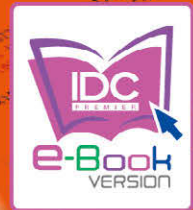




Basic Objective-C



- เรียนรู้การพัฒนาแอปฯ ด้วย Xcode 5
- อธิบายภาษา Objective-C อย่างละเอียด มีตัวอย่างประกอบทุกหัวข้อ
- รองรับการทำงานบน iOS 7
- เหมาะสำหรับนักเรียน นักศึกษา และผู้ที่สนใจพัฒนา iOS App ด้วยตัวเอง

Free : Source Code

<http://www.infopress.co.th/devbook/DevBookObjectiveC.zip>

■ คุุภะชัย ■ สบพานิษ ■ บรรณารักษ์ ■ กิตติพันธ์ ■ พลสวัสดิ์

special Thanks

Infopress Group ขอขอบคุณทุกการสนับสนุน
ที่มีเสมอมา จากประสบการณ์การถ่ายทอด
ความรู้ผ่านตัวหนังสือ เราจะมุ่งมั่นสร้างสรรค์งาน
ที่มีคุณภาพเพื่อคุณผู้อ่านตลอดไป ภาวนาให้ชื่อที่ว่า
" The Best of Knowledge Books "



Basic Objective-C

ผู้แต่ง	ศุภชัย สมพานิช	http://www.facebook.com/thaivb.net
บรรณาธิการ	กิตินันท์ พลสวัสดิ์	kitinan_p@idcpremier.com
ออกแบบปก	वलันต์ พึ่งพูลผล	
ออกแบบและจัดรูปเล่ม	วุฒิพันธ์ สมพระเมฆ, ลีรลักษณ์ วาระเลิศ	
พิสูจน์อักษร	สุนทรี บรรลือศักดิ์, มนฤดี ศรีอุทโยภาส	
eBook Production	สุริย์รัตน์ จิ๋ว, จิราภรณ์ โสภา	

X code เป็นเครื่องหมายการค้าของบริษัท Apple Inc. และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้เทคโนโลยีสารสนเทศทั้งในระดับพื้นฐานและระดับสูง เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน โดยเฉพาะงานที่เกี่ยวข้องกับสารสนเทศ ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ editor@infopress.co.th หรือโทรศัพท์หมายเลข 0-2962-1081 (อัตรา 10 คู่สาย) โทรสาร 0-2962-1084

ข้อมูลทางบรรณานุกรม



ศุภชัย สมพานิช
Basic Objective-C
นนทบุรี : ไอดีซี, 2556
280 หน้า

1. ออบเจกต์ทีฟ-ซี (ภาษาที่ใช้ในการเขียนโปรแกรม)
I ชื่อเรื่อง
006.76

ISBN 978-616-200-498-8

ตุลาคม 2556

สำหรับร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114
โทรสาร 0-2962-1084

สมาชิกสัมพันธ์

โทรศัพท์ 0-2962-1081-3 ต่อ 121
โทรสาร 0-2962-1084

ISBN

พิมพ์ครั้งที่ 1

จัดพิมพ์และจัดจำหน่ายโดย



บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 19 ห้อง 1901 อาคารจัสมินอินเตอร์เนชั่นแนลทาวเวอร์
ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120
โทรศัพท์ 0-2962-1081 (อัตรา 10 คู่สาย) โทรสาร 0-2962-1084

ราคา 265 บาท

บรรณาธิการ

ในปัจจุบันหากจะกล่าวถึงอุปกรณ์ทาง IT คงไม่มีใครไม่รู้จัก iPhone หรือ iPad ซึ่งถือได้ว่าเป็นอุปกรณ์ Device ที่ได้รับความนิยมสูงสุดในขณะนี้ และเมื่อพูดถึง iPhone หรือ iPad สิ่งตามมาก็คือ App ใช้งานต่างๆ ที่มีนับหมื่นนับแสน อีกทั้งมียอดดาวน์โหลดนับล้านโหลด จึงปฏิเสธไม่ได้เลยว่า App สำหรับใช้งานบน iPhone, iPad หรืออุปกรณ์ Device อื่นๆ บน iOS กำลังเป็นที่ต้องการของผู้ใช้งาน และนั่นก็คือ ทางเลือกที่น่าสนใจของนักพัฒนานั่นเอง

การพัฒนา Apps บน iOS จะใช้ภาษา Objective-C ในการพัฒนา โดยหนังสือเล่มนี้จะแนะนำให้ผู้อ่านเรียนรู้ตั้งแต่การติดตั้งเครื่องมือสำหรับพัฒนา หลักการพื้นฐานของภาษา Objective-C ตลอดไปจนถึงสามารถพัฒนา App ให้สามารถใช้งานได้จริง เหมาะสำหรับผู้เริ่มต้นที่ต้องการการเรียนรู้การพัฒนา App อย่างละเอียดและเป็นขั้นตอน โดยมีตัวอย่างประกอบทุกหัวข้อ

ทางสำนักพิมพ์หวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้เป็นพื้นฐานที่ดีในการใช้งานภาษา Objective-C เพื่อพัฒนา App บน iOS และทางเราพร้อมรับฟังข้อติชมทุกประการ หากหนังสือเล่มนี้มีข้อผิดพลาดประการใด ต้องขออภัยมา ณ ที่นี้ด้วย

กิตินันท์ พลสวัสดิ์
ตุลาคม 2556

คำนำ

หนังสือ “Basic Objective-C” เล่มนี้ เป็นการศึกษาการใช้งานภาษา Objective-C ซึ่งเป็นภาษาที่ใช้ในการพัฒนา App บน iOS ประกอบไปด้วยเนื้อหาตั้งแต่ระดับเบื้องต้น ไปจนกระทั่งสร้าง App อย่างง่ายบน iOS ได้จริง

ปัญหาที่สำคัญประการหนึ่งของภาษา Objective-C ก็คือ ไวยากรณ์ทางภาษาค่อนข้างซับซ้อนกว่าภาษาอื่นๆ เล็กน้อย ดังนั้น การนำเสนอเนื้อหาของภาษานี้จึงต้องมีความชัดเจนมากกว่าปกติ จึงจะทำให้เกิดความเข้าใจอย่างแท้จริง

รายละเอียดทางภาษาของ Objective-C มีมากมาย ผู้เขียนไม่สามารถนำเสนอได้ทั้งหมด จึงได้เลือกหัวข้อที่มีความจำเป็นและมีความสำคัญ เพื่อที่จะต่อยอดไปสู่การพัฒนา App บน iOS ในลำดับต่อไป

ในกรณีที่ผู้อ่านต้องการสอบถามปัญหา หรือติชมเนื้อหาในหนังสือ สามารถโพสต์ได้ที่แฟนเพจของผู้เขียนที่ <https://www.facebook.com/thaivb.net>

ศุภชัย สมพานิช
ตุลาคม 2556

Contents

บทที่ 1 เตรียมพร้อมก่อนเริ่มต้นเขียน Objective-C..... 1

เริ่มต้นอย่างไรกับการเขียน Objective-C.....	1
แนวทางการศึกษาของหนังสือเล่มนี้.....	2
การเตรียมสภาพแวดล้อมในการพัฒนา.....	2
การตรวจสอบเวอร์ชันของระบบปฏิบัติการ OS X.....	3
การสมัคร Apple ID.....	5
ขั้นตอนการดาวน์โหลด Xcode	6
ขั้นตอนการติดตั้ง Xcode	8

บทที่ 2 เริ่มต้นใช้งาน Xcode และสร้าง iPhone Apps 11

เริ่มต้นสร้าง App	11
ทดสอบรันโปรเจกต์.....	14
การเปิด/ปิดหน้าต่าง All Output.....	15
พื้นฐานการใช้ภาษา Objective-C ใน Xcode	16
เบื้องหลังการทำงานของ Xcode.....	17
การเปิดโปรเจกต์เดิม.....	18
การบันทึกโปรเจกต์.....	20
การปิดโปรเจกต์.....	21
วิธีการปิด Xcode	21
เริ่มเขียนโค้ดภาษา Objective-C	21
ระบบช่วยเหลือในการเขียนโค้ด.....	24
ทำความเข้าใจกับโครงสร้างของโปรเจกต์แบบ Command Line Tool	25

บทที่ 3 รู้จักไวยากรณ์และชนิดข้อมูลของภาษา Objective-C 27

ชนิดข้อมูลในโลกของ Objective-C	28
พื้นฐานการตั้งชื่อตัวแปร.....	28
ชนิดข้อมูล (Data Type).....	29
ทำความเข้าใจกับตัวอักษรแสดงค่า (format code หรือ format string).....	30
ขอบเขตของชนิดข้อมูลตัวเลข	32
การใช้งานค่าคงที่ (Constant)	32
การหาขนาดของชนิดข้อมูลด้วยฟังก์ชัน sizeof().....	33

การใช้งานข้อมูลชนิดตัวอักษร (char) และข้อความ (NSString).....	34
การใช้งานข้อมูลชนิดตัวเลข NSNumber	36
การใช้งานข้อมูลชนิด _Bool.....	38

บทที่ 4 การใช้งานตัวดำเนินการคำสั่งตรวจสอบและวนลูปในภาษา Objective-C..... 41

การตรวจสอบเงื่อนไขด้วยคำสั่ง if...else.....	41
การตรวจสอบเงื่อนไขด้วยคำสั่ง if...else if.....	43
ทำความรู้จักกับตัวดำเนินการด้านการเปรียบเทียบ.....	45
การใช้ตัวดำเนินการด้านการตรวจสอบเงื่อนไข.....	46
การใช้งานตัวดำเนินการด้านตรรกะ.....	47
การตรวจสอบด้วยเงื่อนไขแบบ and (เครื่องหมาย &&).....	48
การตรวจสอบด้วยเงื่อนไขแบบ or (เครื่องหมาย).....	49
การตรวจสอบผลตรงกันข้าม (เครื่องหมาย !).....	51
การตรวจสอบเงื่อนไขซ้อนเงื่อนไข.....	52
การตรวจสอบเงื่อนไขร่วมกับตัวดำเนินการด้านตรรกะ.....	54
การตรวจสอบหลายเงื่อนไขด้วยคำสั่ง switch...case.....	56
ตัวดำเนินการด้านคณิตศาสตร์.....	59
การวนลูป.....	60
การวนลูปด้วยคำสั่ง for.....	60
การวนลูปด้วยคำสั่ง while	63
การวนลูปแบบ do...while	66

บทที่ 5 การเขียนโปรแกรมเชิงวัตถุในโลกของ Objective-C (Object Oriented Programming) 69

การเขียนโปรแกรมเชิงวัตถุคืออะไร.....	69
OOP ในโลกของ Objective-C	70
การใช้ข้อมูลชนิดข้อความ NSString ในฐานะอบเจ็กต์	70
พื้นฐานการสร้างคลาส.....	71
การกำหนดให้ตัวแปรอบเจ็กต์ไม่ซ้ำกับเจ็กต์ด้วยคำสั่ง nil.....	78

บทที่ 6 ทำความรู้จักกับฟังก์ชัน (Function)..... 81

ฟังก์ชันใน Objective-C	81
การสร้างฟังก์ชันแบบไม่มีการส่งค่ากลับด้วยคำสั่ง void (เมธอดของอบเจ็กต์).....	82
การสร้างฟังก์ชันแบบมีค่าส่งกลับ	84
การสร้างฟังก์ชันแบบมีพารามิเตอร์	86

การสร้างฟังก์ชันที่มีพารามิเตอร์ตั้งแต่ 2 ตัวขึ้นไป	89
การสร้างฟังก์ชันโดยอาศัย Function Prototype	91
การสร้างเมธอดให้กับออบเจกต์	94
การสร้างเมธอดของออบเจกต์แบบหลายพารามิเตอร์	98

บทที่ 7 ทำงานกับข้อมูลชนิดข้อความ, วันที่ และเวลา..... 101

รูปแบบการสร้างตัวแปรชนิดข้อความ NSString	101
การแปลงข้อมูลชนิดข้อความให้เป็นข้อมูลชนิดตัวเลข	103
การค้นหาข้อความในข้อมูลชนิด NSString	104
การแทนที่ข้อความ	106
การเลือกข้อความตามจำนวนตัวอักษร	107
การแปลงข้อความภาษาอังกฤษเป็นตัวพิมพ์ใหญ่หรือพิมพ์เล็ก	108
การสร้างข้อความที่สามารถแก้ไขได้ด้วยคลาส NSMutableString	109
พื้นฐานการใช้งานข้อมูลชนิดวันที่และเวลา (NSDate)	112
การจัดรูปแบบวันที่ให้กับคลาส NSDate	113
การกำหนดวันที่และเวลา โดยอาศัยคลาส NSDateComponents	115
การแสดงวันที่ปัจจุบัน	117
การแสดงเวลาปัจจุบัน	118
การเพิ่ม/ลดจำนวนวัน, สัปดาห์, เดือน หรือปี	119
การคำนวณหาผลต่างระหว่างวัน โดยอาศัยชนิดข้อมูล NSDate	120
การแปลงชนิดข้อมูลข้อความให้เป็นชนิดข้อมูลวันที่และเวลา	122

บทที่ 8 ทำงานกับโครงสร้างข้อมูล..... 125

การใช้งานตัวแปรเป็นชุดแบบอาร์เรย์ด้วยคลาส NSArray	125
การค้นหาข้อมูลในตัวแปรอาร์เรย์ NSArray	127
การตรวจสอบผลการค้นหากรณีไม่พบข้อมูล	129
การเปรียบเทียบอาร์เรย์ NSArray	130
การแปลงอาร์เรย์ NSArray ไปเป็นข้อความ NSString	131
พื้นฐานการใช้งานตัวแปรอาร์เรย์แบบ NSMutableArray	132
การตรวจสอบจำนวนสมาชิกในตัวแปรอาร์เรย์แบบ NSMutableArray	134
การเพิ่มและถอดสมาชิกในตัวแปรอาร์เรย์แบบ NSMutableArray	135
การสลับตำแหน่งสมาชิกในอาร์เรย์ NSMutableArray	137
การเก็บข้อมูลแบบเป็นชุดด้วย NSEnumerator	138
การใช้งานตัวแปรอาร์เรย์ NSArray ในฐานะชุดข้อมูล Enumerator	140
การรวมอาร์เรย์ NSArray เข้าด้วยกัน	141
การสร้างอาร์เรย์ NSArray ใหม่จากอาร์เรย์ NSArray เดิม	143

การเก็บข้อมูลที่มีโครงสร้างแบบ Key และ Value ด้วยคลาส NSDictionary	144
การใช้งานโครงสร้างข้อมูล Key กับ Value แบบ NSMutableDictionary	146
การเก็บข้อมูลแบบ NSSet	149
ทำงานกับข้อมูลชนิด NSSet	150
การใช้งานโครงสร้างข้อมูลแบบ NSMutableSet	152

บทที่ 9 การเพิ่มเติมและแก้ไขคลาสด้วย Category..... 153

พื้นฐานการเพิ่มเติมความสามารถให้กับคลาสเดิมที่มีอยู่.....	153
การแก้ไขความสามารถของคลาสเดิม	159
การใช้งาน Category กับคลาสของ iOS.....	161

บทที่ 10 พื้นฐานการเขียน iPhone Apps 163

การสร้าง iPhone Apps แบบ Empty Application	163
การเพิ่มส่วนแสดงผลแบบ View เข้ามาในโปรเจกต์	166
การกำหนดส่วนแสดงผลเริ่มต้น	168
ทำความเข้าใจกับสภาพแวดล้อมของ Xcode แบบพัฒนา iPhone Apps.....	170
ทำความเข้าใจโครงสร้างโปรเจกต์สำหรับ iOS Application	171
พื้นฐานการออกแบบส่วนแสดงผลใน iPhone Apps	172
การทดสอบรัน iPhone Apps ด้วย iOS Simulator	173
การปิด iOS Simulator	174
แนวความคิดของการเขียนโปรแกรมแบบ MVC	175
การสร้าง iPhone Apps แบบรับการส่งข้อมูลจากผู้ใช้งาน.....	175
การกำหนดโครงสร้างในไฟล์ ViewController.h.....	178
การเขียนโค้ดกำหนดการทำงานในไฟล์ ViewController.m.....	180
การยกเลิกการเชื่อมต่อส่วนแสดงผล (*.xib) กับโค้ดโครงสร้างในไฟล์ (*.h).....	186
แนวทางการตั้งชื่อฟังก์ชันแบบ IBAction	186
พื้นฐานการรับและแสดงข้อมูล	191

บทที่ 11 การทำงานกับ Layout ในส่วนแสดงผล 199

ทำความเข้าใจกับโครงสร้างของส่วนแสดงผล	199
การเปลี่ยนสีพื้นหลังให้กับส่วนแสดงผลแบบ View	200
พื้นฐานการกำหนด Layout แบบแนวตั้งหรือแนวนอนในขณะออกแบบ	201
การควบคุม iOS Simulator แสดงผลแนวตั้งหรือแนวนอน	202
การเปลี่ยน Background ด้วยวิธีการเขียนโค้ด.....	203

บทที่ 12 พื้นฐานการใช้งานกลุ่มออบเจกต์ที่มีส่วนแสดงผล 207

การเพิ่มเติมความสามารถให้กับคลาสด้วยโปรโตคอล (Protocol).....	207
การซ่อนคีย์บอร์ดจำลองของออบเจกต์ Text Field.....	208
การตรวจสอบรูปแบบข้อความในออบเจกต์ Text Field.....	212
การกำหนดช่องใส่รหัส Password.....	212
การป้อนตัวเลขและสัญลักษณ์.....	213
การป้อนที่อยู่เว็บไซต์ (URL).....	214
การป้อนเฉพาะตัวเลข.....	214
การป้อนหมายเลขโทรศัพท์.....	215
การป้อนชื่อติดต่อ.....	216
การป้อน E-mail.....	216
การป้อนตัวเลขจำนวนจริง.....	217
การป้อนข้อความแบบ Twitter.....	217
การตรวจสอบปุ่ม Button ที่ถูกคลิกผ่านทางชนิดข้อมูล id.....	218
การแสดงรูปภาพด้วยคลาส UIImageView	223
การกำหนดรูปภาพในขณะออกแบบ.....	223
การกำหนดรูปภาพโดยการเขียนโค้ด.....	225
การสร้างแถบเลื่อนด้วยคลาส UISlider	230
การสร้างสวิตช์เปิด/ปิดด้วยออบเจกต์ Switch	236
การสร้างแถบปุ่มกดด้วยคลาส UISegmentedControl	240
การแสดงผลการประมวลผลด้วย Activity Indicator View.....	246
การแสดงความคืบหน้าการทำงานด้วยออบเจกต์ Progress View	251
การสร้างตัวเลือกวันที่และเวลาด้วยออบเจกต์ Date Picker	255
การแสดงรายการด้วยออบเจกต์ Picker View.....	260



objective-c

เตรียมพร้อมก่อนเริ่มต้นเขียน Objective-C

Chapter

1

หนังสือ Basic Objective-C เล่มนี้ เป็นการศึกษาการใช้งานภาษา Objective-C และพัฒนา iPhone Apps อย่างง่าย เหมาะสำหรับผู้อ่านที่ต้องการศึกษาไวยากรณ์ของภาษา Objective-C ก่อนที่จะนำไปใช้ในการพัฒนา iPhone Apps ในลำดับต่อไป

ผู้อ่านหลายท่านอาจจะมีคำถามว่า ภาษา Objective-C ใช้พัฒนาแอปพลิเคชันสำหรับ iPhone ได้อย่างเดียวหรือไม่ สามารถนำไปพัฒนาแอปพลิเคชันสำหรับ iPad ได้หรือไม่ คำตอบคือ ภาษา Objective-C สามารถพัฒนาแอปพลิเคชันที่ทำงานได้ทั้งบน iPhone และ iPad หรือจะกล่าวอีกอย่างก็คือ ภาษา Objective-C สามารถใช้พัฒนาแอปพลิเคชันสำหรับทำงานบน iOS ได้ทุก iDevice นั่นเอง

แต่สำหรับหนังสือเล่มนี้ขอยกตัวอย่างการใช้ภาษา Objective-C ในการพัฒนาแอปพลิเคชันบน iPhone เพื่อเป็นแนวทางในการเรียนรู้เท่านั้น

จากประสบการณ์ของผู้เขียน ปัญหาของผู้เริ่มต้นที่ผู้เขียนให้ความสำคัญก็คือ “เริ่มต้นอย่างไรดี” และ “เริ่มต้นอย่างถูกต้อง” เป็นสิ่งที่ผู้เขียนต้องการนำเสนอให้ผู้อ่านเห็นว่า ถ้าผู้อ่านเริ่มต้นอย่างถูกต้อง มีวิธีการที่ดีในการพัฒนา iPhone Apps จะไม่ใช่เรื่องยากอีกต่อไป

คำถามแรกที่หลายคนอยากทราบมากที่สุดก็คือ เราจะเริ่มต้นเขียน iPhone Apps อย่างไรดี ถือเป็นคำถามที่ตอบยากมากที่สุดเช่นกัน เพราะผู้อ่านแต่ละคนมีที่มาของปัญหาแตกต่างกัน แต่มีคำถามเดียวกัน ผู้เขียนขอยกตัวอย่างจุดเริ่มต้นของผู้เขียนมาเป็นกรณีศึกษาว่า ผู้เขียนเริ่มต้นกับ iPhone Apps อย่างไร

ผู้เขียนแยกที่มาของปัญหาของผู้อ่านแต่ละคนออกเป็น 2 กลุ่มใหญ่คือ

1. ผู้อ่านที่เคยศึกษาภาษาอื่นๆ มาบ้างในขั้นต้นในขณะที่เรียน หรือเกิดจากการศึกษาด้วยตัวเอง เขียนโค้ดได้บ้างไม่ได้บ้าง เช่น ภาษา C, C++, Java, PHP, VB, C#, ASP.NET เป็นต้น ศึกษาไม่นานน้อยแตกต่างกันไปตามแต่ละบุคคล แล้วต้องการมาเขียน iPhone Apps
2. ผู้อ่านที่เป็นโปรแกรมเมอร์ที่มาจากสายอื่นๆ แล้วต้องการมาเขียน iPhone Apps

สำหรับตัวผู้เขียนเองถือว่าเป็นกลุ่มที่ 2 กล่าวคือ เป็นโปรแกรมเมอร์สาย .NET แล้วต้องการมาเขียน iPhone Apps ซึ่งวงจรของการเรียนรู้โดยทั่วไปของผู้เขียนก็คือ เริ่มต้นหัดเขียนโปรแกรมง่ายๆ ก่อน จากนั้นศึกษาเพิ่มเติมเพื่อให้รู้มากยิ่งขึ้นว่า ถ้าเราต้องการทำโปรแกรมที่ซับซ้อนมากขึ้นจะมีวิธีการอย่างไร

แต่สำหรับการพัฒนา iPhone Apps แล้ว มีข้อแตกต่างตรงกันข้ามอย่างสิ้นเชิงกล่าวคือ ไม่ว่าเราจะเขียนภาษา Objective-C เป็นหรือไม่เป็นก็ตาม โครงสร้างของ iPhone Apps ถูกกำหนดให้เป็นมาตรฐานตั้งแต่เริ่มต้นเลย ถึงแม้ว่าเราจะเขียน App ง่ายๆ ก็ตาม

หมายความว่า ณ จุดเริ่มต้นของการเขียน iPhone Apps มีทั้งการศึกษารายละเอียดของภาษา Objective-C, การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) และการแบ่งหน้าที่การทำงานแบบ Model-View-Controller (MVC) ในเวลาเดียวกัน

เห็นได้ว่าล้าหลังแค่การศึกษารายละเอียดของภาษา Objective-C ก็ยากลำบากแล้ว ยังมีโครงสร้างที่เป็นมาตรฐานเข้ามาบังคับอีกชั้นหนึ่ง จึงทำให้การเริ่มต้นเขียน iPhone Apps ค่อนข้างลำบากกว่าภาษาอื่นๆ อย่างแน่นอน

สิ่งที่กล่าวมาในข้างต้นคือ โจทย์ของหนังสือเล่มนี้ที่ผู้เขียนต้องตอบผู้อ่านให้ได้ ต้องแก้ไขปัญหาคือที่ผู้เขียนเคยประสบมาให้ได้ ผู้เขียนอยากให้ผู้อ่านเริ่มต้นเขียน iPhone App แบบยั่งยืน ไม่ฉาบฉวย เรียนรู้และเข้าใจได้อย่างแท้จริง

แนวทางการนำเสนอเนื้อหาของหนังสือเล่มนี้

แนวทางการนำเสนอเนื้อหาของหนังสือเล่มนี้ เหมาะสำหรับผู้อ่านมือใหม่ที่ต้องการเริ่มต้นศึกษาโครงสร้างไวยากรณ์ของภาษา Objective-C และต้องการพัฒนา iPhone Apps ในขั้นต้นประกอบด้วย 2 โครงสร้างหลักคือ

- ♦ ศึกษาไวยากรณ์ของภาษา Objective-C อย่างละเอียด อันเนื่องมาจากไวยากรณ์ของภาษา Objective-C ค่อนข้างซับซ้อนพอสมควร เป็นปัญหาสำหรับผู้อ่านที่เป็นมือใหม่ในโลกของการเขียนโปรแกรม
- ♦ การสร้าง iPhone Apps ในขั้นต้น โดยการใช้งานและหน้าที่ของกลุ่มออบเจกต์ที่มีส่วนแสดงผล เพื่อสร้างส่วนแสดงผลอย่างง่าย

โดยโค้ดตัวอย่างที่ลงในหนังสือเล่มนี้ทุกตัวอย่าง จะมีการบอกว่าโค้ดดังกล่าวเป็นของไฟล์ที่ชื่อว่าอะไร เพื่อให้ง่ายต่อการศึกษาโค้ดที่ผู้เขียนนำเสนอในแต่ละหัวข้อ

การเตรียมสภาพแวดล้อมในการพัฒนา

ในการเขียนโปรแกรมภาษา Objective-C เราจะใช้เครื่องมือที่เรียกว่า Xcode ที่ติดตั้งบนระบบปฏิบัติการ OS X ซึ่งเป็นระบบปฏิบัติการที่ทำงานอยู่บนเครื่อง Mac และปัจจุบัน iOS ถูกพัฒนามาจนถึงเวอร์ชัน iOS 7 เพื่อให้การพัฒนา iPhone Apps รองรับ iOS 7 อย่างสมบูรณ์แบบ ให้ผู้อ่านอัปเดต OS X เป็นเวอร์ชัน 10.8.x ขึ้นไปเพื่อให้สามารถรองรับ Xcode 5 ได้

NOTE



ในกรณีที่ผู้อ่านมีแต่เครื่อง PC (ติดตั้งระบบปฏิบัติการ Windows) ผู้อ่านสามารถใช้งาน Xcode ร่วมกับระบบปฏิบัติการ OS X บน PC ได้เช่นกัน ซึ่งผู้อ่านสามารถค้นหาวีธีการได้ใน Google

ในที่นี้สามารถสรุปขั้นตอนการเตรียมสภาพแวดล้อมในการเขียนโปรแกรมภาษา Objective-C ได้ดังนี้

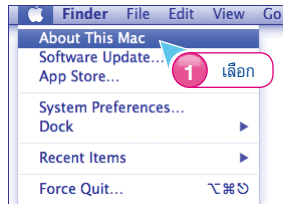
1. ตรวจสอบเวอร์ชันของระบบปฏิบัติการ OS X
2. สมัคร Apple ID
3. ดาวน์โหลด Xcode
4. ติดตั้ง Xcode

การตรวจสอบเวอร์ชันของระบบปฏิบัติการ OS X

1. ตรวจสอบระบบปฏิบัติการ OS X ว่าเป็นเวอร์ชันล่าสุดหรือไม่ สามารถทำได้โดยการคลิกปุ่ม  > About This Mac ดังรูปที่ 1-1

รูปที่ 1-1

แสดงเวอร์ชัน
ปัจจุบันของ
OS X

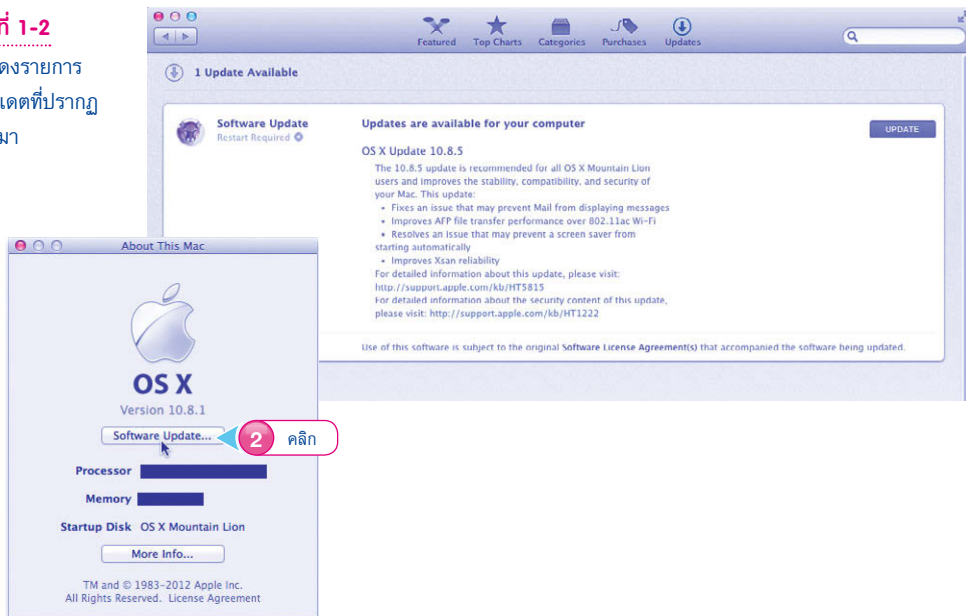


จากรูปที่ 1-1 พบว่าระบบปฏิบัติการ OS X ปัจจุบันของเครื่องที่ทดสอบเป็น OS X เวอร์ชัน 10.8.1 ไม่ใช่เวอร์ชันใหม่ตามที่ผู้เขียนต้องการ เนื่องจากเนื้อหาที่นำเสนอในหนังสือเล่มนี้ใช้ OS X เวอร์ชัน 10.8.5 ขึ้นไปเพื่อใช้ทำงานร่วมกับ Xcode 5 ขึ้นไป

2. คลิกปุ่ม **Software Update...** เพื่อตรวจสอบว่ามีรายการ Update หรือไม่ ดังรูปที่ 1-2

รูปที่ 1-2

แสดงรายการ
อัปเดตที่ปรากฏ
ขึ้นมา



3. หลังจากดาวน์โหลดเสร็จแล้ว ก็จะเข้าสู่การอัปเดตระบบปฏิบัติการ OS X ให้เป็นเวอร์ชันใหม่โดยอัตโนมัติ ดังรูปที่ 1-3

รูปที่ 1-3

แสดงการ
อัปเดต OS X
เป็นเวอร์ชัน
10.8.5



ณ จุดนี้ OS X ของเราพร้อมพัฒนา iPhone Apps สำหรับ iOS 7 แล้ว

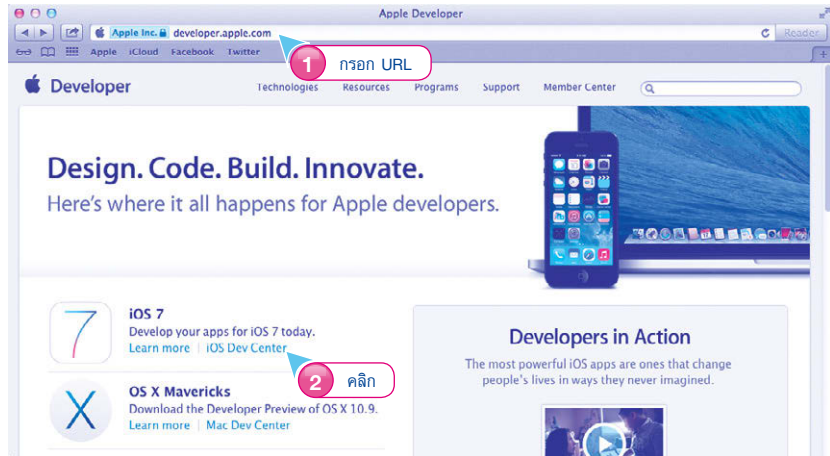
การสมัคร Apple ID

การดาวน์โหลด Xcode จากเว็บไซต์ของ Apple ผู้อ่านต้องมี Apple ID ก่อน ผู้อ่านสามารถสมัครได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆ ทั้งสิ้น จะมีขั้นตอนดังนี้

1. ไปที่เว็บไซต์ของ Apple ที่ <https://developer.apple.com/>
2. คลิกที่ iOS Dev Center ดังรูปที่ 1-4

รูปที่ 1-4

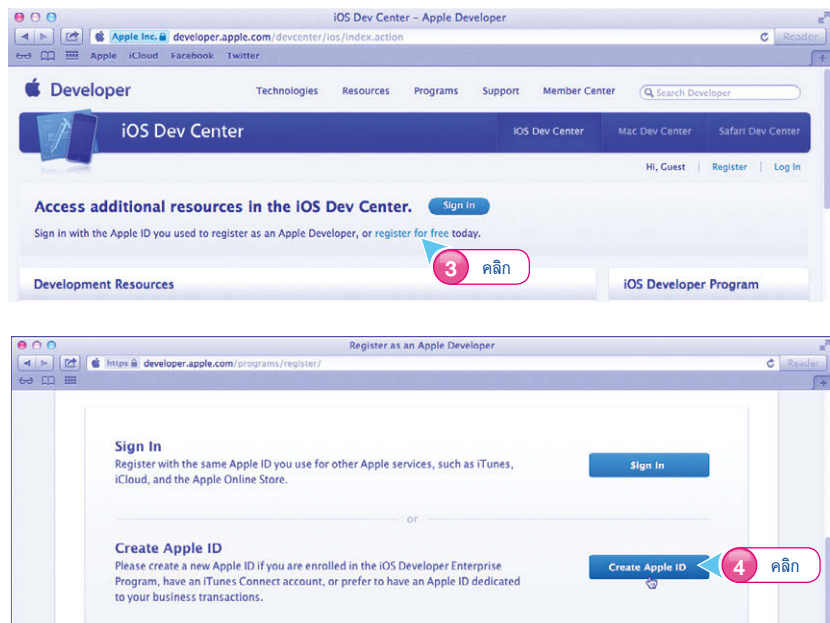
แสดงเว็บไซต์
ของ Apple



3. คลิกที่ register for free
4. คลิกปุ่ม **Create Apple ID** เพื่อเริ่มเข้าสู่ขั้นตอนการลงทะเบียน

รูปที่ 1-5

แสดงการลง
ทะเบียนเพื่อ
ขอรับ Apple ID



จากรูปที่ 1-5 ในระหว่างการลงทะเบียน ขอให้ผู้อ่านป้อนข้อมูลจริงตามที่ปรากฏขึ้นมาเพื่อใช้ในกรณีขาย iPhone Apps ในโอกาสต่อไป

ขั้นตอนสุดท้ายของการลงทะเบียน ผู้อ่านต้องยืนยันตัวเองด้วย E-mail ของผู้อ่าน หากไม่มีข้อผิดพลาดใดๆ ในขั้นตอนการลงทะเบียน ก็จะได้พบกับหน้าจอดังรูปที่ 1-6

รูปที่ 1-6

กรณีลงทะเบียน
เสร็จสมบูรณ์



จากรูปที่ 1-6 ณ จุดนี้ผู้อ่านจะมี Apple ID พร้อมใช้งานแล้ว

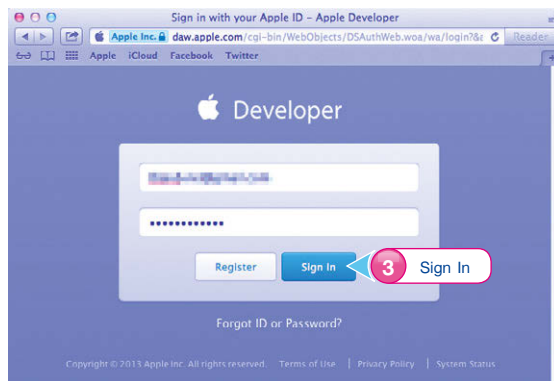
ขั้นตอนการดาวน์โหลด Xcode

หลังจากที่คุณได้ Apple ID มาแล้ว คุณสามารถดาวน์โหลด Xcode มาใช้งานได้ฟรี มีขั้นตอนดังนี้

1. กรอก URL ดังนี้ <https://www.developer.apple.com/xcode/index.php>
2. ให้คลิกที่ Member Center เพื่อ Sign In เข้าสู่ระบบด้วย Apple ID
3. Sign In เข้าสู่ระบบด้วย Apple ID ดังรูปที่ 1-7

รูปที่ 1-7

แสดงการ
Sign In ด้วย
Apple ID
เข้าสู่ระบบ



4. ให้ผู้อ่านคลิกที่หัวข้อ iOS

รูปที่ 1-8

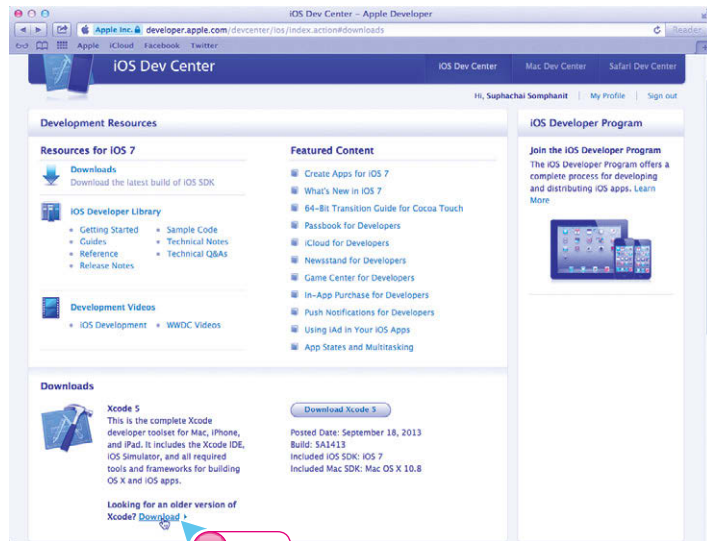
แสดงส่วน
ของ Developer
สำหรับสมาชิก



5. ให้ผู้อ่านคลิกที่ Download ก็จะเข้าสู่หน้าจอตาวนโหลด Xcode 5 สำหรับพัฒนา App บน iOS 7 ดังรูปที่ 1-9

รูปที่ 1-9

แสดงลิงค์
ดาวน์โหลด
Xcode 5
สำหรับ iOS 7



จากรูปที่ 1-9 ขอบเขตเนื้อหาของหนังสือเล่มนี้คือ การใช้งาน Xcode เวอร์ชัน 5 ขึ้นไป หลังจากที่ผู้อ่านดาวน์โหลด Xcode เสร็จเรียบร้อยแล้ว โดยปกติจะถูกเก็บอยู่ในโฟลเดอร์ Downloads ของ OS X

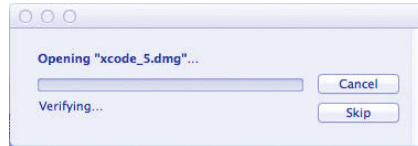
ขั้นตอนการติดตั้ง Xcode

ขั้นตอนการติดตั้ง Xcode มีดังนี้

1. ดับเบิลคลิกไฟล์ xcodexxxxxx.dmg เพื่อเริ่มต้นติดตั้ง ดังรูปที่ 1-10

รูปที่ 1-10

แสดงการ
ตรวจสอบ
ความสมบูรณ์
ของไฟล์ติดตั้ง
Xcode



2. ลาก Xcode เข้าสู่โฟลเดอร์ Applications เพื่อติดตั้ง Xcode เข้าสู่ระบบปฏิบัติการ OS X

รูปที่ 1-11

ระหว่างการ
ติดตั้ง Xcode



ในกรณีที่ OS X ของผู้อ่านมีการติดตั้ง Xcode เวอร์ชันเดิมอยู่ ผู้อ่านสามารถคลิกปุ่ม **Replace** เพื่อติดตั้ง Xcode เวอร์ชันใหม่ทับ Xcode เวอร์ชันเดิมได้เลย

3. หากการติดตั้งไม่มีข้อผิดพลาดใดๆ เกิดขึ้น ก็จะมีไอคอน Xcode ปรากฏใน Dock bar ด้านล่าง ให้ผู้อ่านลองคลิกเพื่อทดสอบเปิด Xcode ขึ้นมา ณ จุดนี้ Xcode ของผู้อ่านจะพร้อมใช้งานแล้ว ดังรูปที่ 1-12



รูปที่ 1-12

แสดงหน้าจอ

แรกของ Xcode



สรุปท้ายบท

เนื้อหาที่นำเสนอในบทนี้เป็นเพียงการเตรียมตัวสภาพแวดล้อมของ OS X ของเราให้พร้อมใช้งาน Xcode เพื่อพัฒนา iPhone Apps ในลำดับต่อไปเท่านั้น



objective-c

เริ่มต้นใช้งาน Xcode และสร้าง iPhone Apps

Chapter

2

เนื้อหาที่น่าสนใจในบทนี้ ถูกจัดอยู่ในระดับพื้นฐานสำหรับผู้เริ่มใหม่ควรทราบในขั้นต้น ผู้เขียนขอให้ผู้อ่านทำความเข้าใจกับโครงสร้างของโปรเจกต์ที่เกิดจาก Xcode ว่ามีส่วนประกอบใดบ้างที่เกี่ยวข้องกับการทำงานของเรา

เริ่มต้นสร้าง App

ในหัวข้อนี้ผู้อ่านจะได้เริ่มทดสอบว่า Xcode ของเราพร้อมใช้งานแล้วหรือไม่ โดยมีขั้นตอนดังนี้

ตัวอย่างที่ 2-1 เริ่มต้นใช้งาน Xcode

1. ให้ดับเบิลคลิกไอคอน Xcode เพื่อเริ่มต้นเขียน App ดังรูปที่ 2-1

รูปที่ 2-1

แสดงไอคอน
ของ Xcode



2. คลิกที่ Create a new Xcode project เพื่อเริ่มต้นสร้างโปรเจกต์ว่างขึ้นมาใหม่ ดังรูปที่ 2-2

รูปที่ 2-2

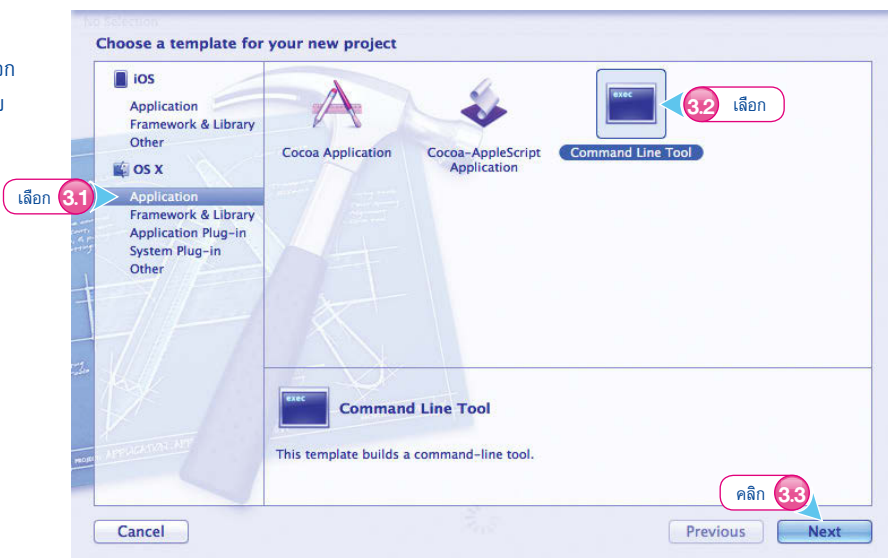
แสดงการ
เลือกรายการ
Create a new
Xcode project



3. สำหรับโปรเจกต์แรกของเราไม่ใช่ของ iPhone Apps แต่เป็นของ OS X ผู้เขียนต้องการทดสอบ Xcode ของผู้أنว่าพร้อมใช้งานแล้วหรือไม่เท่านั้น โดยการเลือกหัวข้อ Application ภายใต้ Template แบบ OS X และเลือกประเภทโปรเจกต์แบบ Command Line Tool (ป้อนคำสั่งแบบข้อความ)

รูปที่ 2-3

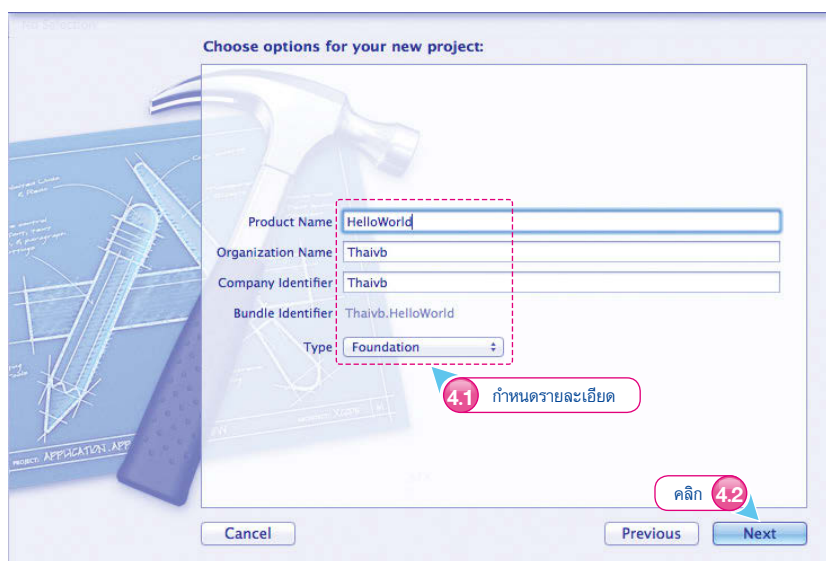
แสดงการเลือก
โปรเจกต์แบบ
Command
Line Tool



4. กำหนดรายละเอียดขั้นต้นของโปรเจกต์ปัจจุบัน โดยกำหนดดังนี้
- ◆ **ช่อง Product Name** หมายถึง ชื่อ App ของเรา ในกรณีนี้ตั้งชื่อว่า HelloWorld
 - ◆ **ช่อง Organization Name และช่อง Company Identifier** หมายถึง ชื่อองค์กรและบริษัท
 - ◆ **ช่อง Type** หมายถึง ประเภทของโปรเจกต์ ในกรณีนี้เลือกรายการ Foundation

รูปที่ 2-4

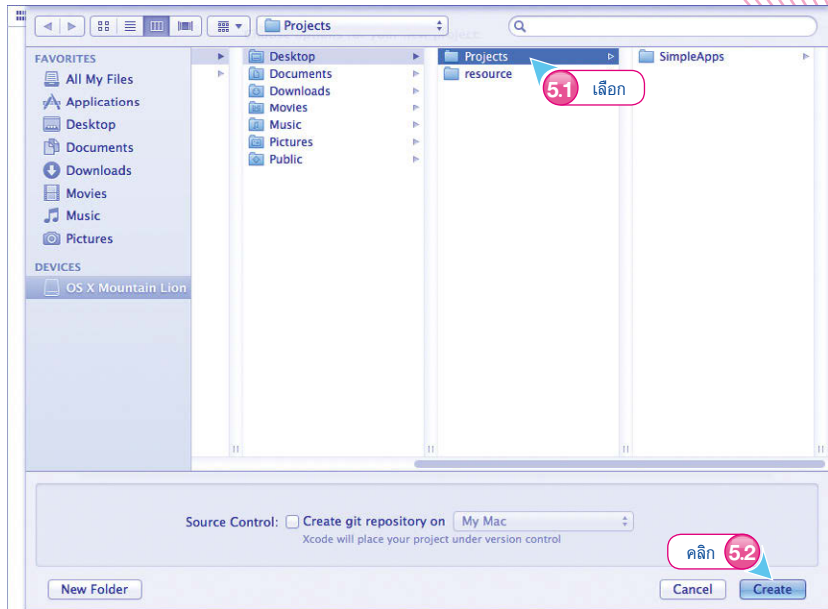
แสดงการ
กำหนด
รายละเอียด
ขั้นต้นของ
โปรเจกต์ปัจจุบัน



- เลือกโฟลเดอร์ที่ใช้เก็บโปรเจกต์ปัจจุบัน ในกรณีนี้คือ โฟลเดอร์ Projects หรือโฟลเดอร์ที่ผู้อ่านต้องการจัดเก็บโปรเจกต์ก็ได้ โดยคลิกปุ่ม **Create** เพื่อเริ่มต้นสร้างโปรเจกต์ต่างๆ ขึ้นมา ดังรูปที่ 2-5

รูปที่ 2-5

แสดงการ
สร้างโปรเจกต์
ต่างๆ ขึ้นมา
ในโฟลเดอร์
Projects



NOTE

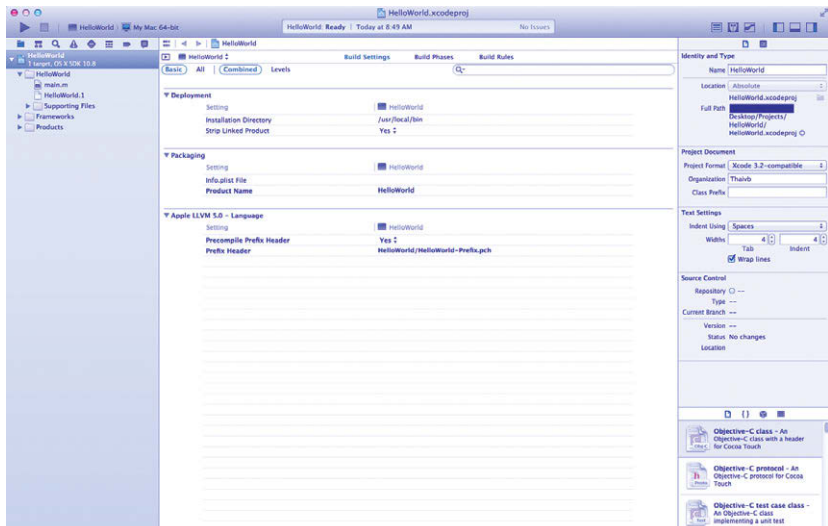


ตัวอย่างโปรเจกต์ต่างๆ ที่นำเสนอในหนังสือเล่มนี้ ผู้เขียนจัดเก็บไว้ในโฟลเดอร์ที่ชื่อว่า Projects บน Desktop ของ OS X สำหรับผู้อ่านก็แล้วแต่ว่าต้องการจัดเก็บไว้ที่ใด

- ที่หน้าต่างด้านซ้ายแสดงโครงสร้างของโปรเจกต์ปัจจุบัน ซึ่งเป็นโปรเจกต์แบบ Command Line Tool

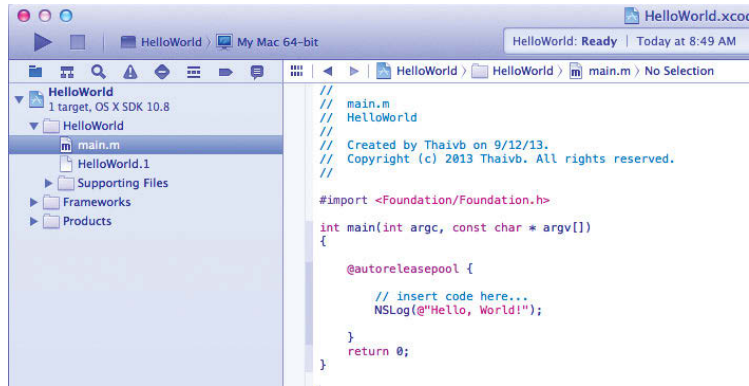
รูปที่ 2-6

แสดงสภาพ
แวดล้อมของ
Xcode แบบ
Command
Line Tool



รูปที่ 2-7

แสดงโครงสร้าง
ของโปรเจกต์
ที่ชื่อว่า
HelloWorld



จากรูปที่ 2-7 ไฟล์ main.m คือ ไฟล์แรกที่ถูกสั่งให้ทำงานในโปรเจกต์แบบ Command Line Tool ให้ผู้อ่านคลิกที่ไฟล์ main.m เพื่อแสดงรายละเอียดโค้ด Objective-C ของไฟล์ main.m ดังได้ต่อไปนี้

โค้ดภาษา Objective-C ที่ 2-1 App แรกของเรา (main.m)

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[])
{
    @autoreleasepool {
        // insert code here...
        NSLog(@"Hello, World!");
    }
    return 0;
}
```

จากโค้ด Objective-C ข้างต้นเกิดจาก Xcode เขียนคำสั่งให้ผู้อ่านโดยอัตโนมัติ ซึ่งก็คือ ฟังก์ชัน NSLog() ที่ทำหน้าที่สั่งให้พิมพ์ข้อความ “Hello, World !” ในช่อง All Output

ทดสอบรันโปรเจกต์

การทดสอบรันโปรเจกต์ ให้ผู้อ่านคลิกปุ่ม ▶ ที่อยู่มุมซ้ายบน ดังรูปที่ 2-8

รูปที่ 2-8

แสดงปุ่ม
สำหรับรัน
โปรเจกต์

