

Curso de programación para iPhone

Nivel I - 03. Programando Objective-C

Instructor: René Cruz Flores

Un poco de historia

- 1972 - Lenguaje C-ANSI
- 1980 - Smalltalk Programación Orientada a Objetos
- 1983 - Objective-C
- 1988 - NeXT [Licencia de Objective-C para NeXT OS]
- 1996 - Apple compra NeXT [Aparece Java]
- 2002 - OS X
- 2007 - iPhone OS, Objective-C 2.0
- 2008 - iPhone OS 2.0, iPhone SDK
- 2009 - iPhone OS 3.0
- 2010 - iPhone OS 4.0, iPad SDK

Lenguajes Basados en C

C++

```
#include <iostream>
int main() {
    std::cout << "Hola, mundo!\n";
}
```

Java

```
public class Hola {
    public static void main(String args[]) {
        System.out.println("Hola, mundo!");
    }
}
```

C#

```
class Hola {
    static void main() {
        System.Console.WriteLine("Hola, mundo!");
    }
}
```


Objective-C vs Lenguaje C

C

```
#include <stdio.h>
int main(int argc, const char *argv[]) {
    printf("Hola, mundo!\n");
    return 0;
}
```

Objective-C

```
#import<Foundation/Foundation.h>
int main(int argc, const char *argv[]) {
    NSLog(@"Hola, mundo!");
    return 0;
}
```

Objective-C Formal

```
#import<Foundation/Foundation.h>
int main(int argc, const char *argv[]) {
    NSAutoReleasePool *pool = [[NSAutoReleasePool alloc] init];
    NSLog(@"Hola, mundo!");
    [pool drain];
    return 0;
}
```


Cocoa Touch

- Foundations Framework
 - Incluye las clases esenciales y datos primitivos (NSString, NSInteger, etc.)
- UIKit Framework
 - Contiene las clases que representan a los componentes gráficos de la interfase de usuario.
- Address Book UI
- Game Kit
- Map Kit
- Core Audio
- Core Graphics
- ...

Vistazo a Objective-C

1. Crear un proyecto para iPhone llamado Test en Xcode y examinar el código del archivo AppDelegate.h y AppDelegate.m

2. Crear un proyecto Objective-C nativo llamado SimpleTest:


1. New Project -> Command Line Utility -> Foundation Tool
2. Ejecutar y ver los resultados en la consola

Ejercicio

1. Crear un proyecto Command Line, llamado **ClimaInfo**.
2. Analizar su funcionamiento
3. Revisar la documentación para la clase NSDate
4. Ejecutar y sacar conclusiones

Especificadores de Formato

```
NSLog(@"Hoy 19/02/09, en Mexicali la temperatura es de 40 Grados!");
```



```
NSLog(@"Hoy %@, en %@ la temperatura es de %d Grados!", today, ciudad, temperatura);
```


Llamada a metodos

Otros Lenguajes:

```
MiObjeto.metodoX();
```

```
resultado = MiObjeto.metodoX();
```

Objective-C:

```
[MiObjeto metodoX];
```

```
resultado = [MiObjeto metodoX];
```


Parámetros

Otros Lenguajes:

```
MiObjeto.metodoX(parametro);
```

```
resultado = MiObjeto.metodoX(parametro);
```

Objective-C:

```
[MiObjeto metodoX:parametro];
```

```
resultado = [MiObjeto metodoX:parametro];
```


Parámetros múltiples

Otros Lenguajes:

```
MiObjeto.inserta("Dato", 0);  
  
resultado = MiObjeto.recupera("Dato", 0);
```

Objective-C:

```
[MiObjeto insertaString:@"Dato" enIndice:0];  
  
resultado = [MiObjeto recuperaString:@"Dato" deIndice:0];
```

El verdadero nombre del metodo es:

`recuperaString:deIndice:`

Crear clases propias

1. Crear un proyecto Command Line llamado CustomClass.
2. Agregar una nueva clase Cocoa Touch Class -> Objective-C class llamada "Persona".
3. Declarar los atributos Edad y Nombre.
4. Crear dos objetos de tipo Persona, asignar sus respectivos valores.
5. Imprimir los datos de cada objeto en consola.

Compilar y ejecutar

1. Abrir el proyecto “CustomClassError”
2. Compilar
3. Revisar los errores (Build Results)
4. Corregir los errores

Repaso de Orientación a Objetos

1. ¿Qué es una clase?
2. ¿Qué elementos tiene una clase?
3. ¿Cómo se usa una clase?
4. ¿Qué es un objeto?
5. Escribir la clase, crear un objeto, usar el objeto, liberar el objeto.
6. APHE?
 - Abstracción
 - Polimorfismo
 - Herencia
 - Encapsulamiento

Práctica

Crear un proyecto Command Line con el siguiente propósito:

1. Manejar los datos de una agenda de contactos básica.
2. Por lo menos manejar 5 contactos diferentes.
3. Los datos importantes de cada contacto son: Nombre, Email y Teléfono.
4. Cada dato, deberá asignarse por medio de un método específico (un método por dato).
5. Los datos de la agenda deberán de imprimirse en la consola.