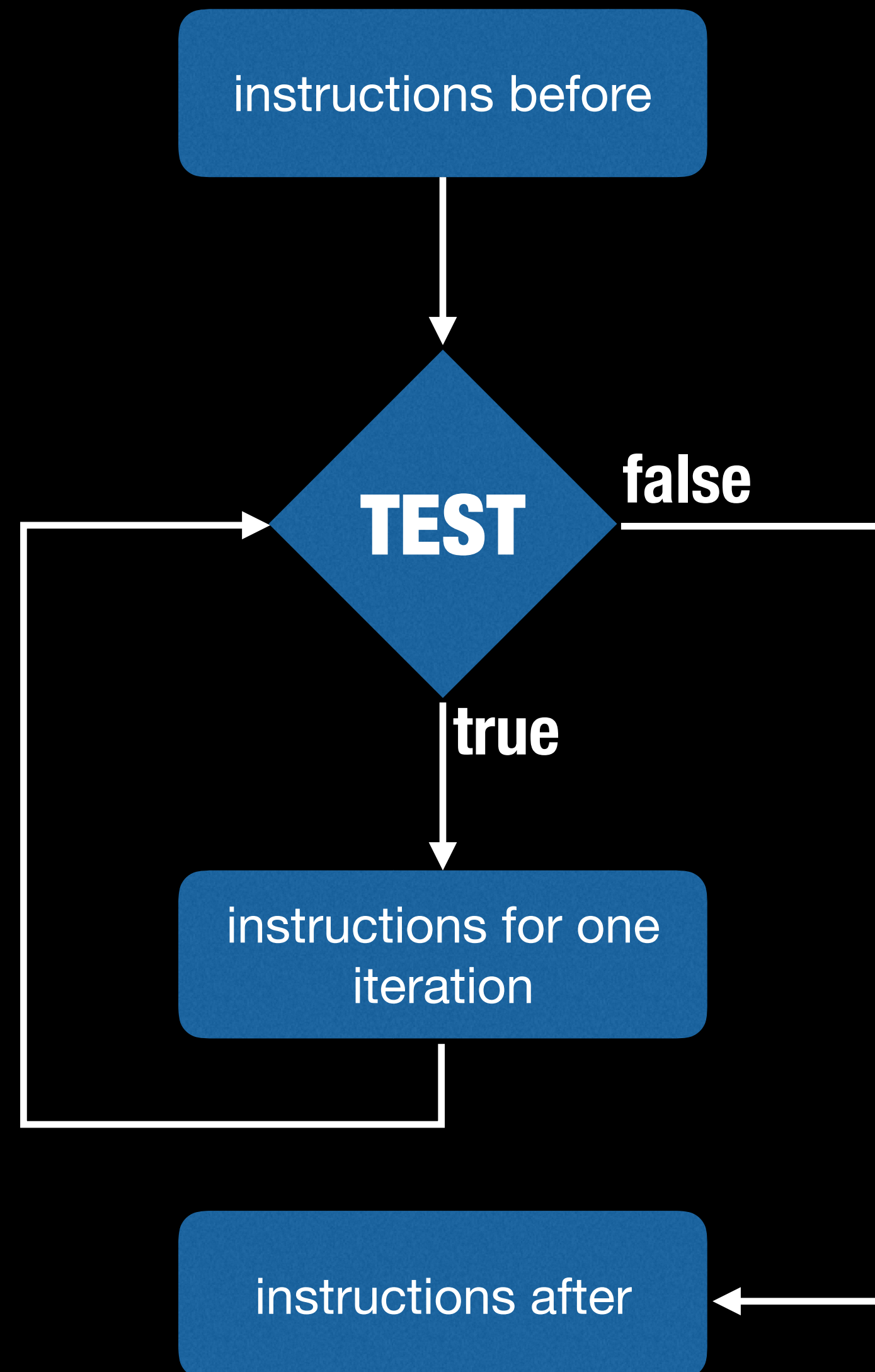


LOOP



while

A while loop performs a set of statements until a test condition becomes false.

```
while test {  
    // if the boolean expression is true  
    // block of code to execute  
    // instructions for one iteration  
}
```

while

A while loop performs a set of statements until a condition becomes false.

```
var i = 1
while i <= 5 {
    let times5 = i * 5
    print("\(i) times 5 is \(times5)")
    i = i + 1
}
```

Results in this output in the console

```
1 times 5 is 5
2 times 5 is 10
3 times 5 is 15
4 times 5 is 20
5 times 5 is 25
```

while

A while loop performs a set of statements until a condition becomes false.

initial condition

```
var i = 1
while i <= 5 {
  let times5 = i * 5
  print("\(i) times 5 is \(times5)")
  i = i + 1
}
```

boolean test
true result means
execute again

condition
development

Results in this output in the console

```
1 times 5 is 5
2 times 5 is 10
3 times 5 is 15
4 times 5 is 20
5 times 5 is 25
```

draw a series of vertical lines across the view

```
var x = 20.0
while x + 20.0 <= tin.width {

    line(x1: x, y1: 50, x2: x, y2: 150)

    x += 40.0

}
```

repeat-while

The repeat-while loop, performs a single pass through the loop block first, before considering the loop's condition. It then continues to repeat the loop until the condition is false.

```
repeat {  
    // block of code to execute  
    // if the boolean expression is true  
} while condition
```

Range operators

Shortcuts for expressing a range of values.

The closed range operator ($a\dots b$) defines a range that runs from a to b , and includes the values a and b . The value of a must not be greater than b .

The values from 1 to 5, including 1 and 5.

$1\dots 5$

The half-open range operator ($a..<b$) defines a range that runs from a to b , but does not include b . It is said to be half-open because it contains its first value, but not its final value. As with the closed range operator, the value of a must not be greater than b . If the value of a is equal to b , then the resulting range will be empty.

Half-open ranges are particularly useful when you work with zero-based lists such as arrays, where it is useful to count up to (but not including) the length of the list

The values from 0 to 4, includes 0 and 4.

$0..<5$

for-in

Use the for-in loop to iterate over a sequence, such as ranges of numbers, items in an array, or characters in a string.

```
for variable in range {  
    // block of code to execute  
    // once for each value in range  
}
```


for-in

Use the for-in loop to iterate over a sequence, such as ranges of numbers, items in an array, or characters in a string.

```
for variable in range {  
    // block of code to execute  
    // once for each value in range  
}
```

```
for index in 1..5 {  
    print("\ (index) times 5 is \ (index * 5) ")  
}  
// 1 times 5 is 5  
// 2 times 5 is 10  
// 3 times 5 is 15  
// 4 times 5 is 20  
// 5 times 5 is 25
```

for-in

Use the for-in loop to iterate over a sequence, such as ranges of numbers, items in an array, or characters in a string.

```
for variable in collection {  
    // block of code to execute  
    // once for each value in collection  
}
```

```
var name = "Tempe"  
for character in name {  
    print(character)  
}  
// T  
// e  
// m  
// p  
// e
```