Swift Developer Cheat-Sheet

Types

Integer:	Int
Floating point:	Float
Double precision:	Double
Boolean:	Bool
String:	String

Array:
Dictionary:

```swift
let array: [String] = ["Apple", "Banana", "Cherry"]
let dictionary: [String: Int] = ["Apple": 1, "Banana": 2]
```

Tuples

```swift
// Defining a tuple
let http404Error = (404, "Not Found")
// Accessing elements of a tuple
print("The status code is \(http404Error.0)")
print("The status message is \(http404Error.1)")

// Decomposing tuples
let (statusCode, statusMessage) = http404Error
print("The status code is \(statusCode)")
print("The status message is \(statusMessage)")

// Named elements in tuples for readability
let http200Status = (statusCode: 200, description: "OK")
print("The code is \(http200Status.statusCode) and the status is
    \(http200Status.description)")
```

Built-in functions

Switch statement

```swift
let someValue = 5
switch someValue {
case 1...4:
    print("Between 1 and 4")
case 5:
    print("Exactly 5")
default:
    print("Something else")
}
```

Optionals / force unwrapping

```swift
var optionalString: String? = "An optional string"
// Optional Binding
if let string = optionalString {
    print(string)
}
// Force Unwrapping
print(optionalString!)
```

Struct

```swift
struct Point {
    var x: Int
    var y: Int
}
let point = Point(x: 10, y: 20)
```

While loop

```swift
var count = 5
while count > 0 {
    print(count)
    count -= 1
}
```

Enum

```swift
enum CompassPoint {
    case north, south, east, west
}
var direction = CompassPoint.west
```

For-in loop

```swift
for item in array {
    print(item)
}
```

Class example

```swift
class Product {
    var name: String
    var price: Double

    // Constructor (Initializer)
    init(name: String, price: Double) {
        self.name = name
        self.price = price
    }

    // Destructor (Deinitializer)
    deinit {
        print("\(name) is being deinitialized")
    }

    // Method to describe the product
    func describe() {
        print("Product: \(name), Price: \(price)")
    }

    // Operator Overloading
    static func + (left: Product, right: Product) -> Product {
        let combinedName = "\(left.name) & \(right.name) Bundle"
        let combinedPrice = left.price + right.price
        return Product(name: combinedName, price: combinedPrice)
    }
}
```

Type to enter text