

# Introduction to Swift

## Lesson 1 - Individual Worksheet

---

### Understanding Types

Write the type of the constant written in the "Type" column. If there's an error with the code, instead write "Error: " and what the error is in the column.

Code	Type
<code>let a: Int = 5</code>	
<code>let b = 8</code>	
<code>let c = -6.7</code>	
<code>let d: Double = 5</code>	
<code>let e = "Catz!"</code>	
<code>let f: Int = 7.8</code>	
<code>let g = "5"</code>	
<code>let h: Double = "98.7"</code>	
<code>let i = 9 + 7</code>	
<code>let j = 9.8 + 3</code>	
<code>let k: Int = 9.5 - 0.5</code>	

---

## Adding Clarity

Read the prompts below and write at least three constants you would create for the programs being made.

Make sure the names clearly represent what values they hold for the context. The values do not have to be the exact values in the context, but they should make sense type-wise (distances should be Ints or Doubles, names should be Strings).

Example: `let distanceToTarget = -800`

1. The prime factors of 13195 are 5, 7, 13 and 29. Create a program to find the largest prime factor of the number 600851475143.
2. Create a program where the user can play Sudoku from.
3. Create a program that will solve your math homework for you by inputting in the problem.
4. Create Siri.

## Evaluating Code - Compilation Errors

Below is some Swift code. Some lines may produce compiler errors.

First mark all of the lines which will produce compiler errors with asterisks.

Then, please rewrite **any** combination of all of the available lines of code, so that the entire snippet compiles. The line numbers on the left should help you.

In addition, try to maintain the original intent of the code.

```
1> // We compute what angle to fire a cannon at to hit a target.
2> // Given parameters:
3> let distanceToTarget = 800 // The target is 800 meters away
4>
5> let cannonMuzzleSpeed = 524.256 // The shot leaves at about
   524 m/s
6>
7>
8> // Computation: First we compute the intermediate value
9> let distanceSin: Int = 9.8 * distanceToTarget /
   (cannonMuzzleSpeed * cannonMuzzleSpeed)
10>
11> // Then we compute the angle to fire the cannon at
12> let angle = asin(distanceSin) / 2
13>
14> // Also, we compute how long it will take to hit the target
15> let horizontalSpeed = cos(angle) * cannonMuzzleSpeed
16>
17> let timeOfFlight = distanceToTarget / horizontalSpeed
```

---

## Evaluating Print Statements

Below is some Swift code. It compiles fine.

Below it, please write what you think the code will print. If you haven't specifically learned what some code means, that's fine, you should still try to guess what it will do!

```
1> let fruit1 = "banana"
2> let fruit2 = "apple"
3> let fruit3 = "🍊"
4>
5> let fruit1_num = 1
6> let fruit2_num = 12.3
7> let fruit3_num = 6
8>
9> let name = "Helen"
10>
11> print("\(name), please go to the store for me.")
12> print("I need \(fruit1_num) \(fruit1)s, \(fruit2_num)
    \(fruit2)s, and \(fruit3_num) \(fruit3)s.")
```