

CESAR Summer School - iOS

User Defaults



User Defaults

A hand holding a smartphone with a dark overlay and the text 'User Defaults'. The background is a blurred image of a hand holding a smartphone, with the screen displaying various app icons like Mail, Calendar, Photos, App Store, Weather, and Music. The text 'User Defaults' is written in a large, white, sans-serif font across the center of the image.

Tipos

Bool

Int

String

Double

Data

Dictionary

Array

scoreRecord

327

nomeUsuario

“Seu Antonio”

modoEscuro

true

Se deletar o app, também deleta todos os dados do UserDefaults



Lanterna RGB


```

import UIKit

class ViewController: UIViewController {

    @IBOutlet var sliders: [UISlider]!
    @IBOutlet weak var tomada: UISwitch!

    override func viewDidLoad() {
        super.viewDidLoad()

    }

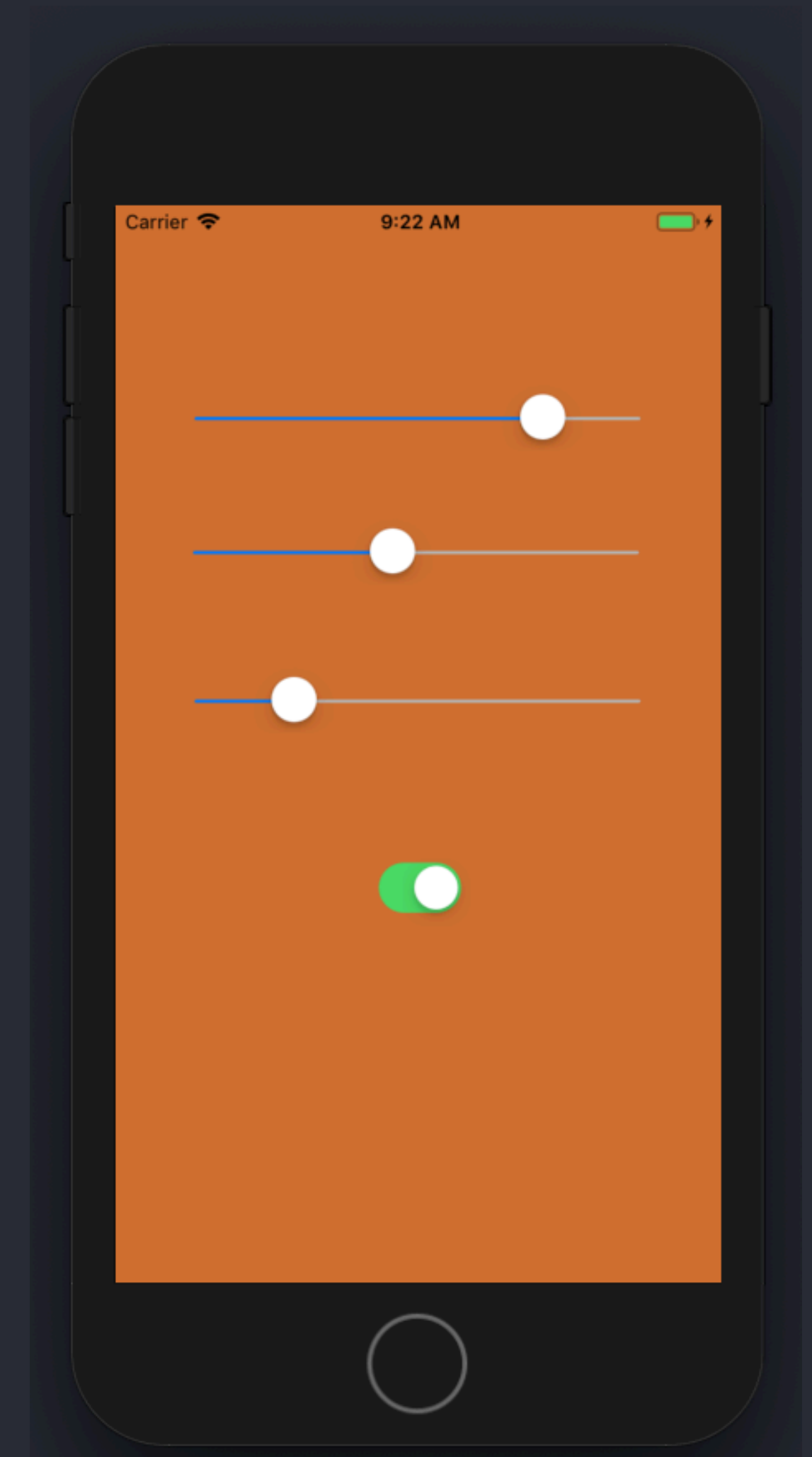
    @IBAction func mudarSlider(_ sender: UISlider) {
        atualizarValor()
    }

    @IBAction func mudarSwitch(_ sender: UISwitch) {
        atualizarValor()
    }

    func atualizarValor(){
        if tomada.isOn{
            view.backgroundColor = UIColor(red: CGFloat(sliders[0].value),
green: CGFloat(sliders[1].value), blue: CGFloat(sliders[2].value), alpha: 1)
        }else{
            view.backgroundColor = UIColor.black
        }
    }

}

```



Crie uma classe para o UserDefaults


```
class DadosSetados{  
    let defaults = UserDefaults.standard  
  
    static let shared: DadosSetados = DadosSetados()  
  
    var sliderR: Float {  
        get{  
            return defaults.float(forKey: "sliderR")  
        }  
        set{  
            defaults.set(newValue, forKey: "sliderR")  
        }  
    }  
}
```

```
import UIKit

class ViewController: UIViewController {

    @IBOutlet var sliders: [UISlider]!
    @IBOutlet weak var tomada: UISwitch!

    let dadosDefaults = DadosSetados.shared

    override func viewDidLoad() {
        super.viewDidLoad()

        sliders[0].value = dadosDefaults.sliderR

        atualizarValor()
    }

    @IBAction func mudarSlider(_ sender: UISlider) {

        switch sender.tag {
        case 0:
            dadosDefaults.sliderR = sender.value
        default:
            break
        }

        atualizarValor()
    }

    @IBAction func mudarSwitch(_ sender: UISwitch) {

        atualizarValor()
    }

    func atualizarValor(){

        if tomada.isOn{
            view.backgroundColor = UIColor(red: CGFloat(sliders[0].value), green: CGFloat(sliders[1].value), blue: CGFloat(sliders[2].value), alpha: 1)
        }else{
            view.backgroundColor = UIColor.black
        }
    }

}
```

```
class DadosSetados{

    let defaults = UserDefaults.standard

    static let shared: DadosSetados = DadosSetados()

    var sliderR: Float {
        get{
            return defaults.float(forKey: "sliderR")
        }
        set{
            defaults.set(newValue, forKey: "sliderR")
        }
    }

    var sliderG: Float {
        get{
            return defaults.float(forKey: "sliderG")
        }
        set{
            defaults.set(newValue, forKey: "sliderG")
        }
    }

    var sliderB: Float {
        get{
            return defaults.float(forKey: "sliderB")
        }
        set{
            defaults.set(newValue, forKey: "sliderB")
        }
    }

    var tomada: Bool {
        get{
            return defaults.bool(forKey: "tomada")
        }
        set{
            defaults.set(newValue, forKey: "tomada")
        }
    }

}
```

Completando os dados

```
import UIKit

class ViewController: UIViewController {

    @IBOutlet var sliders: [UISlider]!
    @IBOutlet weak var tomada: UISwitch!

    let dadosDefaults = DadosSetados.shared

    override func viewDidLoad() {
        super.viewDidLoad()

        sliders[0].value = dadosDefaults.sliderR
        sliders[1].value = dadosDefaults.sliderG
        sliders[2].value = dadosDefaults.sliderB
        tomada.isOn = dadosDefaults.tomada

        atualizarValor()
    }

    @IBAction func mudarSlider(_ sender: UISlider) {

        switch sender.tag {
        case 0:
            dadosDefaults.sliderR = sender.value
        case 1:
            dadosDefaults.sliderG = sender.value
        case 2:
            dadosDefaults.sliderB = sender.value
        default:
            break
        }
        atualizarValor()
    }

    @IBAction func mudarSwitch(_ sender: UISwitch) {

        dadosDefaults.tomada = sender.isOn
        atualizarValor()
    }

    func atualizarValor(){

        if tomada.isOn{
            view.backgroundColor = UIColor(red: CGFloat(sliders[0].value), green: CGFloat(sliders[1].value), blue: CGFloat(sliders[2].value), alpha: 1)
        }else{
            view.backgroundColor = UIColor.black
        }
    }

}
```

Teste as modificações

Como seria pra salvar um objeto?

```
class Jogador: Codable{  
  
    var nome: String  
    var selecao: String  
    var dataImage: Data  
    var numero: Int  
  
    init(nome: String, selecao: String, dataImage: Data, numero: Int) {  
        self.nome = nome  
        self.selecao = selecao  
        self.dataImage = dataImage  
        self.numero = numero  
    }  
}
```


UserDefaults

```
var jogadores: [Jogador]{  
    get{  
        let data = defaults.value(forKey: "jogadores") as! Data  
        let jogadoresSalvos = try! PropertyListDecoder().decode(Array<Jogador>.self, from:  
data)  
        return jogadoresSalvos  
    }  
    set(novosJogadores){  
        let encodedArray = try! PropertyListEncoder().encode(novosJogadores)  
        defaults.set(encodedArray, forKey: "jogadores")  
    }  
}
```

```
extension UIImage {  
    var data: Data? {  
        if let data = self.jpegData(compressionQuality: 1.0) {  
            return data  
        } else {  
            return nil  
        }  
    }  
}
```

```
extension Data {  
    var image: UIImage? {  
        if let image = UIImage(data: self) {  
            return image  
        } else {  
            return nil  
        }  
    }  
}
```

Leitura:

```
cell.imagemJogador.image = self.dados.jogadores[indexPath.row].dataImage.image
```

Escrita:

```
let jogador3 = Jogador(nome: "Van Persie", selecao: "Holanda", dataImage:  
imageLiteral(resourceName: "Robin_Van_Persie1").data!, numero: 9)
```