
FOR INSTRUCTOR PURPOSES ONLY

MATERIALS

- + [Lesson Starter Code](#)
- + [Lesson Solution Code](#)
- + [Lab](#)
- + [Lesson](#)
- + [Lab Solution Code](#)

SWIFT LOOPS

Wellington Moreno

Lead iOS Instructor, General Assembly



LESSON

LEARNING OBJECTIVES

- + **Identify** the purpose of a loop
- + **Explain** the different parts to a for and while loop
- + **Write** and use a for loop
- + **Write** and use a while loop

LESSON

AGENDA

- + Opening Exercise
- + For Loops
- + Practice
- + While Loops
- + Practice

QUESTION

WHAT THE HECK IS A
LOOP? 🐼 □ 🐼 □ 🐼 □

ACTIVITY: DECK OF CARDS



5 mins

DIRECTIONS

Grab a deck of cards.

1. Shuffle the cards
2. Divide the cards amongst the class
3. I will call out a random card (i.e., Jack of Spades)
4. Have the students search through their deck
5. The person who finds it calls out another random card
6. Repeat

DELIVERABLE

Discuss your answers with the person next to you.

QUESTION

WHY IS A LOOP VALUABLE?

CONCLUSION

- + Allows you to repeat a section of code many times.
Write the code once, and have it execute multiple times.
- + It's a more intelligent way of coding.

INTRODUCTION

FOR LOOPS

DEFINITION

A LOOP IS A *SEQUENCE OF INSTRUCTIONS* THAT IS *CONTINUALLY REPEATED* UNTIL A CERTAIN CONDITION IS REACHED.

THE ANATOMY OF A **FOR** LOOP

Type of loop,
keyword



The diagram illustrates the syntax of a for loop. The word **for** is enclosed in a light blue oval, with a black arrow pointing to it from the text 'Type of loop, keyword' located above and to the left. To the right of **for** is the variable **i**, followed by the keyword **in**, then the range **(1...10)**, and an opening curly brace **{**. Below the range **(1...10)** is the text **//code block** in a lighter gray font. The loop is closed by a closing curly brace **}** on a new line.

```
for i in (1...10) {  
    //code block  
}
```

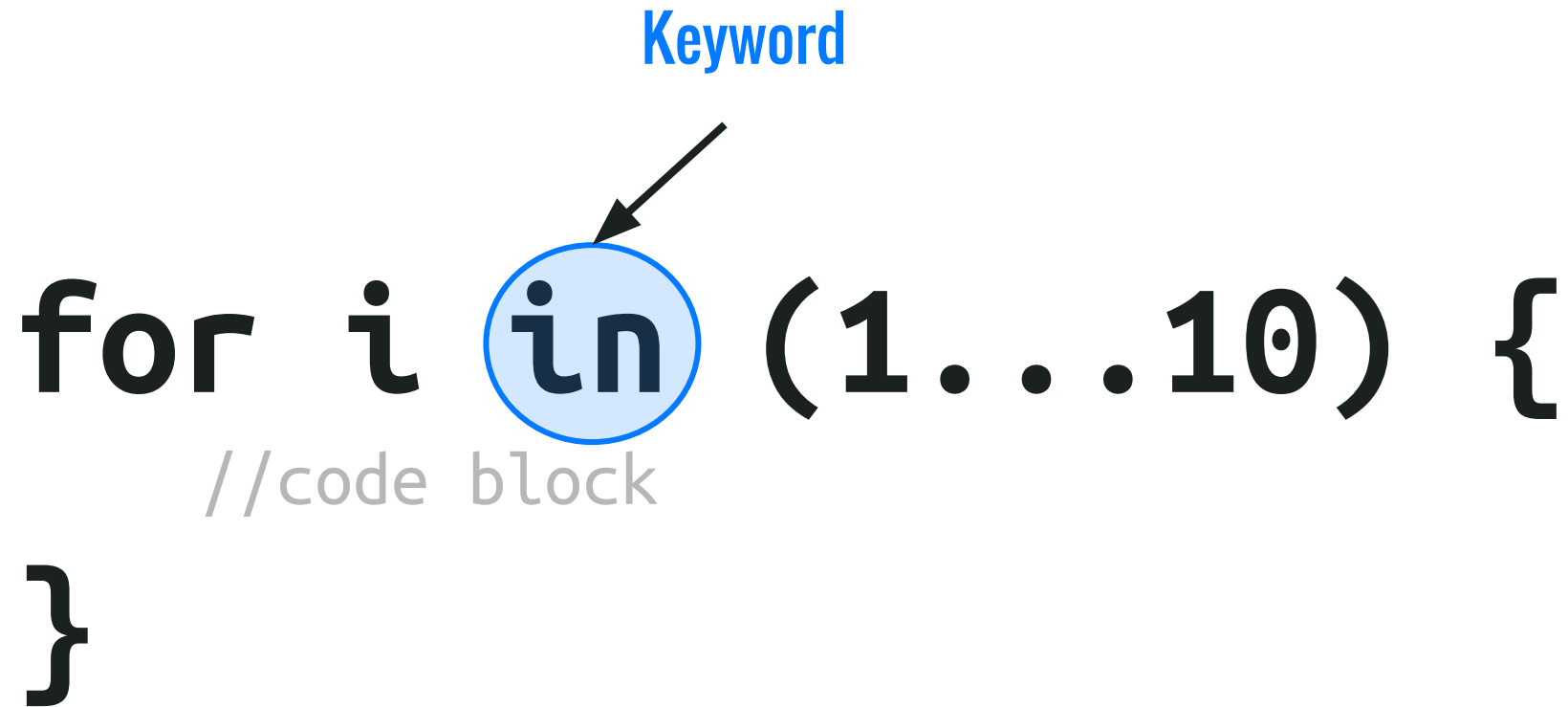
THE ANATOMY OF A **FOR** LOOP

Variable
name
↓
for **i** **in** **(1...10)** **{**
//code block
}

THE ANATOMY OF A **FOR** LOOP

Keyword

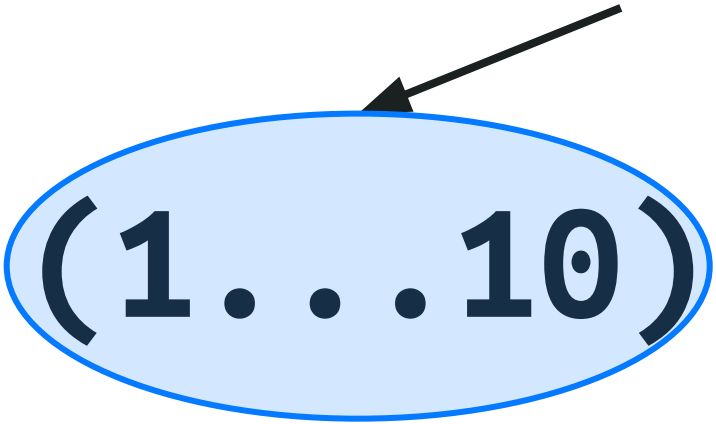
```
for i in (1...10) {  
    //code block  
}
```

The diagram illustrates the syntax of a for loop. The word 'in' is highlighted with a light blue circle, and a blue arrow points to it from the word 'Keyword' above. The variable 'i' is positioned before 'in', and the range '(1...10)' is in parentheses after 'in'. The loop body is enclosed in curly braces '{' and '}', with a comment '//code block' inside. The word 'for' is at the beginning of the loop statement.

THE ANATOMY OF A **FOR** LOOP

Set of Elements

```
for i in (1...10) {  
    //code block  
}
```

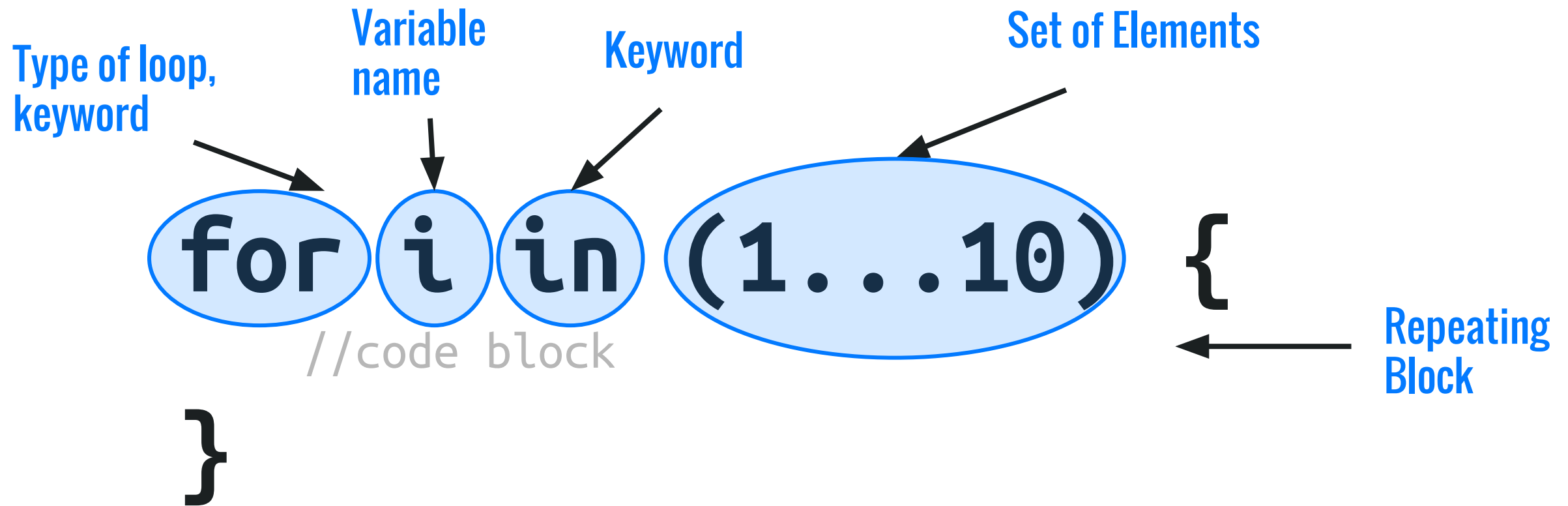
A diagram illustrating the anatomy of a for loop. The code snippet 'for i in (1...10) {' is shown. The expression '(1...10)' is enclosed in a light blue oval. A black arrow points from the text 'Set of Elements' to the top of this oval. Below the opening curly brace, the text '//code block' is written in a lighter gray font, and the closing curly brace is at the bottom.

THE ANATOMY OF A **FOR** LOOP

```
for i in (1..10) {  
    //code block  
}
```

← Repeating
Block

THE ANATOMY OF A **FOR** LOOP



QUESTION

WHAT'S WRONG HERE?

WHAT'S **WRONG?**

```
for i in (1..10) {  
    print(i)  
}
```

WHAT'S **WRONG?**

```
for j in 1...10 {  
    print(i)  
}
```

WHAT'S **WRONG?**

```
for number in 1...100 {  
    print(number)  
    number -= 1  
}
```

WHAT'S WRONG?

```
for _ in (1...10) {  
    print("yo")  
}
```

DEMO

GUIDE ME

- + Find the sum of all the numbers from 1 to 1,000

WHAT'S **WRONG?**

```
for number in 1...100 {  
    print(number)  
}  
number += 1
```

WHAT'S **WRONG?**

```
for i in (1000...1) {  
    print(i)  
}
```


RIGHT!

```
for i in (1...1000).reversed() {  
    print(i)  
}
```

WHAT'S **WRONG?**

```
for i in -100...100 {  
    print(i)  
}
```

WHAT IS THE RESULT?



EXERCISE

2 mins

```
for i in (1...100) {  
    for j in (1...10) {  
        print("\(i)-\(j)")  
    }  
}
```

DELIVERABLE

Stop-And-Jot

INDEPENDENT PRACTICE: PLAYGROUNDS



CODE

7 mins

DIRECTIONS

1. Find the sum of all the numbers between 5 and 9,000 (1 min)
2. Find the sum of all the numbers between -500 and 200 (1 min)
3. Find the sum of all of the even numbers between 1 and 1,000 (1 min)
4. Print all of the multiples of 5 between the numbers 5 and 5,000 (2 min)
5. Compare your results with a partner (2 min)

DELIVERABLES

A Playgrounds file with your answers

LESSON

Q & A - FOR LOOPS

INTRODUCTION

WHILE LOOP

THE ANATOMY OF A **WHILE** LOOP

```
while workRemains {  
    //code block  
}
```

THE ANATOMY OF A **WHILE** LOOP

Type of loop

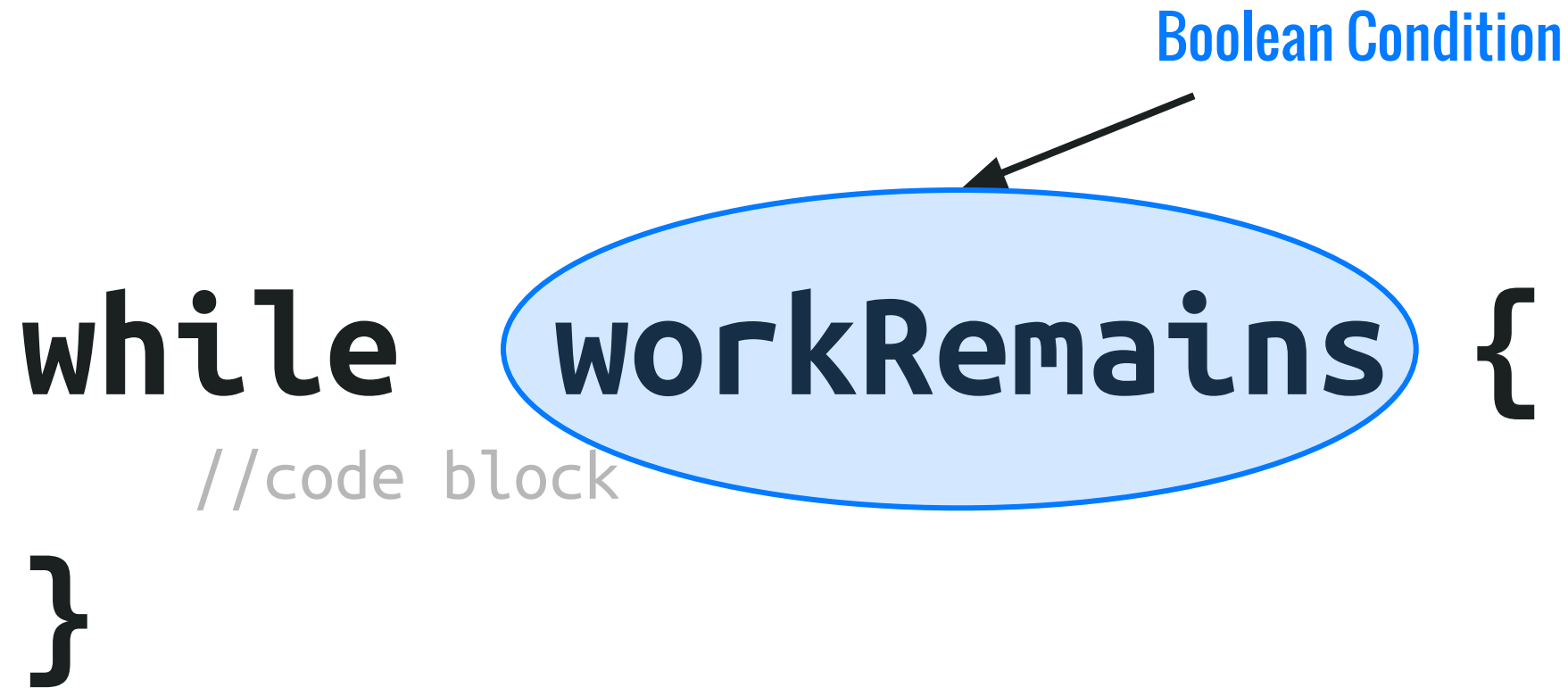


```
while workRemains {  
    //code block  
}
```


THE ANATOMY OF A **WHILE** LOOP

Boolean Condition

```
while workRemains {  
    //code block  
}
```

The diagram illustrates the components of a while loop. The word 'while' is in a large, bold, black font. The condition 'workRemains' is enclosed in a light blue oval with a blue border. A black arrow points from the text 'Boolean Condition' to the top of this oval. The opening curly brace '{' is to the right of the oval. Below the opening brace, the text '//code block' is written in a smaller, gray font. The closing curly brace '}' is at the bottom of the loop structure.

THE ANATOMY OF A **WHILE** LOOP

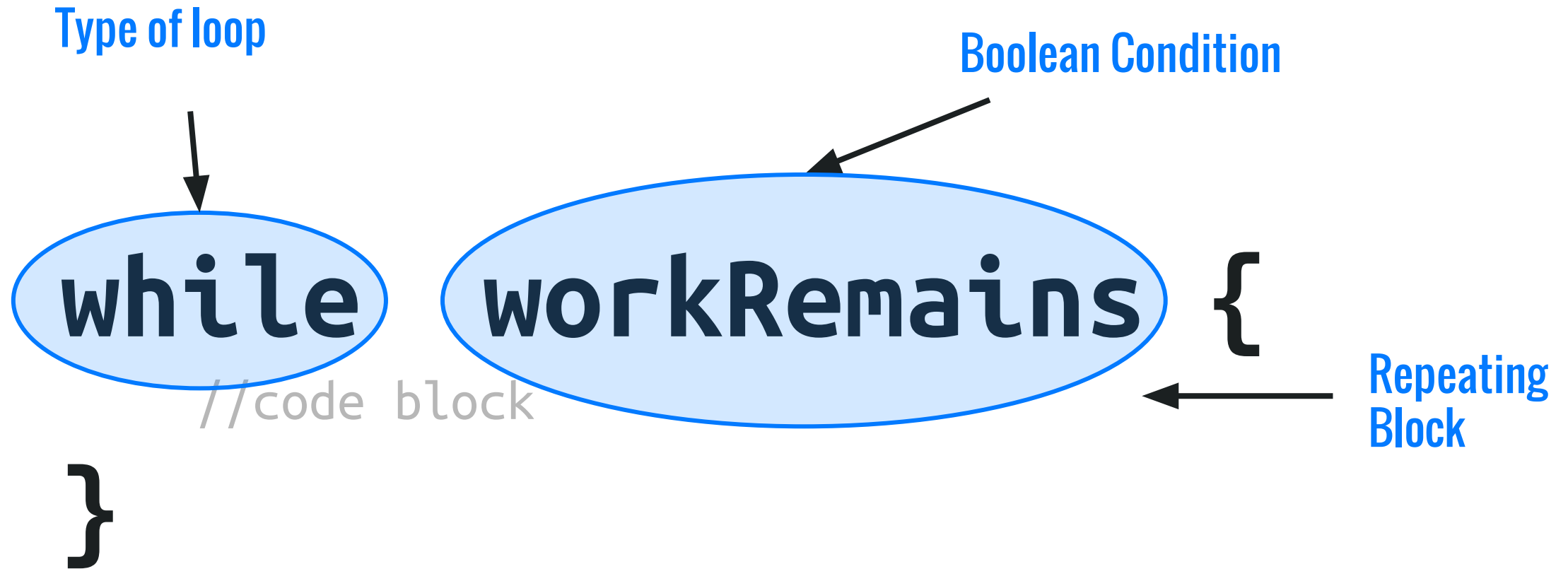
```
while workRemains {  
    //code block  
}
```



Repeating Block

The diagram illustrates the components of a while loop. The code is shown in a monospace font. The word 'while' is in black, followed by a space, then 'workRemains' in black, a space, and an opening curly brace '{'. Below the opening brace is the text '//code block' in a lighter gray font. The closing curly brace '}' is on the next line. To the right of the opening brace, a black arrow points left towards it. To the right of the arrow, the text 'Repeating Block' is written in blue, with 'Repeating' on the top line and 'Block' on the bottom line.

THE ANATOMY OF A **WHILE** LOOP



QUESTION

WHAT'S THE DIFFERENCE?

ACTIVITY: THINK-PAIR-SHARE



2 mins

DIRECTIONS

1. Write your thoughts down. (1 min)
2. Discuss with a partner (1 min)

DELIVERABLE

Insert Deliverable

WHAT'S WRONG?

```
var count = 0
```

```
var number = 1
```

```
while number < 1000 {
```

```
    count += number
```

```
}
```

WHAT'S **WRONG?**

```
var count = 0
```

```
var number = 1
```

```
while number > 0 {
```

```
    count += number
```

```
    number += 1
```

```
}
```

WHAT'S WRONG?

```
var count = 0
```

```
var number = 1
```

```
while number > 1000 {
```

```
    count += number
```

```
    number += 1
```

```
}
```


WHAT'S **WRONG?**

```
var count = 0
var number = 1

while number < 1000 {
    count += number
    number += 1
}
```

RIGHT!

```
var count = 0
var number = 1

while number <= 1000 {
    count += number
    number += 1
}
```

DEMO

GUIDE ME

- + Find the sum of all the numbers between 5 and 9,000
- + Find the sum of all the numbers between -500 and 200

INDEPENDENT PRACTICE: PLAYGROUNDS



CODE

7 mins

DIRECTIONS

1. I have some money, say \$2,000. I want to buy apps until I have no more money left. Each app costs \$7. Use a while loop to determine:
 - a. How many apps I can buy before running out of funds (3 mins)
 - b. How much money is left over after buying all the apps, if any (2 mins)
2. Compare your results with a partner (2 min)

DELIVERABLES

A Playgrounds file with your answers

LESSON

Q & A - WHILE LOOPS

CONCLUSION

- + Useful when you are searching through a list of something and looking for something
- + Useful for performing an action to each element of a collection
- + Useful for doing a task repeatedly

LESSON

BEFORE NEXT CLASS

- + Read up on the "break" keyword
- + Read up on the "continue" keyword

THANKS!

WELLINGTON MORENO

- + GitHub/Slack: @SirWellington
- + Twitter: @SirWellingtonZ
- + Email: wellington.moreno@ga.co