Girls' Programming Network

Guess Who!

Workbook G

Create a program that picks a Guess Who character at random and gives you hints to help you guess it!

TUTORS ONLY

This project was created by GPN Australia for GPN sites all around Australia!

This workbook and related materials were created by tutors at:

Sydney, Canberra and Perth

Girls' Programming Network

If you see any of the following tutors don't forget to thank them!!

Writers

Jeannette Tran
Fiona Lin
Renee Noble
Alex McCulloch
Annie Liu
Bryony Lanigan
Heather Catchpole
Sam Mangan
Ella Ferris

Testers

Vivian Dang
Courtney Ross
Rachel Alger
Libby Berrie
Caitlin Macleod
Sheree Pudney
Rashmica Gupta
Caitlin Shaw

A massive thanks to our sponsors for supporting us!

ATLASSIAN

Part 1: Welcome to 'Guess Who'

Task 1.1: Welcome to 'Guess Who'

Let's **print** out a welcome message to the players. You can make the computer say anything you want!

```
----------------------
Welcome to Guess Who!
----------------------
```

You can print multiple lines using multiple print statements, like this:

```
print('Mix ingredients.')
print('Bake for 20 minutes.')
```

TUTOR TIPS

You may need to remind them to use the run button or the Ctrl-Enter key to run the program

You may need to remind them to copy their code between lessons, remember to use Ctrl-A, Ctrl-C, Ctrl-V

Task 1.2: Who is playing my game?

Let's find out who's playing!

Use **input** to ask the user for their name. Store their answer in a variable called **player_name** so we can use it in our code!

Task 1.3: Let's play!

Now that we know the player's name, let's **print** out a customised message to them.

For example, if the player typed in Annie, we might see

```
Let us start playing Annie
```

Hint

Remember to use the `player_name` variable that you made in Task 1.2!

TUTOR TIPS

Some students may try and `print` 'Let's start playing', and use single quotes without escaping the apostrophe.

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 2:

- ☐ Print a welcome message to the player
- ☐ Ask for the player's name
- ☐ Print a customised message to the player
- ☐ Try running your code!

TUTOR TIPS

The code should look like this (no bonuses):

```
# <the student's name>

print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us start playing " + player_name)
```

Part 2: Picking a person!

Lists

Task 2.1: Creating a person

Let's create our own character using a list, and store it in a variable called **`character`**.

You can make the person to be anyone you want!

We want the list to store the character's name, eye colour, hair colour, and accessory. We'll store them in this order:
`[<name>, <eye colour>, <hair colour>, <accessory>]`

For example:
`['Annie', 'brown', 'blue', 'glasses']`

Hint

We can create a list of pet names and store it in a variable called `pets` like this:
`pets = ['Fluffy', 'Oscar', 'Audrey', 'Molly']`

TUTOR TIPS

Make sure the student uses square brackets!

Task 2.2: Print out the character

Let's **`print`** out the character that you have created. It should look like this:

```
['Annie', 'brown', 'blue', 'glasses']
```

TUTOR TIPS

Make sure the students print out the person so that they can see that it's a list.

Task 2.3: Splitting up the list!

It's easier to access the character's individual features if we store each one in its own variable. Let's do that now!

Make a variable called **name**. Get the name from the character list you created in Task 2.1 and store it in this variable!

Hint

To get Fluffy out of a list, you add this to your program:

```
pets = ['Fluffy', 'Oscar', 'Audrey', 'Molly']
cat = pets[0]
```

Don't forget that lists start at index 0!

Task 2.4: More features!

Let's also get the other features of our person out and store them in variables.

1. Make a different variable called **eye_colour**. Get the character's eye colour from the list, and store it in here.

2. Make another variable - this time for the hair colour. You can decide what to call it (**hair_colour** could be good). Store the character's hair colour in here.

3. The last feature we need is the character's accessory, so make a variable for this!

TUTOR TIPS

The students may try to do something like:

```
eye_colour = "brown"
```

Instead of:

```
eye_colour = character[1]
```

This code will work fine, but will cause issues when it comes to later parts, when there will be different people with different characteristics.

Task 2.5: Print out each of the features

Use `print` statements to print out the name, eye colour, hair colour and accessory of our character.

It should look like this when you run your code:

```
Name is: Annie
```

```
Eye is: brown
Hair is: blue
Accessory is: glasses
```

Now you've checked your code is working, comment out the print statements you wrote in 2.2 and 2.5. It would be a pretty boring game if we just told the player who the character is!

Hint

Comment out a line a code by putting # at the front like this:

```
# this is a comment
# print("Hello")
# code won't run when there's a # at the start of the line!
```

TUTOR TIPS

Some students may try and do this with just a single print statement. It will probably be easier to split it up!

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 3:

- ☐ Created a person
- ☐ Print out the person
- ☐ Split up the features of the person and store them in variables
- ☐ Print out the features of the person (and then comment it out)
- ☐ Run your code!
- ☐ Comment out printing the features of the person

TUTOR TIPS

The code should look like this (no bonuses):

```
# <the student's name>
```

```
print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us start playing " + player_name)

# start part 2
character = ["Annie", "brown", "blue", "glasses"]
# print(character)

name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]
# print("Name is:", name)
# print("Eye is:", eye_colour)
# print("Hair is:", hair_colour)
# print("Accessory is:", accessory)
```

Part 3: Guess who?

Task 3.1: Guessing someone's name

Use `input` to ask the player to guess the name of your character. Save their answer in a variable - name it something like `guess`.

Hint

Don't forget to comment out the code where we `print` out the character we created and their features!
It would be a pretty boring game if we just told the player who the character is.

TUTOR TIPS

It doesn't explicitly say to print out what the player's guess is. But if the student is struggling to understand `input`, printing it might help!

If Statements

Task 3.2: Check if they have guessed correctly!

Use `if` and `else` statements to tell the player whether or not they have guessed the name correctly.

You should also congratulate them if they have guessed it right:

```
Guess who? Annie
You got it right!
```

You should print out the correct name if they have guessed wrong, like below:

```
Guess who? Mary
Nope, sorry, it was Annie!
```

Hint

You can write an if else statement like this

```
word = "GPN"
if word == "GPN":
  print("GPN is awesome!")
else:
  print("The word isn't GPN :(")
```

TUTOR TIPS

Some students may use a single equal sign to test for equality. Make sure they are using double equals and also check they've used :

If it's not matching, make sure the name the students are typing in as a guess has the same upper/lower case letters as the name in their code (ie Annie (with a capital) doesn't equal annie (all lower case).

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 4:

- ☐ Ask the player to guess who and store it in a variable
- ☐ Congratulate them if they guess it right
- ☐ Tell them what the correct name was if they are wrong
- ☐ Run your code and test different names

TUTOR TIPS

The code should look like this (no bonuses):

```
# <the student's name>

print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us start playing " + player_name)

# start part 2
character = ["Annie", "brown", "blue", "glasses"]
# print(character)

name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]
# print("Name is:", name)
# print("Eye is:", eye_colour)
```

```
# print("Hair is:", hair_colour)
# print("Accessory is:", accessory)

# start part 3
guess = input("Guess who? ")
if (guess == name):
 print("You got it right!")
else:
 print("Nope, sorry, it was " + name + "!")
```

★ BONUS 3.3: ALEX alex or AlEx

Waiting for the next lecture? Try adding this bonus feature!!

We can use `word = word.title()` to change what the player entered to title case. Title case is when the first letter is upper case and all the rest are lower case, like a name!

Update your code so we're always using the title case version of what your player entered!

TUTOR TIPS

The code should look like this (with bonuses):

```
# <the student's name>

print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us start playing " + player_name)

# start part 2
character = ["Annie", "brown", "blue", "glasses"]
# print(character)


name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]
# print("Name is:", name)
# print("Eye is:", eye_colour)
```

```
# print("Hair is:", hair_colour)
# print("Accessory is:", accessory)

# start part 3 + bonus
guess = input("Guess who? ")
guess = guess.title()
if (guess == name):
 print("You got it right!")
else:
 print("Nope, sorry, it was " + name + "!")
```

Part 4: Let's get more information!

We are just guessing blindly at the moment, which isn't very fun! Let's let the player get more information about the person before they have to guess who.

Task 4.1: Asking about eyes

Before your code that asks the player to guess who, use `input` to ask the player to guess what the person's eye colour is! Store their answer in a variable - call it something like: **`eye_guess`**

Your question should look something like this, if the user guesses brown eyes.

```
Guess their eye colour? brown
```

Task 4.2: Check the eyes

After you have asked them to guess the eye colour, use an `if` and `else` statement to check to see if the eye colour that the player guessed is the correct one.

If they are right, tell them `"Yes"`, otherwise tell them `"No"`.

For example, if your character's eye colour is brown and the user guesses right:

```
Guess their eye colour? brown
Yes
```

Task 4.3: What's their hair colour?

Now do the same thing that you did for 4.1 and 4.2, but this time ask to guess the person's hair colour and check if they got it right!

Put this after you have checked if they guessed the eye colour correctly, but before your code that asks the player to guess who.

Task 4.4: What's their accessory?

Do the same thing again that you did for 4.1 and 4.2, but for the person's accessory!

Put this after you checked if they guessed hair colour correctly, but before your code that asks the player to guess who.

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 5:

- ☐ Ask the user for their eye colour guess
- ☐ Print out whether or not their eye guess was right
- ☐ Ask the user for their hair colour guess
- ☐ Print out whether or not their hair guess was right
- ☐ Ask the user for their accessory guess
- ☐ Print out whether or not their accessory guess was right
- ☐ Run your code!

TUTOR TIPS

The code should look like this (no bonuses):

```
# <the student's name>
print("Welcome to Guess Who!")

player_name = input("What is your name? ")
print("Let us start playing " + player_name)


# start part 2

character = ["Annie", "brown", "blue", "glasses"]

# print(character)
```

```
name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]
# print("Name is:", name)
# print("Eye is:", eye_colour)
# print("Hair is:", hair_colour)
# print("Accessory is:", accessory)

# start part 4
eye_guess = input("Guess their eye colour? ")
if (eye_guess == eye_colour):
 print("Yes")
else:
 print("No")

hair_guess = input("Guess their hair colour? ")
if (hair_guess == hair_colour):
 print("Yes")
else:
 print("No")

accessory_guess = input("Guess their accessory? ")
if (accessory_guess == accessory):
 print("Yes")
else:
 print("No")

# start part 3
guess = input("Guess who? ")
if (guess == name):
 print("You got it right!")
else:
 print("Nope, sorry, it was " + name + "!")
```

★ BONUS 4.4: BLUE Blue bLuE

Waiting for the next lecture? Try adding this bonus feature!!

We can use `word = word.lower()` to change what the player entered to lowercase. Update your code so we're always using the lowercase version of what your player entered for their guesses (except the name!)

TUTOR TIPS

Make sure the students reassign the variable when changing case.
`word = word.lower()`**, not just** `word.lower()`

★ BONUS 4.5: Not so fast!

Waiting for the next lecture? Try adding this bonus feature!!

This would look cooler if the computer paused before it said each line!

1) At the top of your file write `import time`
   This will let us use what we need to use to make our program sleep for a few seconds.

2) Before we tell the user whether or not they guessed correctly, add a line that says `time.sleep(1)`
   This will make our program 'sleep' for a second! You can adjust it to any time you want.

TUTOR TIPS

Don't forget the `import`**!**

`sleep` **counts in seconds, but can do decimals (e.g. 0.5) for shorter intervals.**

TUTOR TIPS

The code should look like this (with bonuses):

```
# <the student's name>

import time
print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us start playing " + player_name)

# start part 2
character = ["Annie", "brown", "blue", "glasses"]
# print(character)
```

```
name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]
# print("Name is:", name)
# print("Eye is:", eye_colour)
# print("Hair is:", hair_colour)
# print("Accessory is:", accessory)

# start part 4
eye_guess = input("Guess their eye colour? ")
eye_guess = eye_guess.lower()
time.sleep(1)
if (eye_guess == eye_colour):
 print("Yes")
else:
 print("No")

hair_guess = input("Guess their hair colour? ")
hair_guess = hair_guess.lower()
time.sleep(1)
if (hair_guess == hair_colour):
 print("Yes")
else:
 print("No")

accessory_guess = input("Guess their accessory? ")
accessory_guess = accessory_guess.lower()
time.sleep(1)
if (accessory_guess == accessory):
 print("Yes")
else:
 print("No")

# start part 3
guess = input("Guess who? ")
guess = guess.title()
if (guess == name):
 print("You got it right!")
```

```
else:
 print("Nope, sorry, it was " + name + "!")
```

Part 5: Choose a random person.

Task 5.1: One character is not enough

Comment out the line where you created a **`character`** in Task 2.1. It should look something like this:

```
#character = ['Annie', 'brown', 'blue', 'glasses']
```

We're about to get a whole bunch of characters and we'll choose a character randomly from that. You can add your own character to the group later.

TUTOR TIPS

This will cause errors. It is meant to. Students keep working through the chapter to fix them!

Task 5.2: Copy the list of people

Copy the list below and then paste it at the top of your file so that it looks like this:

```
people = [["Aleisha", "brown", "black", "hat"],
          ["Brittany", "blue", "red", "glasses"],
          ["Charlie", "green", "brown", "glasses"],
          ["Dave", "blue", "red", "glasses"],
          ["Eve", "green", "brown", "glasses"],
          ["Frankie", "hazel", "black", "hat"],
          ["George", "brown", "black", "glasses"],
          ["Hannah", "brown", "black", "glasses"],
          ["Isla", "brown", "brown", "none"],
          ["Jackie", "hazel", "blonde", "hat"],
          ["Kevin", "brown", "black", "hat"],
          ["Luka", "blue", "brown", "none"]]
```

TUTOR TIPS

Get the students to note that this is a list of lists! A list of characters and each character has a list with their name, eye colour, hair colour and accessory.

You'll need to mention this often as they get different people from the list, and access the different attributes.

Random

Task 5.3: Import Random Library

To get access to cool random things we need to import random!

At the top of your file add this line:

```
import random
```

Task 5.4: Choose a random person

Let's make the computer pick a random person out of the list that we have to guess!

Use **`random.choice`** to pick from the list of people. Store it in a variable called **`character`**.

Hint

If I wanted to choose a random food for dinner I could use code like this:

```
food_list = ["pizza", "curry", "nutella", "omelette"]
dinner = random.choice(food_list)
```

TUTOR TIPS

Choosing randomly from a list of lists will give us a single list.

Task 5.5: Print out the character

Print out the **`character`** that the computer has chosen.

Try running your code a couple of times! You should get a random person each time.

```
['Hannah', 'brown', 'black', 'glasses']
```

After you have run your code a couple of times, comment out printing the character the computer has chosen - you don't want to tell the player the answer!

TUTOR TIPS

Make sure the students print out the person so that they can see that it's a list for later steps.

Make sure the students comment out the print after they can see it is working.

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 6:

- ☐ Comment out the character you made in Task 2.1
- ☐ Copied and pasted the list
- ☐ Print the list
- ☐ Chosen a random person from the list
- ☐ Print out the randomly chosen person
- ☐ Run your code!

TUTOR TIPS

The code should look like this (no bonuses):

```
# <the student's name>
import random

people = [["Aleisha", "brown", "black", "hat"],
          ["Brittany", "blue", "red", "glasses"],
          ["Charlie", "green", "brown", "glasses"],
          ["Dave", "blue", "red", "glasses"],
          ["Eve", "green", "brown", "glasses"],
          ["Frankie", "hazel", "black", "hat"],
          ["George", "brown", "black", "glasses"],
          ["Hannah", "brown", "black", "glasses"],
          ["Isla", "brown", "brown", "none"],
          ["Jackie", "hazel", "blonde", "hat"],
          ["Kevin", "brown", "black", "hat"],
          ["Luka", "blue", "brown", "none"]]

print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us start playing " + player_name)

# start part 2
```

```
# character = ["Annie", "brown", "blue", "glasses"]
# part 5
character = random.choice(people)
print(character)
name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]
# print("Name is:", name)
# print("Eye is:", eye_colour)
# print("Hair is:", hair_colour)
# print("Accessory is:", accessory)

# start part 4
eye_guess = input("Guess their eye colour? ")
eye_guess = eye_guess.lower()
time.sleep(1)
if (eye_guess == eye_colour):
 print("Yes")
else:
 print("No")

hair_guess = input("Guess their hair colour? ")
hair_guess = hair_guess.lower()
time.sleep(1)
if (hair_guess == hair_colour):
 print("Yes")
else:
 print("No")

accessory_guess = input("Guess their accessory? ")
accessory_guess = accessory_guess.lower()
time.sleep(1)
if (accessory_guess == accessory):
 print("Yes")
else:
 print("No")

# start part 3
```

```
guess = input("Guess who? ")
guess = guess.title()
if (guess == name):
 print("You got it right!")
else:
 print("Nope, sorry, it was " + name + "!")
```

★ BONUS 5.6: Get creative!

You can add your original character to the list, and more!

We've left room for you to draw your own people in the character sheet. Once you've given them a name, eye colour, hair colour and accessory add them into the list of people at the top of your code!

Feel free to add yourself, your friends or one of the wonderful tutors at GPN!

Part 6: Again, Again, Again!

We want to play 'Guess Who' until we guess the correct person! Let's add a loop to guess on repeat!

While Loops

Task 6.1: Loop time!

Create a while loop that runs forever after you've selected the character and separated the features, but before you get them to guess the eye colour. This lets us ask as many questions as we want!

Use this line to make the game play on repeat

```
while True:
```

TUTOR TIPS

Make sure you the student picks a random character and assigns the features to variables BEFORE the loop. Otherwise, they will spend a lot of time guessing and wondering who the character is.

Hint

We want to repeat asking lots of questions and checking if the info is correct. This is why it is important to put this before your questions and if statements about hair/eyes/accessories/name.

Task 6.2: Indenting your code

We want to guess the hair colour, eye colour, accessory and name every time until we guess the name correctly! Things we want to do every time need to be indented inside the loop.

Indent all you code after while True: so it is indented under the while loop.

TUTOR TIPS

Make sure you check that all the code that needs to be indented is.

The student is still printing out the randomly chosen character that the player is trying to guess. Get them to look at this when testing their code so they can see that it's working properly.

Hint

Indented lines have a tab (the big empty space) at the start like this, they look this:

```
while True:
    # THIS IS INDENTED
```

Hint

Did you notice your game doesn't quite work yet! It keeps making you guess when you get it right and tells you the answer when you get it wrong. Don't worry - we'll fix that in the next few steps!

Task 6.3: Stopping

We want our program to stop when we guess the right name!

After we congratulate the user on guessing the right name, add a `break` to stop the loop. Make sure the break is inside the if statement.

Task 6.4: Update the wrong answer!

We don't want our program to tell us the correct name when we guess it wrong - we want to keep guessing until we can get it right!

- Update your `else` statement so that the user is no longer told who the correct person's name when they get it wrong! Change it to say something like "Nope, sorry, that's not right. Try again."

Your game is still printing the character that the computer randomly chose at the start of the game!

- Comment out that print as we don't want to give the player the answer - we want them to Guess Who!

Now you've done all the hard work, you can play your game!

- Use your Guess Who Character Sheet and some paper squares to help you keep track of who you can eliminate with each answer the computer gives to your questions. How many questions will you need to correctly Guess Who the computer chose?

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 7:

- [ ] Create a while loop that lets your game keep going!
- [ ] Your game code is inside the while loop
- [ ] The game only ends when you guess the right person!

TUTOR TIPS

The code should look like this (no bonuses):

```
# <the student's name>
import random
people = [["Aleisha", "brown", "black", "hat"],
          ["Brittany", "blue", "red", "glasses"],
          ["Charlie", "green", "brown", "glasses"],
          ["Dave", "blue", "red", "glasses"],
          ["Eve", "green", "brown", "glasses"],
          ["Frankie", "hazel", "black", "hat"],
          ["George", "brown", "black", "glasses"],
          ["Hannah", "brown", "black", "glasses"],
          ["Isla", "brown", "brown", "none"],
          ["Jackie", "hazel", "blonde", "hat"],
          ["Kevin", "brown", "black", "hat"],
          ["Luka", "blue", "brown", "none"]]

print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us start playing " + player_name)

# character = ["Annie", "brown", "blue", "glasses"]
character = random.choice(people)
# print(character)
name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]

#part 6
while True:
```

```
eye_guess = input("Guess their eye colour? ")
eye_guess = eye_guess.lower()
if (eye_guess == eye_colour):
  print("Yes")
else:
  print("No")

hair_guess = input("Guess their hair colour? ")
hair_guess = hair_guess.lower()
if (hair_guess == hair_colour):
  print("Yes")
else:
  print("No")

accessory_guess = input("Guess their accessory? ")
accessory_guess = accessory_guess.lower()
if (accessory_guess == accessory):
  print("Yes")
else:
  print("No")

guess = input("Guess who? ")
guess = guess.title()
if (guess == name):
  print("You got it right!")
  break #part6
else:
  print("Nope, sorry, that's not right. Try again.") #part 6
```

7. Extension: Which questions?

So far, we've had to ask each question every time we want to guess - even if you only needed to find out what their accessory is. Now, we want to let the human player decide which question they want to ask.

Task 7.1: Eye colour, hair colour, accessory, or name?

At the start of your code inside the `while` loop, add a question that asks them what kind of question they would like to ask (eye colour, hair colour, accessory, or name?).

Task 7.2: Question time!

You already have 4 kinds of questions and if statement checks. Put each of these 4 chunks inside an if statement that means that it will only occur if this was the type of question the user wanted to ask.

Hint

Remember that `input` returns a `string`. Make sure that your type of `input` (string, int, etc.) matches the `if` statement!

Don't forget to indent the code under your new `if` statements!

TUTOR TIPS

The finished code for all extensions (no bonuses) is right at the end of this booklet.

8. Extension: How many questions?

Now, let's track how many (or how few) questions it takes you each game to guess correctly!

Task 8.1: Counter!

Before your loop, create a `variable`, this will be your guess counter. Start by setting it to `0`.

TUTOR TIPS

Some students will just call this variable '`variable`'. It will be less confusing later if they gave this a meaningful name, like `counter`!

Task 8.2: Add 1!

Every time the user makes a guess (a name guess or any other feature guess), add one to this counter.

Hint

You'll need to add to the counter at the beginning of your `while` loop!

We can add 1 to a variable called counter like this:

```
counter = counter + 1
```

TUTOR TIPS

This counter has to be created before the `while` loop, otherwise they'll reset it every time.

Task 8.3: How many questions?

At the end of the game, `print` out how many questions the user has asked.

Hint

Make sure you tell the user what you're printing for them. If we were printing how many times a user entered number eight, we'd do it like this:

```
print("You entered number eight", counter, " times")
```

TUTOR TIPS

The finished code for all extensions (no bonuses) is right at the end of this booklet.

9. Extension: I give up!

What if you're sick of guessing, and just want to find out who it is? Let's now add to our code so that we can decide to give up and finish the game.

Task 9.1: I give up!

Update the `input` statement where the user says what they want to guess. Add "give up" as another option.

Add another `if` statement to check if the user entered "give up".

If they did, let's reveal the answer and end the game.

Hint

Remember, you can use `break` to get out of the loop.

TUTOR TIPS

The finished code for all extensions (no bonuses) is right at the end of this booklet.

★ Challenge 9.2: Cases and Spaces

What we've written so far will only give the user the answer and quit the game if they input `"give up"` .

What if we want our game to be more robust, and understand that someone typing `"giveup"` or `"Give Up"` wants the same result?

How could you modify your code so they can quit the game if they input `"giveup"` or `"Give Up"`.

Hint

We can use `text = text.lower()` to change what the player entered to all lower case characters.

We can use `text = text.replace(" ","")` to remove any spaces from what the player entered

TUTOR TIPS

If they use `text.lower()` and `text.replace` to reformat `question` make sure they change their if statement to check for "giveup" instead of "give up".

`text.lower()` **can be used to solve problem of user inputting things like NAME or Hair (ie with capitals).**

★ Challenge 9.3: Even more options!

What if we want the user to be able to input even more options, like 'I QUIT', or 'WHY CANT I GUESS THIS', and have the computer know that this means the person does not want to play any more and to end the game and tell them the answer?

TUTOR TIPS

The code should look like this (no bonuses):

```
# <the student's name>
import random
people = [["Aleisha", "brown", "black", "hat"],
          ["Brittany", "blue", "red", "glasses"],
          ["Charlie", "green", "brown", "glasses"],
          ["Dave", "blue", "red", "glasses"],
          ["Eve", "green", "brown", "glasses"],
          ["Frankie", "hazel", "black", "hat"],
          ["George", "brown", "black", "glasses"],
          ["Hannah", "brown", "black", "glasses"],
          ["Isla", "brown", "brown", "none"],
          ["Jackie", "hazel", "blonde", "hat"],
          ["Kevin", "brown", "black", "hat"],
          ["Luka", "blue", "brown", "none"]]

print("Welcome to Guess Who!")
player_name = input("What is your name? ")
print("Let us play " + player_name)

# character = ["Annie", "brown", "blue", "glasses"]
character = random.choice(people)
```

```
# print(character)
name = character[0]
eye_colour = character[1]
hair_colour = character[2]
accessory = character[3]

counter = 0

while True:
  question = input("Which question? eye, hair, accessory, name or give
up? ")
  #9.2
  question = question.lower()
  question = question.replace(" ","")

  counter = counter + 1

  if (question == "eye"):
    eye_guess = input("Guess their eye colour? ")
    eye_guess = eye_guess.lower()
    if (eye_guess == eye_colour):
      print("Yes")
    else:
      print("No")

  if (question == "hair"):
    hair_guess = input("Guess their hair colour? ")
    hair_guess = hair_guess.lower()
    if (hair_guess == hair_colour):
      print("Yes")
    else:
      print("No")

  if (question == "accessory"):
    accessory_guess = input("Guess their accessory? ")
    accessory_guess = accessory_guess.lower()
    if (accessory_guess == accessory):
      print("Yes")
```

```
    else:
      print("No")

if (question == "name"):
    guess = input("Guess who? ")
    guess = guess.title()
    if (guess == name):
      print("You got it right!")
      print("You got it in", counter, "guesses!")
      break
    else:
      print("Nope, sorry, that's not right. Try again.")

  if (question == "giveup"): #9.2 change from "give up" to "giveup"
     print("It was", name)
     break
```