1. Introduction to level 2 Python

1.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

1.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS/PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

1.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

1.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

1.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

2. Creating lists

2.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

2.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS/PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

2.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

2.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

2.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

3. Accessing list values

3.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

3.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

3.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

3.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

3.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

4. Adding and removing list values

4.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

4.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

4.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

4.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

4.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

5. Looping through lists

5.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

5.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

5.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

5.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

5.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

6. Changing the order of lists

6.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

6.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS/PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

6.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

6.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

6.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

7. List subsections

7.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

7.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

7.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

7.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

7.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

8. Debugging lists

8.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

8.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

8.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

8.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

8.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

9. Using 2D lists

9.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

9.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

9.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

9.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

9.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

10. Review lessons 1-10

10.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

10.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

10.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

10.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

10.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

11. Introduction to functions

11.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

11.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS/PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

11.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

11.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

11.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

12. Creating and running functions

12.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

12.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

12.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

12.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

12.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

13. Function paramters

13.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

13.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

13.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

13.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

13.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

14. Return values from functions

14.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

14.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS/PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

14.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

14.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

14.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

15. Turtle function arguments

15.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

15.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

15.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

15.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

15.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

16. Scope of variables

16.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

16.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

16.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

16.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

16.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

17. Reasons to use functions

17.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

17.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

17.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

17.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

17.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

18. Changing lists within functions

18.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

18.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

18.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

18.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

18.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

19. Testing functions

19.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

19.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS/PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

19.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

19.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

19.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

20. Review lessons 11-20

20.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

20.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

20.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

20.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

20.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

21. Introduction to dictionaries

21.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

21.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

21.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

21.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

21.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

22. Creating dictionaries

22.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

22.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS/PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

22.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

22.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

22.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

23. Accessing dictionary values

23.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

23.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

23.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

23.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

23.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

24. Looping through dictionaries

24.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

24.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

24.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

24.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

24.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

25. Adding and changing values

25.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

25.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

25.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

25.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

25.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

26. Debugging dictionaries

26.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

26.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

26.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

26.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

26.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

27. Dictionaries and functions

27.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

27.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

27.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

27.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

27.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

28. Testing and commenting code

28.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

28.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

28.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

28.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

28.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

29. Turtle dictionaries

29.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

29.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

29.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

29.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

29.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?

30. Review lessons 21-30

30.1. Introduction

So you've decided to extend your Python knowledge. Great!

This level 2 Python course will teach you about using **collections** such as **lists** and **dictionaries**, as well as how to write **functions**, in order to make your code more **robust** and versatile.

Let's start with a little recap to see what you remember!

1. The `print()` function outputs text to the screen. On line 2, print "Welcome to Level 2 Python!"
2. On line 5, use `input()` to ask the user for their name and store it in a variable called `name`.
3. On line 6, print out "Nice to meet you [...]" where [...] is the name the user types in.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Print statements (lessons 2 and 4)
- Variables and input (lesson 7)

30.2. Revisit math

Python is a great calculator. We have a collection of basic math operators to use:

`+ - * / **`, and it even understands **BEDMAS**/**PEMDAS** or the **order of operations**.

1. Click to remind yourself of how the print statement on line 2 works.
2. Use a `print` statement on line 5 to output the result of **3 * 7**.
3. **Add brackets** `()` to make the print statement on line 8 output **48** instead of 30.
4. Write a print statement that uses **division** on line 11.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Math (lesson 3)

30.3. Revise **conditions** and **if statements**

In level one we learned about **conditional** structures, or ones that have different **branches** of code that can be run, depending on whether or not a condition is `True` or `False`

In the code editor are some print statements with **Boolean** expressions, and a basic **if statement**.

1. Change the print statement on line 2 so it prints `False`.
2. Change the **operator** on line 3 so it prints `True`.
3. Test the **if statement** code by clicking and typing in "happy". Test it again with any other input to see what happens.
4. Add an `elif` branch that will say "Tomorrow is another day..." if the user types "sad".
5. Add an `else` branch with any other message for people who aren't happy or sad.
6. Click to test these new branches.

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- Boolean values (lesson 13)
- If statements (lessons 11-12)

30.4. Revisit Loops

In level 1 we learned about **for loops** and **while loops**. Take a look at the code editor to see the **syntax** for these again, and have a go at modifying the code.

1. Change the loop on lines 2-3 so that it prints the numbers from **1-15**.
2. Change the loop on lines 6-7 so that it prints the **string 3 times**.
3. Try changing the **operator** in the condition on line 11 so that the number **10** is also printed.
4. The password has changed to "monty", modify the loop on lines 16-19 to accept the correct password.
5. **(Optional)** Write a loop that prints **from 4 to 20**, but **counts up in twos.**

If you found these exercises a bit challenging, you may like to revisit the following lessons in the level 1 course:

- For loops (lessons 21-24)
- While loops (lessons 25-27)

30.5. Review Quiz!

Let's review our level 1 basics with a quiz.

Review Quiz Questions:

1. What symbol goes around a string?
2. What will be the result of this print statement?
3. What is the output of the following print statement?
4. What is the last number printed by this loop?
5. What is the output of this if statement if the user types in "robert"?