

Основы языка Swift

Урок 1

Введение

Знакомство с основами языка

Регламент

- 8 уроков по 2 часа.
- Домашние задания.
- Видеозапись будет.
- Задавайте вопросы!



Что будем изучать на курсе?

- Xcode.
- Язык Swift.
- Переменные. Коллекции. Управляющие конструкции.
- ООП.



Почему Swift?



Почему Swift?

1. Один из двух доступных языков для приложений Apple.
2. Современный, простой, удобный, мощный.
3. Почти все новые проекты пишутся на Swift.



План урока

- Xcode.
- Переменные.
- Коллекции.
- Дебаггер.



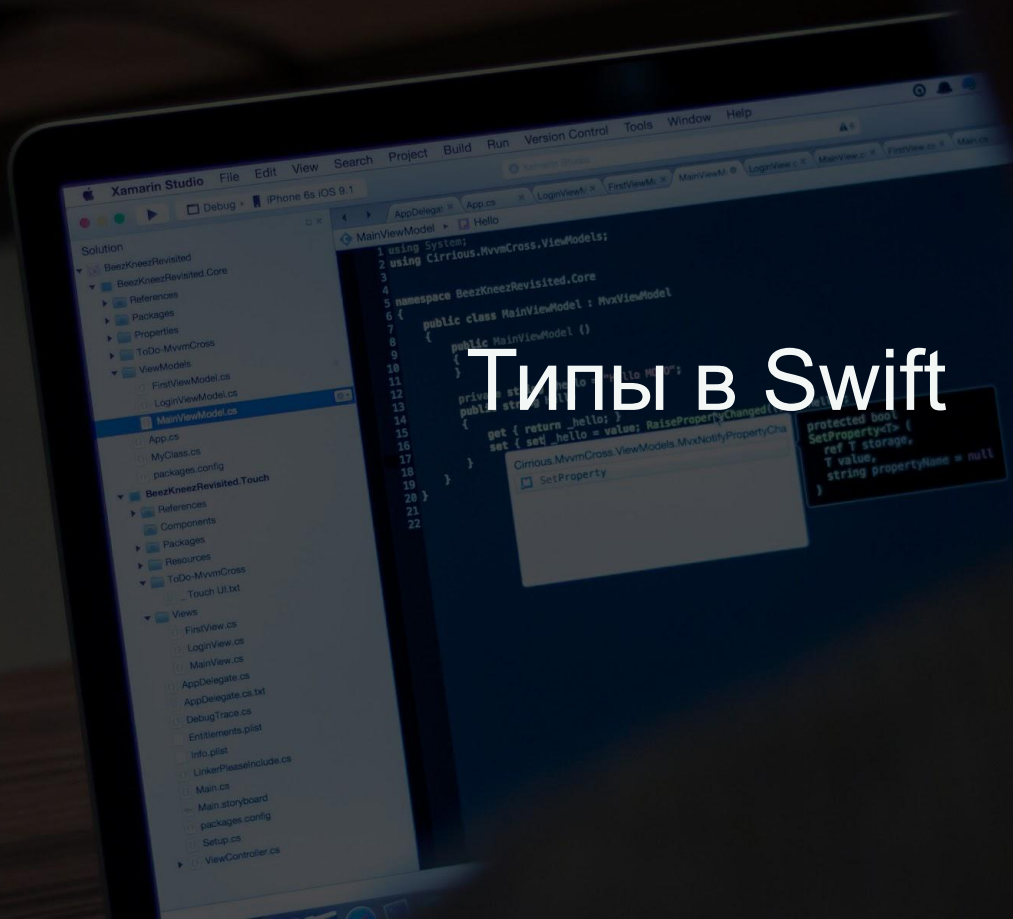
Переменные и константы



1. Объявление констант – `let`.
2. Объявление переменных – `var`.
3. Всегда объявляем через `let`, меняем на `var` только при реальной необходимости.
4. Константа `let` в Swift – фактически константный указатель на константные данные. Это касается как примитивных типов, так и объектных, в частности, коллекций (`String`, `Array`).



Типы в Swift



Тип данных	Описание	Диапазон
Int	Целочисленный тип	Диапазон соответствует разрядности ОС-Int32 или Int64
UInt	Целочисленный тип, только с положительными значениями	Диапазон соответствует разрядности ОС-UInt32 или UInt64
Double	64-битное число с плавающей точкой	15 десятичных цифр



Тип данных	Описание	Диапазон
Float	32-битное число с плавающей точкой	6 десятичных цифр
Bool	Логический тип	Может принимать значение true и false
Character	Символьный тип	Один символ
String	Строка	Любые символы



Преобразование типов

- Предельно строгая типизация, отсутствие скрытых конверсий.
- Язык может угадывать тип (Type Inferring) при присваивании значения, но это далеко не всегда полезно.
- Любое несовпадение типов приведет к ошибке времени компиляции.



Преобразование типов

- Рекомендуется явно указывать тип переменной или константы, если тип не получается однозначно определить, основываясь на контексте.
- Конвертация значения и типа переменной: `let a: Double = Double(b)`
- Конвертация только типа переменной: `let a: Double = b as? Double`



Опциональные типы

- Существенно повышают безопасность вашего кода.
- Значение с обычным типом никогда не содержит `nil`, а значение с опциональным типом – может содержать.

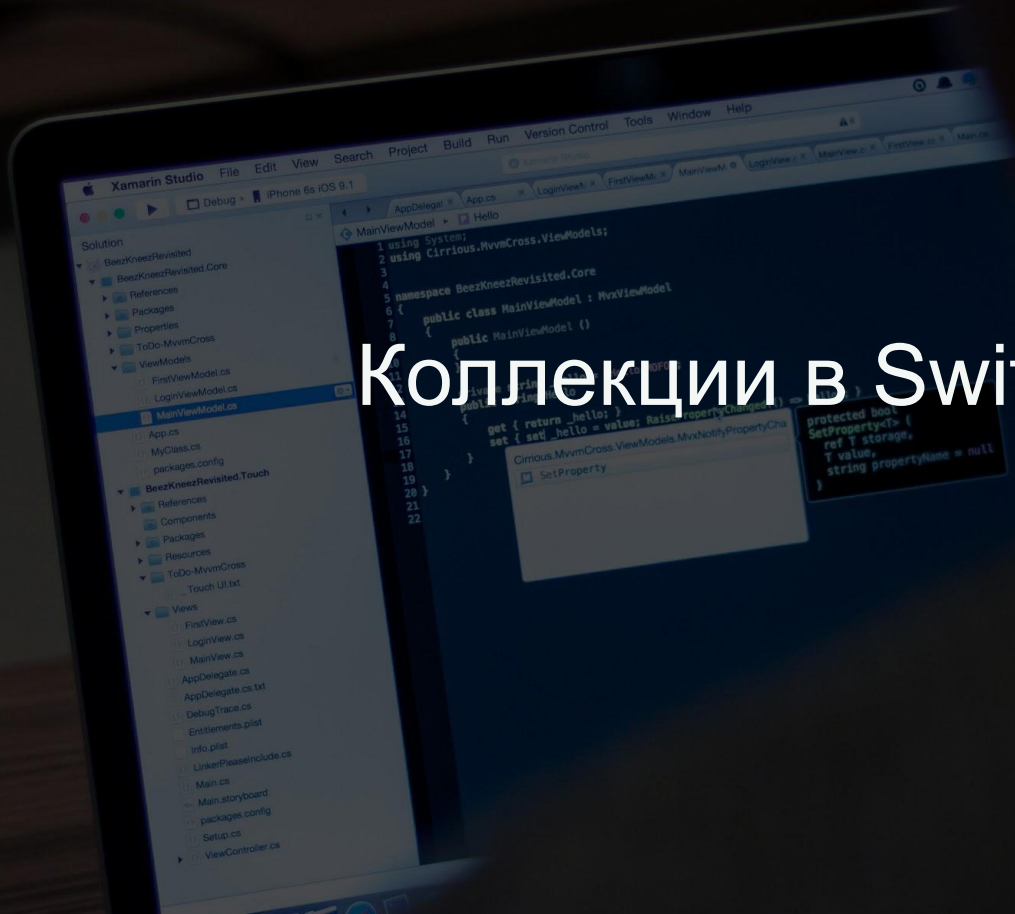


Опциональные типы

- Для работы с опциональным значением нужно применить оператор принудительного извлечения.
- Опциональным может быть любой Swift-тип, включая все ваши собственные типы без исключения.
- Знак опционала – ?, знак принудительного извлечения – !
- Swift предоставляет Optional Binding для удобства работы.



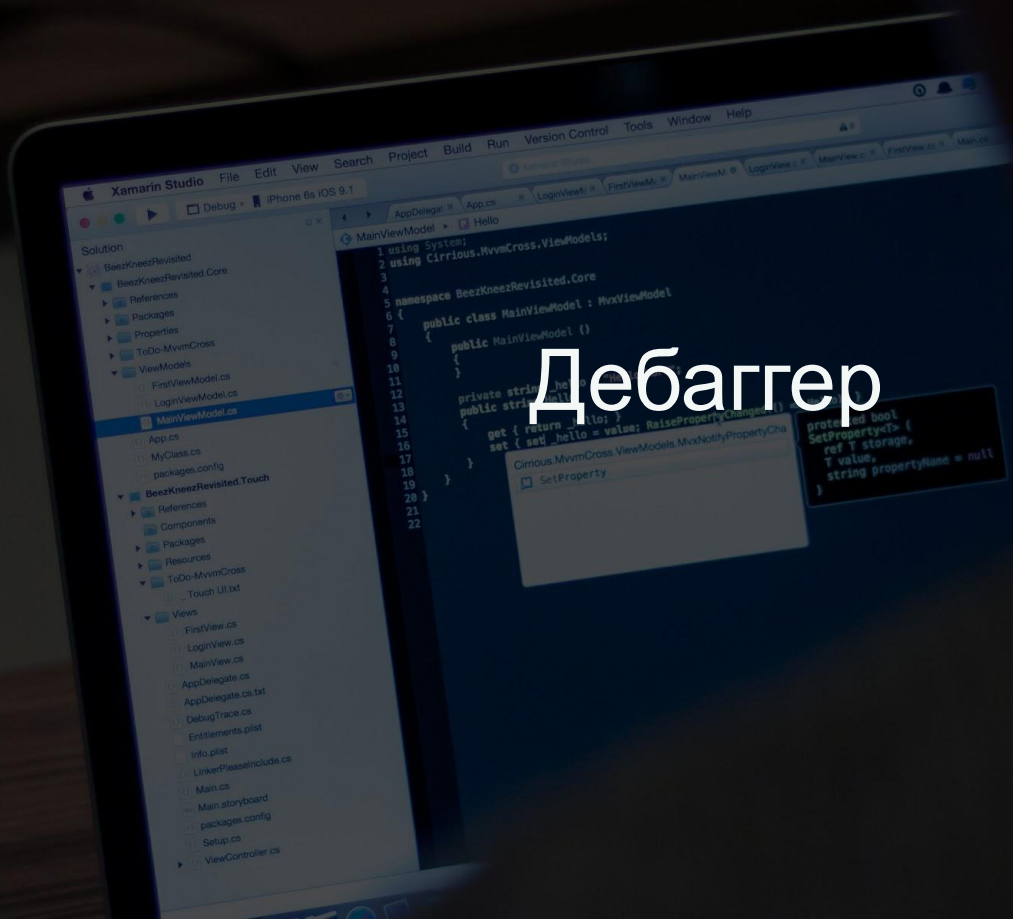
Коллекции в Swift



Тип данных	Описание	Применение
Array	Упорядоченный массив значений. Доступ по индексу	Когда необходимо хранить данные в определенном порядке и извлекать по номеру позиции
Dictionary	Неупорядоченная коллекция <i>ключ-значение</i> . Доступ по ключу	Когда порядок данных не важен, но важно извлекать по какому-либо ключу.
Set	Неупорядоченная коллекция уникальных значений. Быстрый поиск по значению. Операции над множествами	Когда порядок данных не важен, но нужно, чтобы данные были уникальны. Основная задача – проверка, содержит ли коллекция определенное значение



Дебаггер



- Можно поставить точку останова (breakpoint) на любой строке с кодом.
- Можно выполнять программу по шагам.
- Можно посмотреть значение любой переменной на конкретном шаге.



Домашнее задание

Формат файла: «1I_ФИ.playground»

1. Решить квадратное уравнение.
2. Даны катеты прямоугольного треугольника. Найти площадь, периметр и гипотенузу треугольника.
3. *Пользователь вводит сумму вклада в банк и годовой процент. Найти сумму вклада через 5 лет.



Вопросы участников

