```
for index in 1...5 {
}
```

```
for _ in 1...n {
}
```

```
for x in xs {
}
```

```
for (key, value) in dictionary {
}
```

```
while `condition` {
    `statements`
}
```

```
repeat {
    `statements`
} while `condition`
```

- NOTE: review special features of switch statements in Swift
- NOTE: review "labeled" while loops
- NOTE: review enumeration case patterns and their use in the Optional Pattern

```
// Optional Pattern
if case let x? = someOptional {
}
```

```
/* 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 */
for var i = 0; i < 10; i++ {
    print(i)
}

// becomes:
for i in 0..<10 {
    print(i)
}
```

```
/* 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 */
for var i = 10; i > 0; i-- {
    print(i)
}

// becomes:
for i in (1...10).reversed() {
    print(i)
}
```

```
/* 0, 2, 4, 6, 8 */
for var i = 0; i < 10; i += 2 {
    print(i)
}

// NOTE: Strideable's stride method was replaced by a global stride function
// becomes:
for i in stride(from: 0, t0: 10, by: 2) {
    print(i)
}
```

- NOTE: additionally review the global sequence functions

```
let someNumbers = [2, 3, 45, 6, 8, 83, 100]
/* 2, 3, 45, 6, 8, 83, 100 */

// instead of this
for var i = 0; i < someNumbers.count; i++ {
    print(someNumbers[i])
}

// use this
for number in someNumbers {
    print(number)
}

// or this
someNumbers.forEach { number in
    print(number)
}
```

```
let someNumbers = [2, 3, 45, 6, 8, 83, 100]

/* 100, 83, 8, 6, 45, 3, 2 */

// instead of this
for var i = someNumbers.count - 1; i >= 0; i-- {
    print(someNumbers[i])
}

// use this
```

```
for number in someNumbers.reverse() {
    print(number)
}
```

```
let someNumbers = [2, 3, 45, 6, 8, 83, 100]

/*
 1: 2
 2: 3
 3: 45
 4: 6
 5: 8
 6: 83
 7: 100
 */

// instead of this
for var i = 0; i < someNumbers.count; i++ {
    print("\(i + 1): \(someNumbers[i])")
}

// use this
for (index, number) in someNumbers.enumerated() {
    print("\(index + 1): \(number)")
}

// or this
someNumbers.enumerated().forEach { (index, number) in
    print("\(index + 1): \(number)")
}
```

```
let someNumbers = [2, 3, 45, 6, 8, 83, 100]

/* 0, 1, 2, 3, 4, 5, 6 */

// instead of this
for var i = 0; i < someNumbers.count; i++ {
    print(i)
}

// use this
for index in someNumbers.indices {
    print(index)
}
```

- also map, flatMap, filter, and reduce