Swift

Variables & Constants

Use the 'var' keyword to specify a variable
Use the 'let' keyword to specify a constant
Use constants whenever possible to improve readability, thread safety, and compiler optimization
Type Inference

- Swift is strongly typed
- Swift can often infer types contextually

Collection Types

- Standard library provides Array, Dictionary, and Set
- Collections allow any types
- Subscript syntax and operator support
- Generic support
Control Statements

- while
- for
- repeat/while
- for in
- if/else
- switch

Tuples

- Any number of values
- Any combination of types
- Supports decomposition
- Supports named values
Optionals

Nil values not allowed by default, optionals can be nil
Types specified with '?'
Explicit compile time reasoning about nil values
Use if let statements with '?' operator to unwrap safely
Force unwrap with '!' operator; program will crash if a nil value is unwrapped
Optional chaining

Implicitly Unwrapped Optionals

Can contain a nil value
Types specified with '!'  
Implicitly unwrapped (i.e., can be used as normal without an unwrap operator)
Program will crash if a nil value is unwrapped
Useful when a value must start out as nil, but will always be initialized before use
Documentation

“The Swift Programming Language”