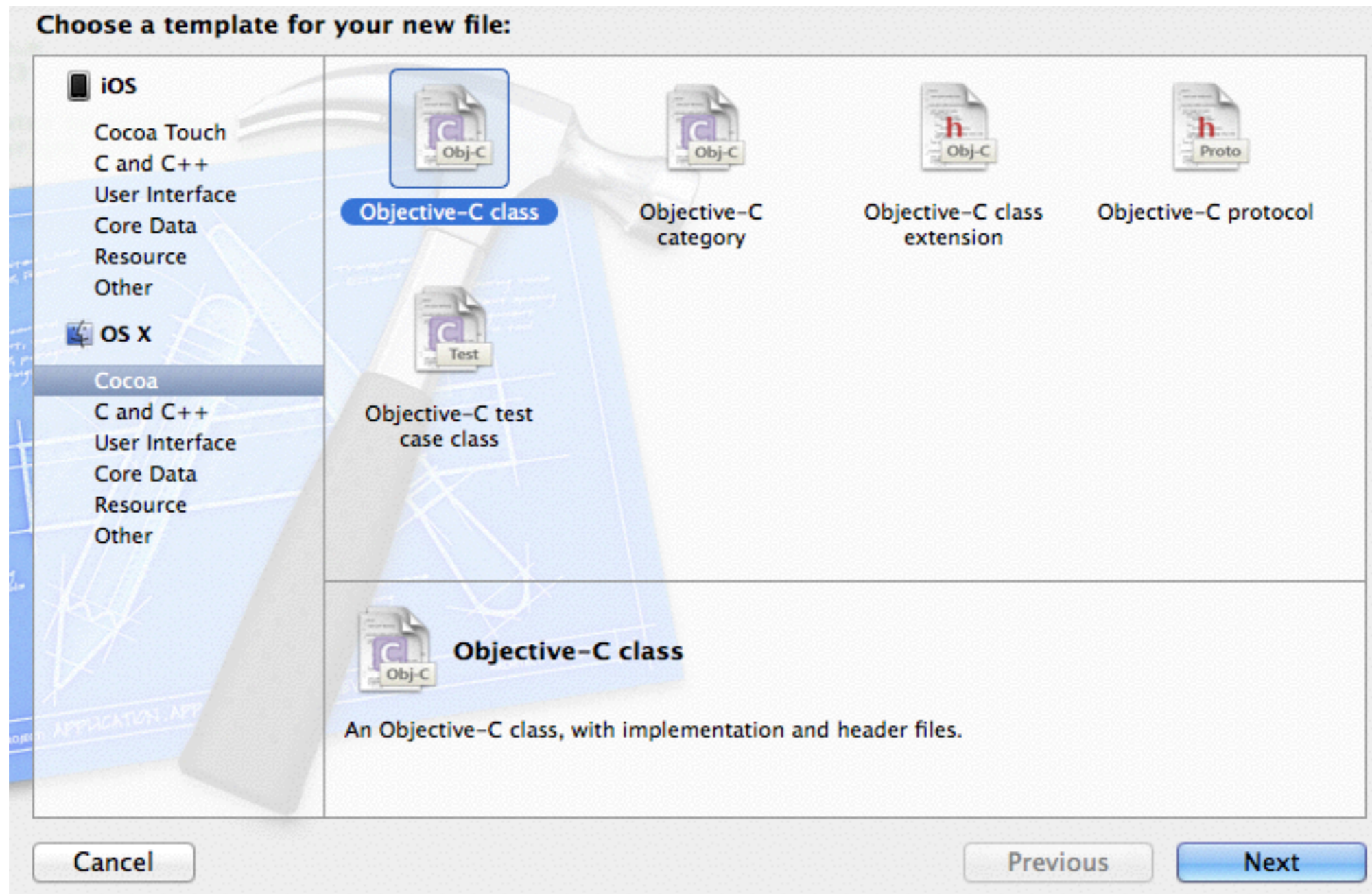
- range:NSMakeRange(20, 10)];

# Message & Method

- parameter & return type
  - – (id) objectForKey: (id) aKey
    - id i = [object objectForKey: 3];
  - – (NSMenuItem\*) itemWithTag: (NSInteger) aTag;
    - NSMenuItem\* mi = [object itemWithTag: 100];
  - –(void) addAttribute: (NSString\*) name value:(id) value range:(NSRange) aRange;
    - ex) [obj23 addAttribute: @"height" value: @"4" range: rangeInFeet];

# 실습준비

- File > New > File (⌘-N)



# Class

- interface

- @interface Person : NSObject {
  - NSString\* firstName;
  - NSString\* **secondName**;
  - BOOL adult;
- }
- @property (copy) NSString\* firstName;
- @property (copy) NSString\* **lastName**;
- @property (**getter=isAdult**) BOOL adult;
- @property (**readonly, nonatomic**) NSString\* **fullName**;
- @end

- @implementation Person
- @synthesize firstName;
- @synthesize lastName = secondName;
- @synthesize adult;
- - (NSString\*) fullName
- {
  - return ([NSString  
stringWithFormat:@"%@ %@",  
firstName, secondName]);
- }
- @end

# Class

- main.m
  - `Person* person = [[Person alloc] init];`
  - `person.firstName = @"James";`
  - `person.lastName = @"Dean";`
  - `person.adult = YES;`
  - `NSLog(@"%@ %d", person.fullName, person.isAdult);`

# Class

- property attribute

readonly /  
readwrite

copy / assign

copy the object value or the reference

retain

like assign, but retains the reference to the  
object in a managed memory

getter=name  
setter=name

nonatomic

non-synchronized method in Java  
manually written accessor method

# Protocol

- @class Thing;
- @protocol Living
  - – (float) age;
  - – (float) health;
  - – (NSDictionary\*) healthInfo;
- @end
- @protocol Communicating
  - @end
- @end
- @protocol Storing
  - – (NSDictionary\*) inventory;
  - – (BOOL) giveItem: (Thing\*) item to: (id<Storing>) recipient;
  - – (BOOL) acceptItem: (Thing\*) item from: (id<Storing>) recipient;
- @end
- @interface Character: Thing <Living, Communicating>
- @end
- @interface Earthlings: Character <Storing>
- @end

# Exception

- – (void) catcher
  - {
    - @try {
      - NSLog(@"%s trying", \_\_func\_\_);
      - [self thrower];
    - } @catch (SpecificException\* se) {
      - NSLog(@"caught SpecificException: %@", se);
    - } @catch (NSEException\* e) {
      - NSLog(@"caught NSEException: %@", e);
      - @throw e;
    - } @finally {
      - NSLog(@"%s finished", \_\_func\_\_);
    - }
  - }
- – (void) thrower
  - {
    - @throw
      - [NSEException exceptionWithName:@"MyException"
        - reason:@"doesn't work"
        - userInfo:nil];
  - }

# References

- <http://rypress.com/tutorials/objective-c/>
- <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>

Thank You

