



# Chapter 6/7/8

## Operators

### Ranges

### Ternary Operator

### Nil Coalescing

### Loops

### Guard

### Switch

## Operators

### Ranges

```
x...y // closed range    (x and y are inclusive)
x..<y // half-open range (x inclusive, but not y)
...x  // one-sided range (all values <= x)
x...  // one-sided range (all values >= x)
```

### Ternary Operator

```
cond ? true_expr : false_expr
```

### Nil Coalescing

- allows a **default value** to be used in the event that an optional has a nil value

$$Statement = \begin{cases} lhs & value = nil \\ rhs & otherwise \end{cases}$$

```
let customer: String? = nil
print(customer ?? "User")
```

## Loops

```
for _ in 1..5 {
    // code to be executed
}

var i = 10
repeat {
    print(i)
    i -= 1
} while (i > 0)
```

## Guard

- unwrapping is *in-scope* of its outer-least scope

```
func foo(value: Int?) {
    guard let number = value, number < 10 else { // executes if expression is false
        print("number is too high")
        return // exit-statement
    }

    print(number * 10) // executes if expression is true
}
```

## Switch

- case-statements do not fall-through

```
let value = 15

switch (value) {
    case 0..
```

```
        print("below 20")
        fallthrough // next case-statement will proceed
    case 10..<30:
        print("below 30")
    case 31, 32, 33:
        print("vowel")
    default:
        break
}
```