Operators

```
1 + 2
"a" + "b"
2 - 3
2 * 3
20 / 7
20 / 7.0
20 % 7
1 == 2
"asdf" == "asdf"

class A {}
let obj1 = A()
let obj2 = obj1
let obj3 = A()
obj1 === obj2
obj1 === obj3

1 > 2
13 >= 1
99 < 3
14 <= 5
1 != 2
true || false
true && false
!false

1...3
1..<4
```

Built-in types

Value types

```
let integerNumber: Int = 42
//other: Int8, Int16, Int32, Int64, UInt...

let floatNumber: Float = Float(5.3)
//other: Float32, Float64

let doubleNumber: Double = 5.6
let booleanValue: Bool = true // false
let stringValue: String = "I'm a string"
```

Collection types

```
let namesArray: [String] = ["George", "Paul", "John"]
```

```
let numbersArray: [Int] = [124, 2, 8, -35]
let numbersDictionary: [Int : String] = [1 : "one", 14 :
"fourteen", 3 : "three"]
let numbersDictionary2: [String : Int] = ["two" : 2,
"nine" : 9, "eight" : 8]
let uniqueNames: Set<String> = Set<String>(["Paul",
"John", "John"])
let uniqueNumbers: Set<Double> = [3.5, 2.0, 6.1, 19, 6.1]

print(namesArray[0])
print(numbersDictionary[1])
print(numbersDictionary2["two"])

//print(uniqueNames[0])
print(uniqueNames.contains("Janusz"))
print(uniqueNumbers.first)
```

Optional type

```
let nameOrNil: Optional<String> = "It's a name after all"
print(nameOrNil)

var definitelyNil: String?
print(definitelyNil)

print(uniqueNumbers.first!)
let emptyArray: [Int] = []
//print(emptyArray.first!)
```

Let vs var

```
let constantString: String = "I'm a constant string"
//constantString = "uncomment me"

var variableString: String = "I can change"
variableString = "See? I told you I can!"

var calculatedNumber: Int {
    return 2 * 3
}
print(calculatedNumber)

var interpolatedString: String {
    return "Today's lucky number is \(calculatedNumber)"
}
print(interpolatedString)
```

Control flow

If - else

```
let trueOrFalse: Bool = true

if trueOrFalse {
     print("It's true")
} else {
     print("It's false")
}

if 1 > 2 {
     print("It's greater")
} else {
     print("Not really")
}

let isGreater = (2 > 1) ? true : false
```

For - in

```
for _ in 0..<10 {
}

var sum: Int = 0
for index in 0..<10 {
     sum += index
}
print(sum)

for element in [1, 2, 3, 4] {
    print(element)
}

for element in Set<Int>([1, 2, 3, 4]) {
     print(element)
}

for (key, value) in numbersDictionary {
    print("number: ", key, " numberString: ", value)
}
```

While loop

```
var shouldLoop: Bool = true
var counter: Int = 10
while(shouldLoop) {
```

```
    print("Loooping")
    counter -= 1
    shouldLoop = (counter != 0)
}
```

Repeat - while

```
shouldLoop = true
counter = 10
repeat {
    counter -= 1
    shouldLoop = (counter != 0)
} while(shouldLoop)
```

Switch - case

```
for index in 0..<10 {
    if index < 5 {
        print("Index smaller than 5")
    } else if index == 5 {
        print("Index is equal 5")
        continue
    } else {
        print("Index greater than 5")
        break
    }

    print("okay, next one")
}

for index in 0..<10 {
     switch index {
         case 0:
             print("It was a 0")
         case 1:
             print("It was a 1")
         case 2, 3:
             print("It was either 2 or 3")
         case 4..<9:
             print("It was a number between 2 and 9")
         default:
             print("It was .... a number")
     }
}

let switchName: String = "Patryk"
switch switchName {
    case "Patryk":
        print("Hello Patryk")
    default:
```

```
        print("Hello \(switchName)")
}
```

Optional & optional unwrapping

```
var optionalInt: Int? = 1

print(optionalInt)
print(optionalInt!)

if let shouldBeInt = optionalInt {
     print(shouldBeInt)
}

guard let shouldBeInt2 = optionalInt else
{ fatalError() }
print(shouldBeInt2)
```