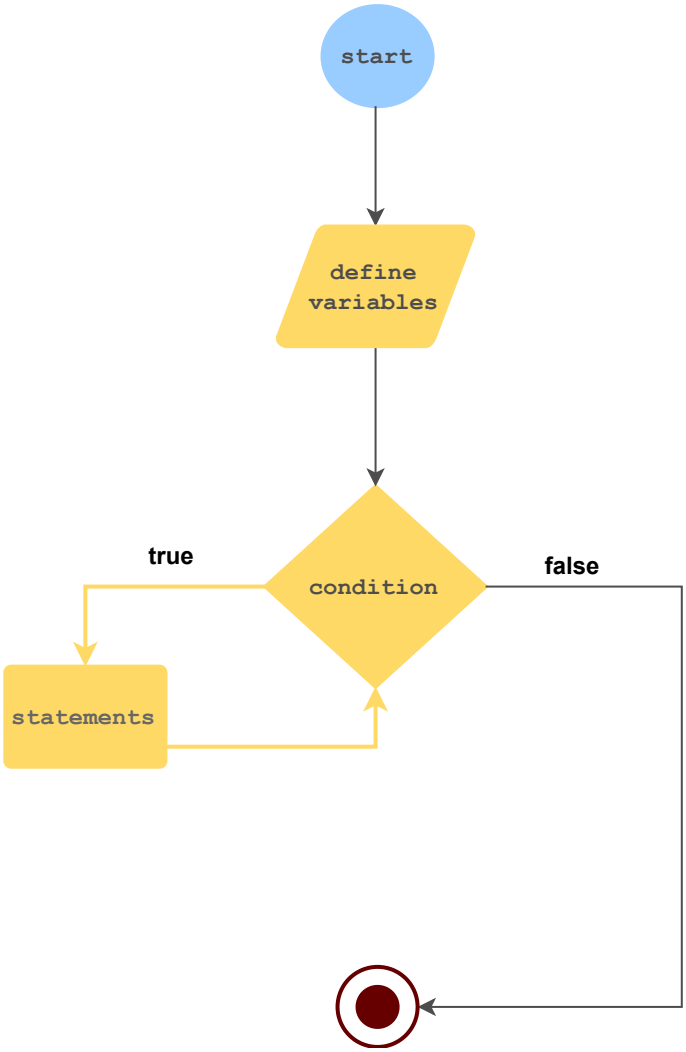
Indefinite Loop - While and Loop

This lesson will discuss two indefinite loop: While and Loop.

We'll cover the following

While loop

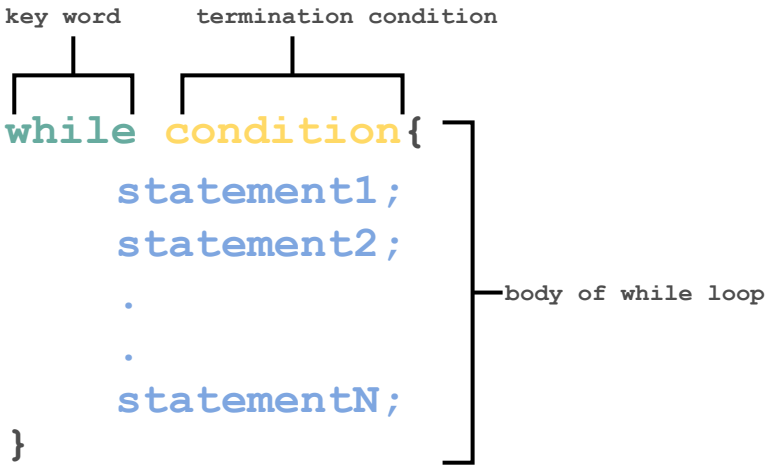
While loop also iterates until a specific condition is reached. However, in the case of a `while` loop, the number of iterations is not known in advance.



Syntax

The `while` keyword is followed by a condition that must be true for the loop to continue iterating and then begins the body of the loop.

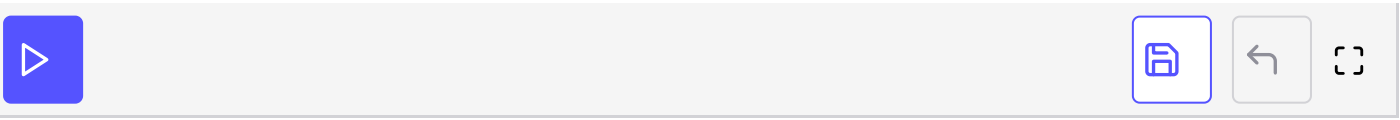
The general syntax is :



Example

The following example makes use of a `while` loop to print a variable value. The loop terminates when the variable's value modulus 3 equals 1.

```
fn main() {
  let mut var = 1; //define an integer variable
  let mut found = false; // define a boolean variable
  // define a while loop
  while !found {
      var=var+1;
      //print the variable
      println!("{}", var);
      // if the modulus of variable is 1 then found is equal to true
      if var % 3 == 1 {
        found = true;
      }
      println!("Loop runs");
  }
}
```



Explanation #

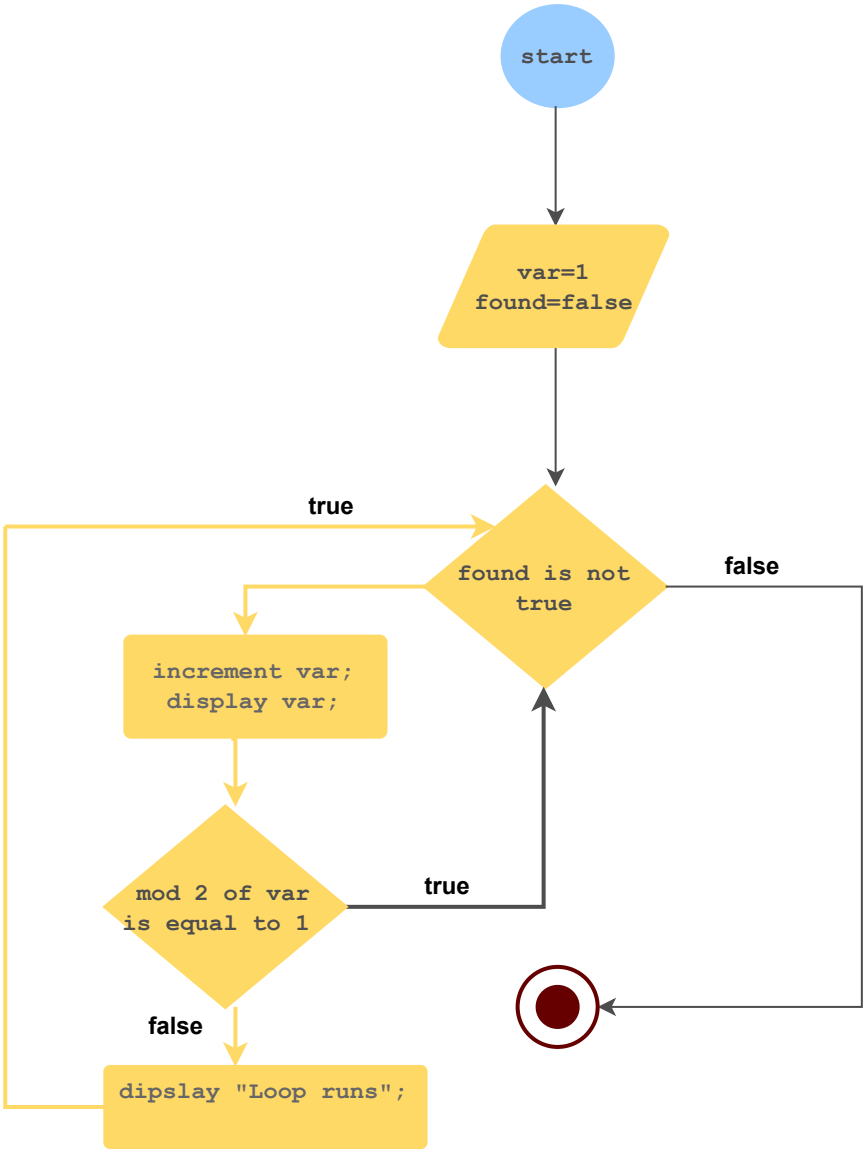
- A mutable variable, `var`, is defined on **line 2**.
- A mutable variable, `found`, is defined on **line 3**.

`while` loop definition

`while` loop is defined on **line 5**. `while` loop is followed by a variable `found`. `found` is initially set to `false`. `!found` means that the loop will continue to iterate until the value of found evaluates to be true.The loop terminates when `found` is set to `true`.
From here the body of the while loop starts

`while` loop body

- The body of the loop is defined from **line 5 to line 14**.
- In each iteration:
  - The value of the variable `var` is incremented by 1 on **line 6** and then printed on **line 8**.
  - If the value of the `var` modulus 3 is equal to 1, then the value of `found` is set to true else it prints "loop runs" on **line 13** and the loop continues.



The following illustration traces the execution of the program:

```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

```
1
```

Output:


```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

```
1     false
```

Output:

```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

1 false

found not equals true

Output:


```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

2 false

Output:

```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

2 false

Output: 2


```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {  2 % 2 =1 => False
      found = true;
    }
    println!("Loop runs");
}
```

2 false

Output: 2

```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

2 false

Output:
```
2
Loop runs
```



```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

found not equals true

2 false

Output:
```
2
Loop runs
```

```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

3 false

Output:
```
2
Loop runs
```



```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

3 false

Output:
```
2
Loop runs
3
```

```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

3 false

3 % 2 =1 => True

Output:
```
2
Loop runs
3
```



```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

3 true

Output:
```
2
Loop runs
3
```

```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

3 true

Output:
```
2
Loop runs
3
Loop runs
```



```
let mut var = 1;
let mut found = false;
while !found {
    var=var+1;
    println!("{}", var);
    if var % 2 == 1 {
      found = true;
    }
    println!("Loop runs");
}
```

3 true

found is true

Loop terminates

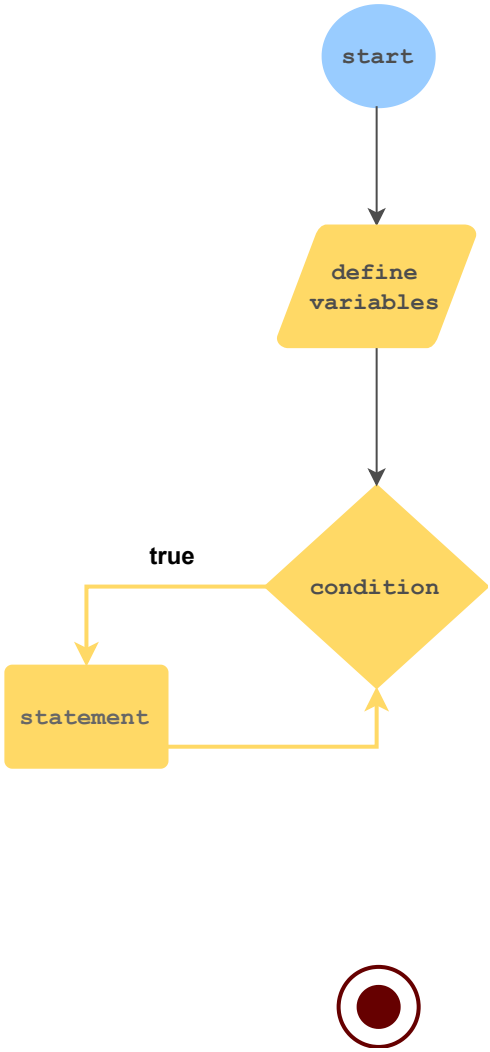
Output:
```
2
Loop runs
3
Loop runs
```

```
    let mut var = 1;
    let mut found = false;
    while !found {
        var=var+1;
        println!("{}", var);
        if var % 2 == 1 {
          found = true;
        }
        println!("Loop runs");
    } end of program code
```

```
Output: 2
        Loop runs
        3
        Loop runs
```


Loop

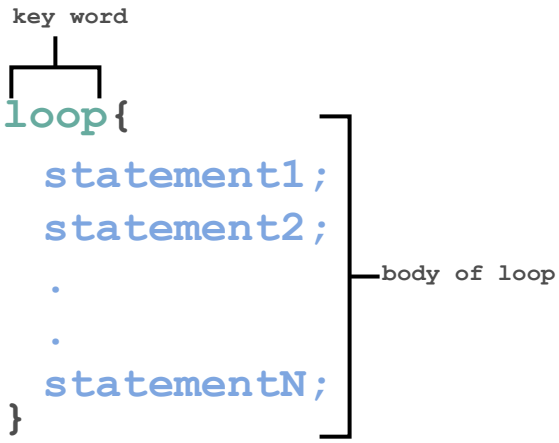
If you want the iteration to continue infinitely, then use the `loop` keyword before the block of code.



Syntax

The `loop` keyword is followed by the body of the loop enclosed within curly brackets `{}`.

The general syntax is :

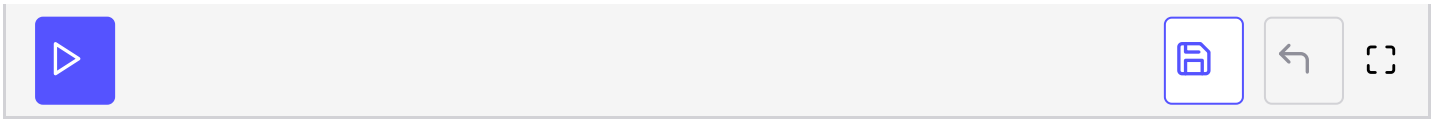


Example

The following example shows how the loop runs infinitely using a `loop`.

> **Note:** The maximum time that is set for the code to run on our platform is 30sec. Since the code below runs more than that, it won't execute here. However, the code will continue to run indefinitely.

```
fn main() {
  //define an integer variable
  let mut var = 1;
  // define a while loop
  loop {
      var = var + 1;
      println!("{}", var);
  }
}
```



Explanation #

- A mutable variable `var` is defined on **line 3**.

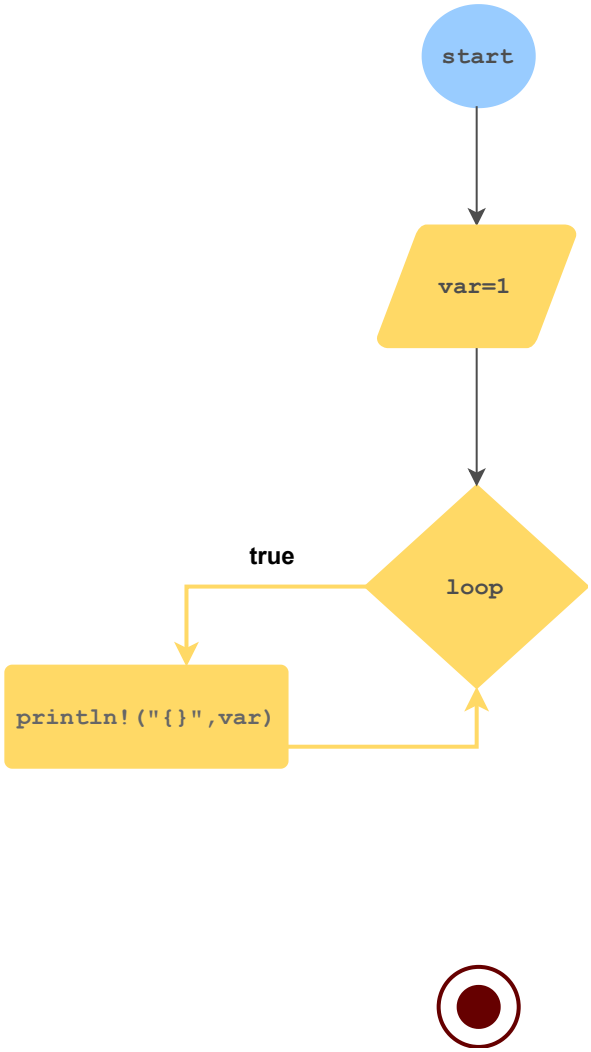
`loop` definition

- `loop` is defined on **line 5**.

From here the body of the loop starts

`loop` body

- The body of the loop is defined from **line 5 to line 8**.
- In each iteration, the value of the variable `var` is incremented by 1 on **line 6** and then printed on **line** 7.
- The loop continues to iterate infinitely.



The following illustration explains the concept:

> **Note:** In the illustration below, only three iterations are shown. However, the loop runs an infinite number of times.

```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

Output:

```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

1

Output:


```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

1

Output: 1


```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

1

Output: 1

```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

2

Output: 1


```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

2

Output: 1
2


```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

2

Output: 1
2

```
let mut var = 1;
  loop {
      var=var+1;
      println!("{}", var);
  }
```

3

Output:
```
1
2
3
```


Loop continues infinite times

Quiz

Test your understanding of indefinite loops.

Quick Quiz on Indefinite Loops!

1 The number of iterations are known in which loop?

2

In a loop that has no terminating condition, which of the following is true?

3

How many times does the loop run?

```
fn main() {
  let mut var = 1; //define an integer variable
  let mut found = false; // define a boolean variable
  // define a while loop
  while !found {
      var=var+1;
      //print the variable
      println!("{}", var);
      // if the modulus of variable is 1 then found is equal to true
      if var % 3 == 1 {
        found = true;
      }
      println!("Loop runs");
  }
}
```

Retake Quiz

What if we want to stop the repetition of an indefinite loop when a certain condition becomes true? Learn about this in the next lesson.