QND Computer Science Day 1

Mark Schmidt

Agenda

- Learning philosophy
- Programming Languages
- Swift
- Tools
- Your first (actual) program

Learning Philosophy

- You will not master everything today
- You will (probably) be lost
- That's okay!
- Practice, practice, practice
- Help each other!

Machine Code

- Hard to write and understand
- Different computer types use different machine code

```
.text:00000000 sub:
                        push
                               ebp
.text:00000001
                               ebp, esp
                        mov
.text:00000003
                               eax, [ebp+8]
                        mov
.text:00000006
                               ecx, [ebp+0Ch]
                        mov
.text:00000009
                              eax, [ecx+eax*2]
                        lea
.text:0000000C
                               ebp
                        pop
.text:000000D
                        retn
.text:00000010 main:
                              ebp
                        push
.text:00000011
                        mov
                               ebp, esp
.text:00000013
                        push
                               ecx
.text:00000014
                               eax, [ebp+0Ch]
                        mov
                               ecx, [eax+4]
.text:00000017
                        mov
.text:0000001A
                        push
                               ecx
.text:0000001B
                             dword ptr ds:__imp__atoi
                        call
.text:00000021
                        add
                              esp, 4
.text:00000024
                               [ebp-4], eax
                        mov
.text:00000027
                               edx, [ebp-4]
                        mov
.text:0000002A
                        push
                               edx
.text:0000002B
                               eax, [ebp+8]
                        mov
.text:0000002E
                        push
                               eax
.text:0000002F
                            _sub
                        call
.text:00000034
                        add
                              esp, 8
.text:00000037
                               esp, ebp
                        mov
.text:00000039
                              ebp
                        pop
                                                   88
.text:000003A
                        retn
```

Programming Languages

- Compile to machine code
- Many, many different languages

Swift

- Programming Language created by Apple
- Used for creating iOS and macOS applications
- Simple and modern!



My Website

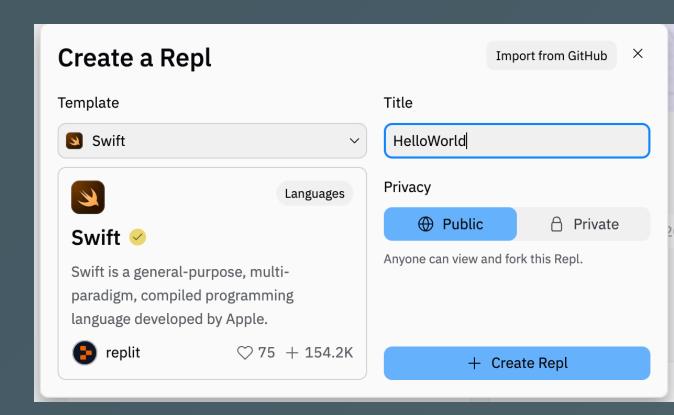
- Everything you need is at markschmidt.io/qnd
- Presentation slides
- Code links
- Livestream

Replit

- We need a coding environment
- **Replit** is an online Integrated Development Environment
- Skip installation/package management/local device issues

Setup

- 1. Go to Replit.com
- 2. Create an account (use your Google account)
- 3. Press "+" in the top right
- 4. Select **Swift** as a template
- 5. Title **Hello World**
- 6. Press "Create REPL"



Hello World!

Type the following into the code editor:

```
print("Hello World!")
```

Then tap the **Run** button on top

You should see Hello World! in the console below

Strings

- A sequence of characters
- Characters
- Always between quotation marks

The Print Function

- print("Hello 🧩 🧩 \\")
- Prints the provided string to the console output
- Add emoji with control + command + spacebar
- We will see many, many other functions
- function(input1, input2, input3...)

Repitition

• Our program can have multiple lines!

```
print("Hello ***")
print("Hello ***")
print("hello from Swift!")
print("hello from Swift!")
print("hello from Swift!")
```

Variables

- *Declare* variables with let
- Just like in math, replaces the variable with a value
- Read it as "Let greeting equal ..."

```
let greeting = "Hello ***!"
print(greeting)
print(greeting)
print(greeting)
```

Concatenation

• Combine variables together!

```
let greeting = "Hello "
let name = "Mr. Schmidt"
print(greeting + name)
```

This program is boring!

- It needs to respond to user input
- Use the readLine()! function!
 - Note the !
- Waits for the user to type in the console and press Enter
- Stores the value in a variable

```
print("What is your name?")
let name = readLine()!
```

Use concatenation to say hello!

One More Thing

- Comments + Spacing
- Empty lines can be ignored
- Anything after a // is ignored by the program

```
// Ask the user for their name
print("What is your name?")
let name = readLine()!

// Note the space after Hello
let greeting = "Hello "
print(greeting + name)
```

Extra Challenges

- Print the greeting 3 times
- Ask a user for their name and their favorite color. Print each out 3 times.
- Print empty lines between each line of output
 - What happens if we pass "" to print?