

# Advanced Topics in iOS & Swift

Wednesday, 09/26/16



# OpenWeatherMap API



```
class OpenWeatherMapAPI {  
    // ...  
}
```

- passing API key in *initialiser*
- 2 main *public* function calls to create the API for the library
  - today's weather
  - x day forecast



# OpenWeatherMap API



```
class OpenWeatherMapAPI {  
    // ...  
}
```

- what *properties* should the class have?
- how to return response data?
- what kind of *model* objects?



# OpenWeatherMap API



## Requirements:

```
class OpenWeatherMapAPI {  
    // ...  
}
```

Our library should...

... provide easy access to **today's weather**

... provide easy and flexible access to the **forecast for the next x days**

... return **temperature values** (min, max, avg) in Fahrenheit (default), Celsius or Kelvin

... return **short description** (e.g. "clear sky")

# Generics<T> in Swift

- *generics* is a fancy word for *type parameters* (rather than passing values, we are passing types as arguments into a function)
- allows to write functions that can be called with different argument types
- large parts of the Swift Standard Library are written using generics (e.g. **Arrays** and **Dictionaries**)

# { Coding Challenge }

1. write a function that swaps the first and the last element of an **Int** array

```
func swapFirstAndLast(array: [Int]) -> [Int]
```

2. write a function that swaps the first and the last element of a **String** array

```
func swapFirstAndLast(array: [String]) -> [String]
```

# { Coding Challenge }

1. write a function that swaps the first and the last element of an **Int** array

```
func swapFirstAndLast(array: [Int]) -> [Int]
```

2. write a function that swaps the first and the last element of a **String** array

```
func swapFirstAndLast(array: [String]) -> [String]
```

# { Coding Challenge }

1. write a function that finds the first occurrence of an element in an array and returns its index

```
func findFirst<T>(element: T, array: [T]) -> Int
```

2. write a function that deletes the first occurrence of an element in an array:

```
func deleteFirst<T: Equatable>(element: T, array: [T])  
-> [T]
```

3. [optional] make these an extension of **Array**



# Wrapping a Web API

## **Goals**

- No direct networking calls (unnecessary complexity)
- Strongly typed return data (rather than untyped JSON)
- No string-based interaction
- Not dealing with building URLs

# Using **enums** over String

**General goal:** no *magic* strings or numbers

```
let url = URL(string: "http://weathermap.org?q=London")
```

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 10  
}
```

