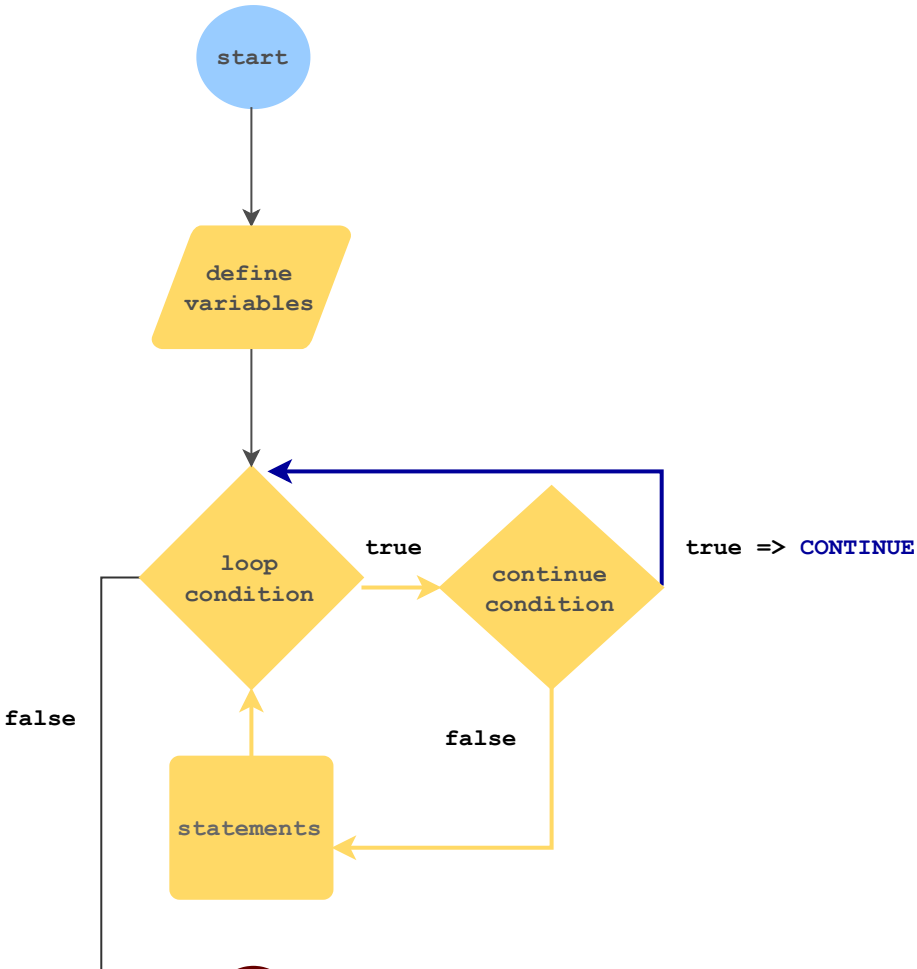
Continue Statement

This lesson will discuss continue statements

We'll cover the following

What Is a `continue` Statement?

The `continue` statement, when encountered inside a loop, skips the execution of the rest of the statements in the loop's body for the current iteration and returns the control to the start of the loop.



The following illustration explains the concept:

```
    for var in 0..5 {
         if var == 2 {
            println!("Continue statement");
            continue;
          }
          println!("var: {}", var)
      }
```

Output:

0


```
    for var in 0..5 {
         if var == 2 {  False
            println!("Continue statement");
            continue;
          }
          println!("var: {}", var)
      }
```

Output:

0

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
   }
```

0

Output: var:0


```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
   }
```

1

Output: var:0


```
for var in 0..5 {
     if var == 2 {  False
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
   }
```

1

Output: var:0

5 of 18

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

1

Output: var:0
var:1

6 of 18

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

2

Output: var:0
var:1

7 of 18

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

True

2

Output: var:0
var:1

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

2

```
Output:  var:0
         var:1
         Continue statement
```



```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;control goes to the start of the loop
      }
      println!("var: {}", var)
  }
```

2

```
Output:  var:0
         var:1
         Continue statement
```

```
    for var in 0..5 {
         if var == 2 {                                    3
            println!("Continue statement");
            continue;
          }
          println!("var: {}", var)
      }
```

```
Output:  var:0
         var:1
         Continue statement
```



```
    for var in 0..5 {
         if var == 2 { False                              3
            println!("Continue statement");
            continue;
          }
          println!("var: {}", var)
      }
```

```
Output:  var:0
         var:1
         Continue statement
```



```
    for var in 0..5 {
         if var == 2 {                                    3
            println!("Continue statement");
            continue;
          }
          println!("var: {}", var)
      }
```

```
Output:  var:0
         var:1
         Continue statement
         var:3
```

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

4

```
Output:  var:0
         var:1
         Continue statement
         var:3
```



```
for var in 0..5 {
     if var == 2 { False
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

4

```
Output:  var:0
         var:1
         Continue statement
         var:3
```



```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

4

```
Output:  var:0
         var:1
         Continue statement
```

```
var:3
var:4
```

16 of 18

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

condition becomes false =>loop breaks

5

```
Output:  var:0
         var:1
         Continue statement
         var:3
         var:4
```

17 of 18

```
for var in 0..5 {
     if var == 2 {
        println!("Continue statement");
        continue;
      }
      println!("var: {}", var)
  }
```

end of program code

```
Output:  var:0
         var:1
         Continue statement
         var:3
         var:4
```

18 of 18

Using With a `for` Loop

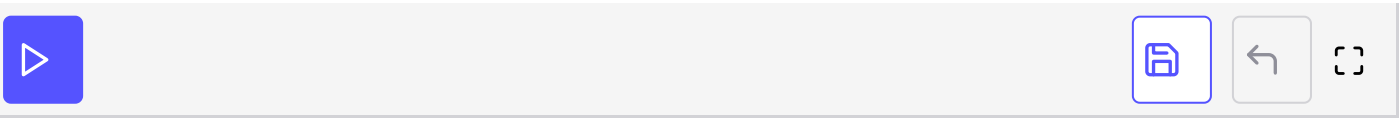
Below is an example of a `continue` expression, using a `for` loop.

- The range defined in the `for` loop is from 0 to 10 with `var` variable used for iterating over the loop
- Within the `for` loop:

- The value of `var` is printed

- When the value of `var` is equal to 4, the control goes to the start of the loop
- The loop executes until the upper bound for the defined range is reached

```
fn main() {
  // define a for loop
  for var in 0..10 {
     if var == 4 {
        println!("I encoutered a continue statement");
        continue;
      }
      println!("var: {}", var);
      println!("I did not encounter continue statement");
  }
}
```



Using With a `while` Loop

Below is an example of `continue` expression, using a `while` loop.

- A mutable variable `var` is defined
- A boolean variable `found` is defined
- Within the `while` loop body:
  - The value of `var` is printed
  - When the value of `var` is equal to 4, the control goes to the start of the loop.
  - The loop executes until the value of `found` does not equal true.

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

      if var == 4 {
          println!("I encoutered a continue statement");
          continue;
        }
        println!("I did not encounter continue statement");
```

```
        if var == 10{
          found = true;
        }
    }
}
```




Using With a `loop`

Below is an example of `continue` expression, using a `loop` .

- A mutable variable `var` is defined
- A boolean variable `found` is defined
- Within the `loop` body:
  - The value of `var` is printed
  - When the value of `var` is equal to 4, the control goes to the start of the loop
  - The loop executes infinitely

> **Note:** This code widget will give an error, ✕, due to limitations of our platform but on the local machine, it will run an infinite loop.

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

     if var == 4 {
        println!("I encoutered continue statement");
        continue;
      }
      println!("I did not encounter continue statement");
  }
}
```




Quiz

Test your understanding of `continue` statement in Rust.

Quick Quiz on Continue Statement.

Q

How many times is the statement “I did not encounter continue statement” printed in the code below?

```
fn main() {
      let mut var = 1;
      let mut found = false;
      while !found {
          var = var + 1;
          println!("{}", var);

          if var == 5 {
              println! ("I encoutered a continue statement");
              continue;
            }
          println!("I did not encounter continue statement");

           if var == 6 {
               found = true;
           }
        }
}
```

Retake Quiz

In the next lesson, you'll learn about nested loops.