Swift 3 Fundamentals

iOS Workshops

www.visual-engin.com

Alberto Irurueta - Pia Muñoz

VISUAL
ENGINEERING
ADVANCED MOBILE
TECHNOLOGY

Overview

- Language Basics
- Playgrounds
- Variables
- Functions
- Optionals
- Control Flow

Language Basics

Language Overview

- No semicolons

```
print(“Hello world!”)
```

- Dot notation. No arrow operator (->).

```
obj1.use(obj2 and: obj3)
```

- C like comments (support for Javadoc like docs)

```
// /* */
```

Basic Types

- Int
- Long
- Float
- Double
- Character
- String

Others:

- Uint8, Int8, Uint16, Int16, Uint32, Int32, Uint64, Int64

Basic operators

- Operators are C like
  - Sum: a + b, a += b
  - Subtraction: a - b, a -= b
  - Product: a * b, a *= b
  - Division: a / b, a /= b
  - Modulus: a % b, a %= b
  - Booleans: a == b, a != b, a && b, a || b, !a
  - Bitwise: a & b, a | b, a ^ b, ~a, a << 1, a >> 1
  - Ternary: a > b ? a : b

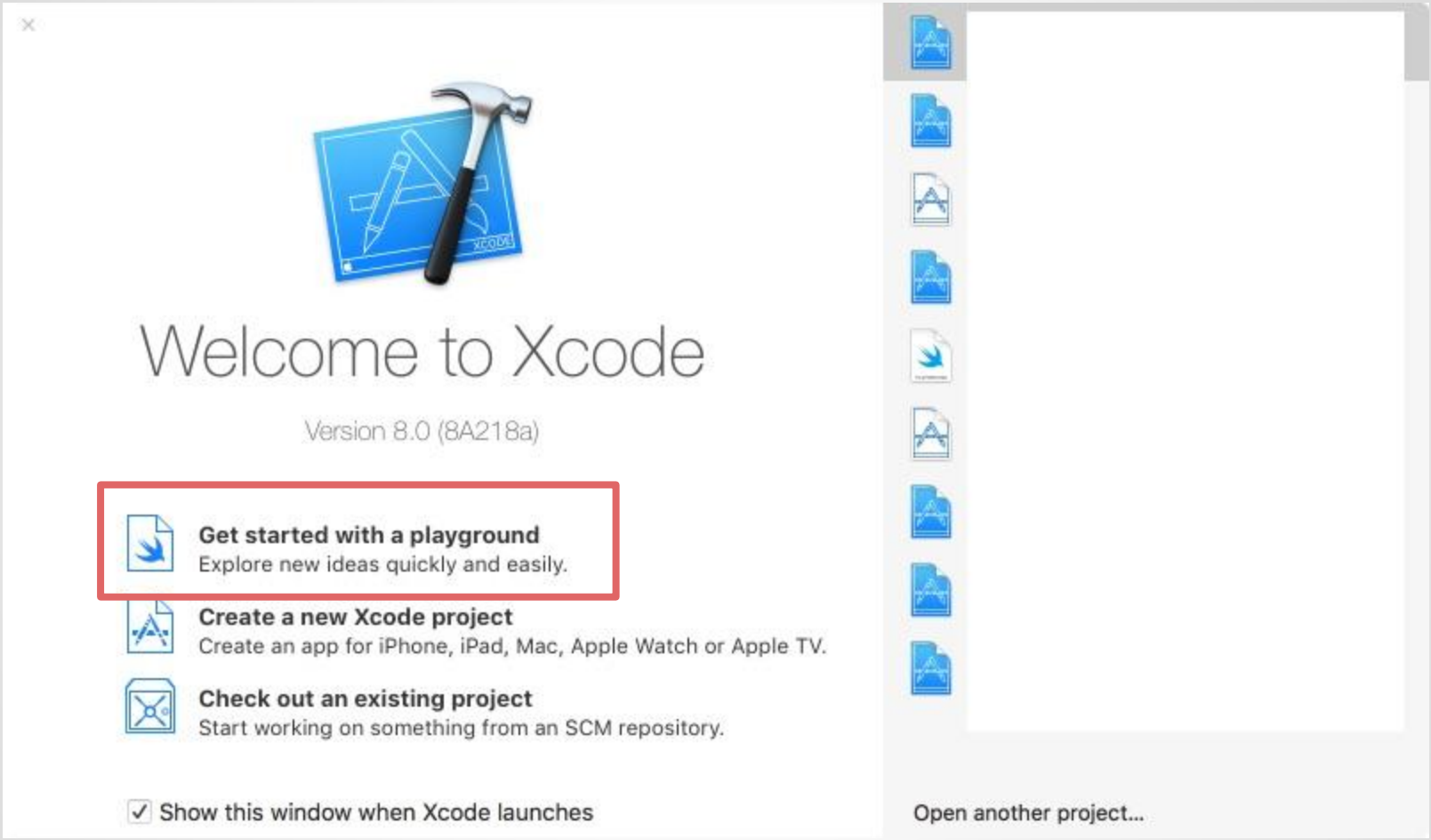
- String concatenation: “hello” + “ world”
- String interpolation: “a + b is \(a+b)”

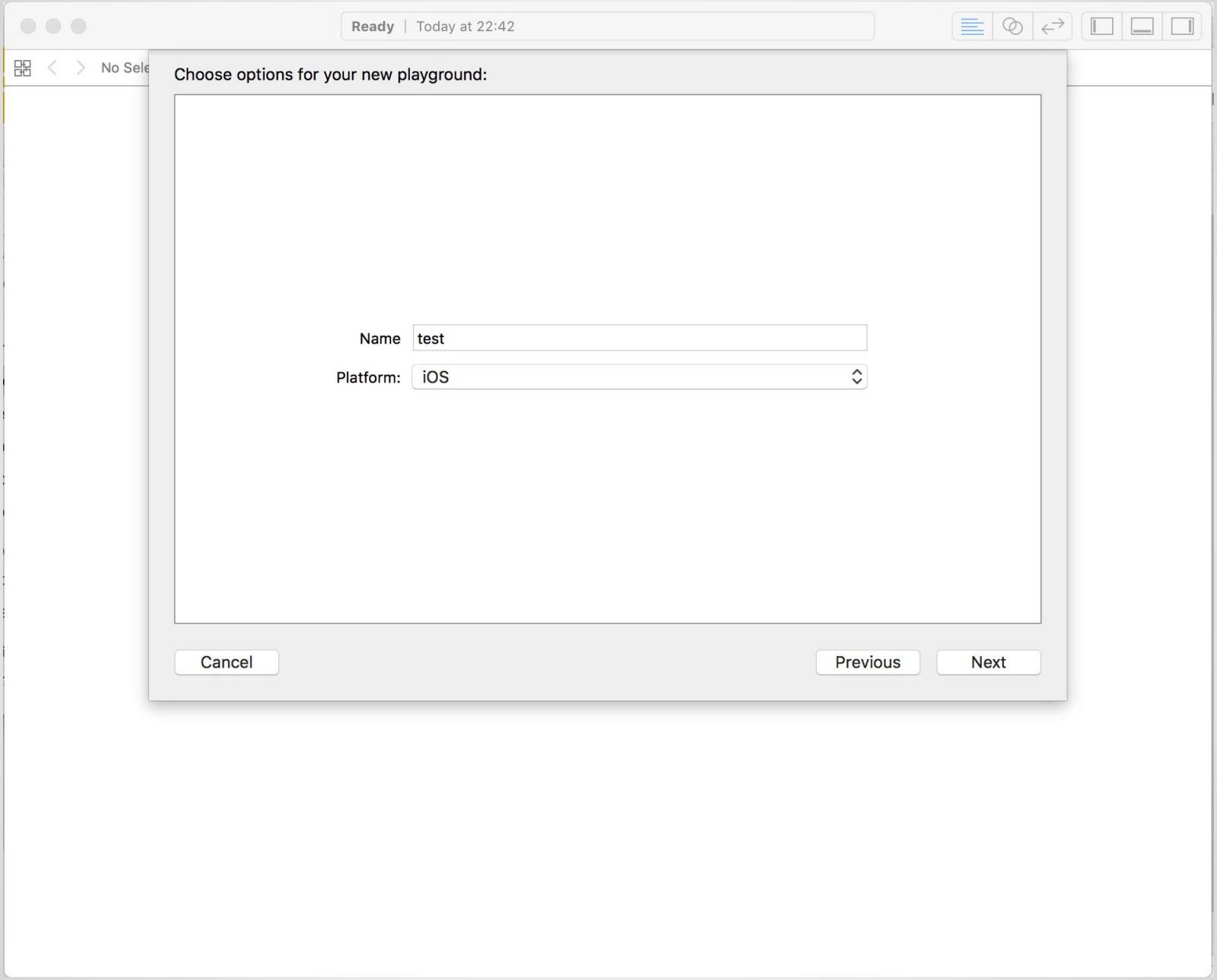
Playgrounds

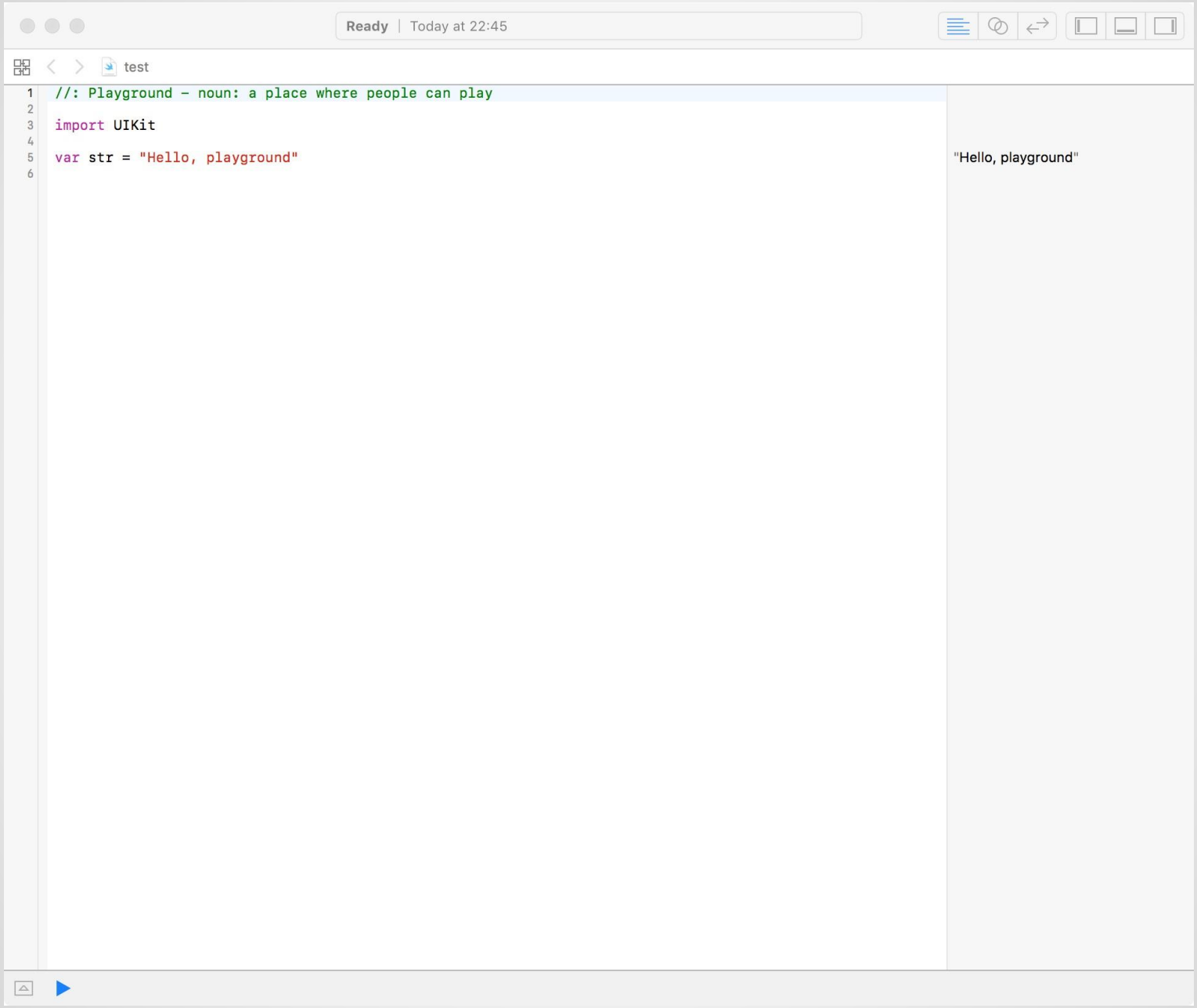
Playgrounds

- Is a simple way to learn and test pieces of code
- Perfect for newbies and students
- Available in:
  - Xcode
  - iPad as a standalone app

https://developer.apple.com/videos/play/wwdc2016/408/


Welcome to Xcode
Version 8.0 (8A218a)
Get started with a playground
Explore new ideas quickly and easily.
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.
Check out an existing project
Start working on something from an SCM repository.
Show this window when Xcode launches
Open another project...


Ready | Today at 22:42
No Sele
Choose options for your new playground:
Name test
Platform: iOS
Cancel
Previous
Next


Ready | Today at 22:45
test
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
"Hello, playground"

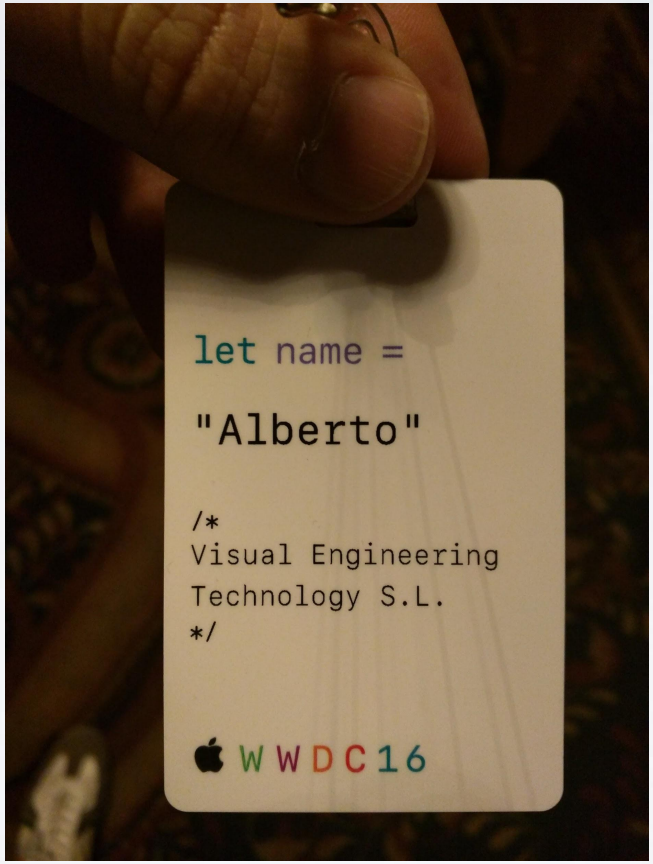
Language Basics

Hands on

Variables

Variables

Let (constants) vs. var (variables)



Variables

Goodbye nil!

Variables

Casting:

```
var decimal: Double = 12.5
var integer: Int = Int(decimal)
```

Inference:

```
let typeInferredDouble = 3.14159

let actuallyDouble2: Double = 3
let actuallyDouble = Double(3)
let actuallyDouble3 = 3 as Double
```

Variables

Arrays:

```
let emptyArray = [String]()
```

Dictionaries:

```
let emptyDictionary = [String: Float]()
```

Context

Instances only live within their context

```
{
   var a = 1
   {
      var b = 2
      print(“b is \(b)”)
   }
   var b = 3
   print(“b is \(b)”)
}
```

Variables

Hands on

Functions

Functions

Functions are types!

Definition

```
func name(_ param: Int) -> Int {
...
}
```

Type

```
(Int) -> Int
```

Functions

Functions can have named parameters like objective C

Definition

```
func name(p param: Int) -> Int {
...
}
```

Type

```
(p: Int) -> Int
```

Functions

Functions can be nested

```
func func1() -> Void {
   print("func1")

   func func2(){
      print("func2")
   }

   func2()
}
```

Functions

Sometimes you need to import

```
import UIKit

var a = max(1,2)
```

Functions

Hands on

Optionals

Optionals

Nil is not gone…

```
var a: Int?
var b: Int? = 3
var c: String? = nil
var d: String = “hello”
```

Providing default

```
let a: String? = nil
let b: String = “class”
let c = “Hi \(a ?? b)”
let d = a!= nil ? a! : b
let e = a??b
```

Unwrapping optionals

Execution only when optional is defined

```
var a: String? = "hello"
if let b = a {
   //a is defined
}
```

Chained evaluation

```
a?.someProperty?.someOtherProperty
```

Forced unwrapping

```
var b = a!.someProperty
```

Optionals

Hands on

Control Flow

Control Flow

Hands on

switch

```
switch variable {
   case “var1”:
      print(“variable is var1”)
   case “var2”:
      print(“variable is var2”)
   default:
      print(“variable is something else”)
}
```

for-in

```
for score in scores {
   //Being scores an array
   //And score each object in the array
}

for i in 0…<4 {
   //It enters 4 times
   //with values i = 0, 1, 2 and 3
}
```

while and repeat-while

```
while n < b {
    n *= 2
}

repeat {
    m *= 2
}while m < b
```

VISUAL
ENGINEERING
ADVANCED MOBILE
TECHNOLOGY

www.visual-engin.com
info@visual-engin.com

Passeig de Gràcia, 67 1º 2ª
(08008) Barcelona. España
T. : (+34) 93 215 77 35

Linked in
www.linkedin.com/companies/visual-engineering

twitter
www.twitter.com/visualengin