

# Welcome to the Labs

Scissors Paper Rock!

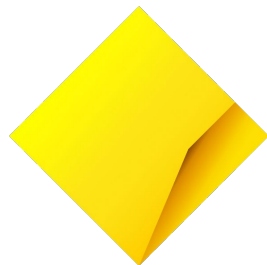


# Thank you to our Sponsors!

Platinum Sponsor:



Gold Sponsor:



**Commonwealth  
Bank**



# Who are the tutors?



Who are you?



# Log on

## Log on and jump on the GPN website

[girlsprogramming.network/workshop](https://girlsprogramming.network/workshop)

Click Content for your room. You can see:

- These **slides** (to take a look back or go on ahead).
- A link to our **workbook** in EdStem.
- A **cheatsheet** of python shortcuts!

There's also links to places where you can do more programming!



Tell us you're here!

Click on the  
**Start of Day Survey**  
and fill it in now!

# Today's project!

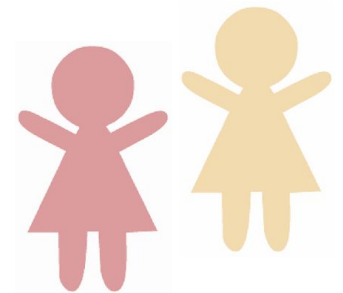
## Scissors Paper Rock



# Ultimate Scissors Paper Rock

1. Start with a partner
2. play scissors paper rock!

**Who will be the champion?**

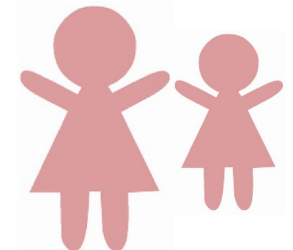
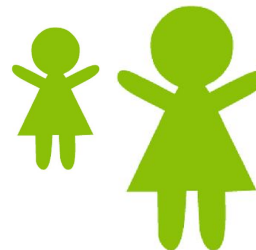
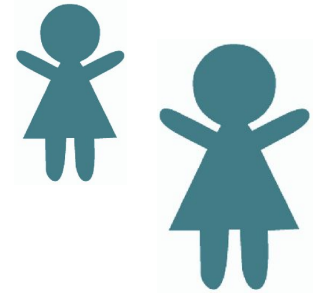




# Ultimate Scissors Paper Rock

1. Start with a partner
2. play scissors paper rock!
3. If you win they become your cheer squad!  
And their squad becomes your squad!
4. Find a new partner!
5. Keep playing until there is only one person left!

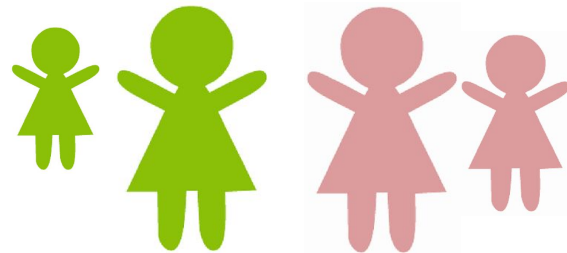
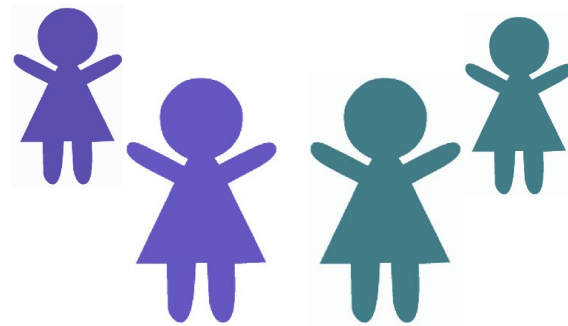
**Who will be the champion?**



# Ultimate Scissors Paper Rock

1. Start with a partner
2. play scissors paper rock!
3. If you win they become your cheer squad!  
And their squad becomes your squad!
4. Find a new partner!
5. Keep playing until there is only one person left!

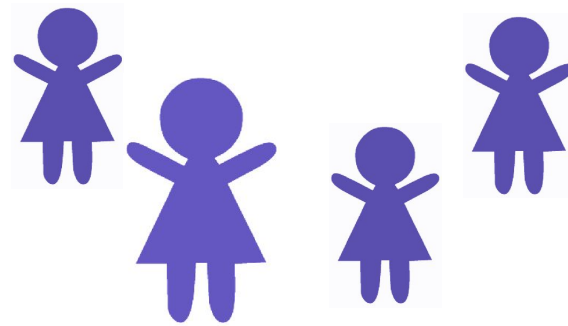
**Who will be the champion?**



# Ultimate Scissors Paper Rock

1. Start with a partner
2. play scissors paper rock!
3. If you win they become your cheer squad!  
And their squad becomes your squad!
4. Find a new partner!
5. Keep playing until there is only one person left!

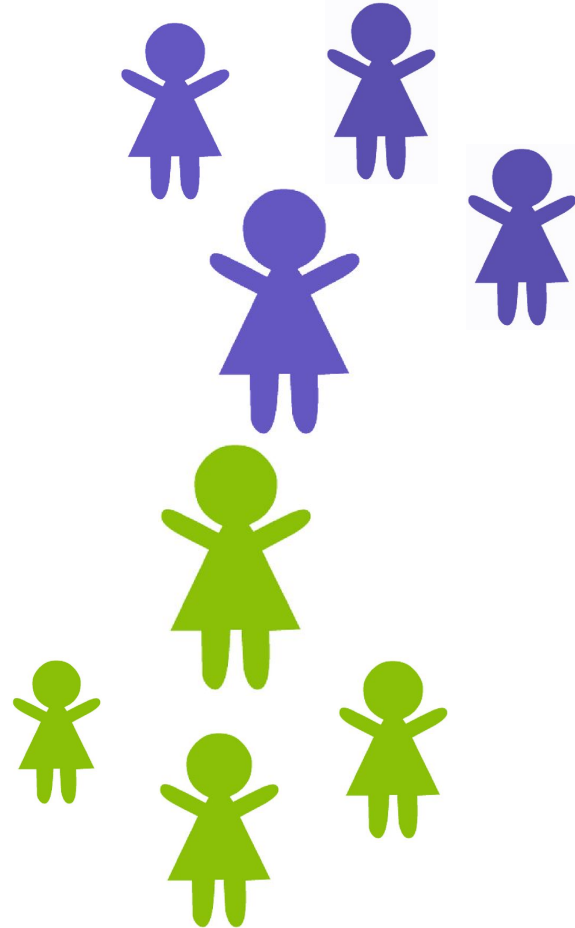
**Who will be the champion?**



# Ultimate Scissors Paper Rock

1. Start with a partner
2. play scissors paper rock!
3. If you win they become your cheer squad!  
And their squad becomes your squad!
4. Find a new partner!
5. Keep playing until there is only one person left!

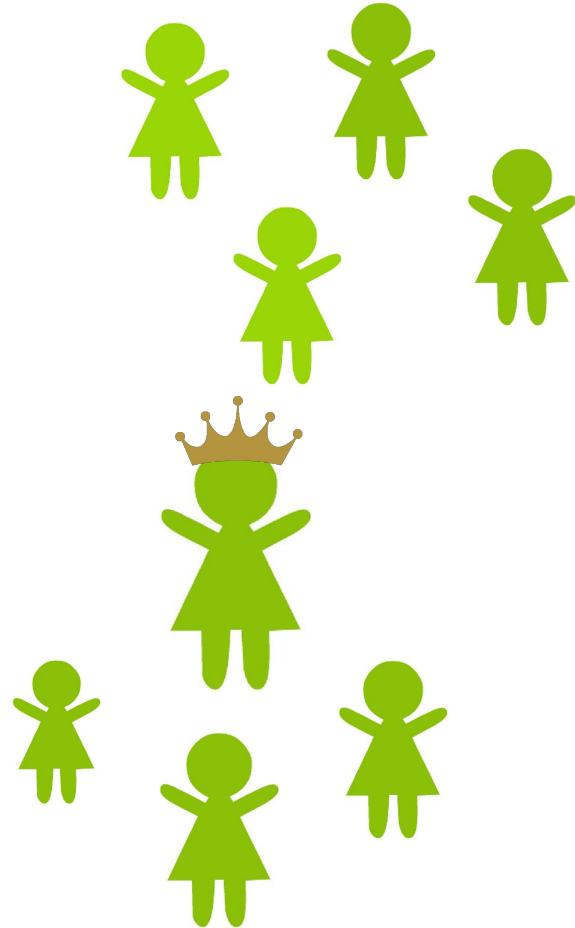
**Who will be the champion?**



# Ultimate Scissors Paper Rock

1. Start with a partner
2. play scissors paper rock!
3. If you win they become your cheer squad!  
And their squad becomes your squad!
4. Find a new partner!
5. Keep playing until there is only one person left!

**Who will be the champion?**



# Scissors Paper Rock

How did you go? Did you win?

Some of the things that we need to do to play scissors paper rock include:

- We have to select a move (out of scissors, paper and rock)
- Our opponent has to select a move
- We need to know what combinations of moves result in win, lose or tie
- We need to compare our moves to see who won
- We have to congratulate the winner!

We'll be programming these actions today! Our opponent is going to be the computer.



# Signing up to Edstem

We are shifting all our courses to a new website called “Edstem” so here’s an overview of how to sign up and how to use it.

First let’s go through how to create an account.

1. Follow the link on the website [www.girlsprogramming.network/workshop](http://www.girlsprogramming.network/workshop)
2. Type in your name and your personal email address
3. Click Create Account
4. Go to your email to verify your account
5. Create a password
6. It should then take you to the courses home page.

*If you don’t have access to your email account, ask a tutor for a GPN edStem login*

# Getting to the lessons

1. Once you are in the course, you'll be taken to a discussion page.
2. Click the button for the lessons page (top right - looks like a book)

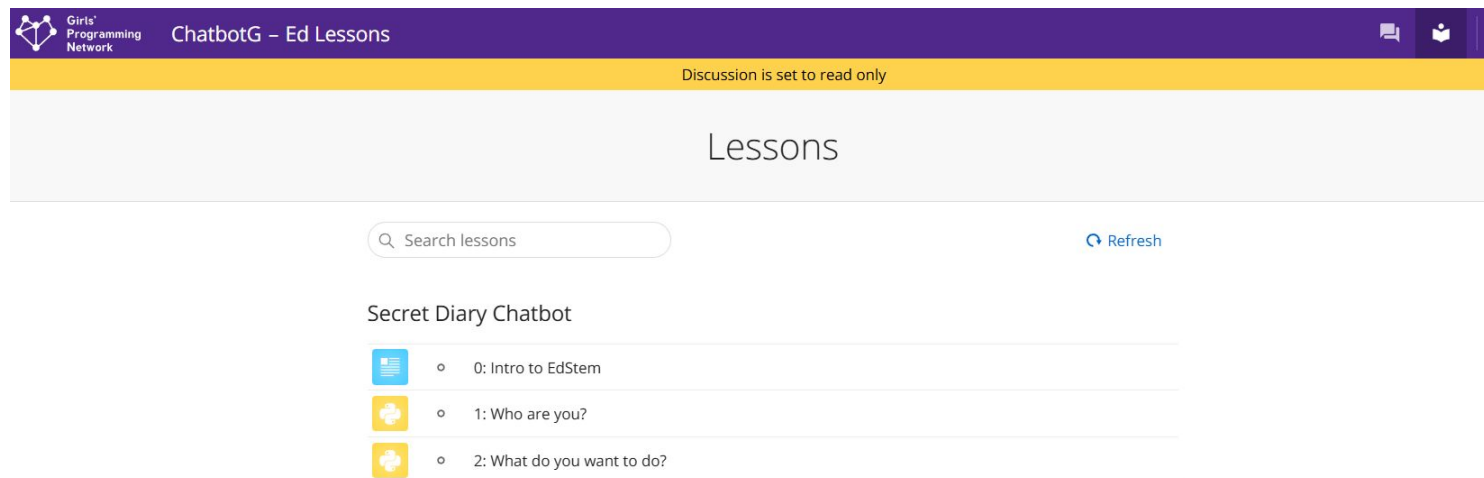




# The Anatomy of the workbook

## The main page:

- Heading at the top that tells you the project (SPR)
- List of “Chapters” - they have icons that looks like this:
- To complete your project, work through the chapters one at a time



# Inside a Chapter

Inside a chapter there are two main types of pages:

1. **Lessons** - where you will do your coding.

They have this icon:



2. **Checkpoints**



Checkpoint

Each chapter has a checkpoint to complete to move to the next chapter. Make sure you scroll down to see all the questions in a checkpoint.

≡ 2: What do you want to do?

<> 2.1 Welcome to the Secret Diary

<> 2.2 What do you want to write?

<> 2.3 Do you want to read?

<> 2.4 I don't understand!

 Checkpoint

# How to do the work

In each lesson there is:

- A section on left with instructions
- A section on right for your code

You will need to **copy your code from the last lesson**, then follow the instructions to change your code

There are also  
Hints and  
Code Blocks to  
help you



**Description**

### Example Lesson Page

This is an example lesson page. This is where you're instructions and hints will go in each of your lessons.

To test out how to write code in this new online workbook...

1. First, print `"Hello EdStem!"`
2. Then run your code in the terminal. Remember you will need to enter the code `python chatbot.py`

**Hint:** You can print using the code;

**Run** PYTHON

```
1 print("Hello World")
```

**Files** chatbot.py

```
1 # Put your testing code here!!!
2 print("Hello EdStem!")
```

/home/chatbot.py Spaces: 4 (Auto) All changes saved

**Terminal**



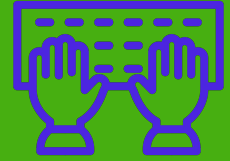
# Some shortcuts...

There are a couple things you can do to make copying your code from one page to another easier.

- 1) **Ctrl + A**      Pressing these keys together will select all the text on a page
- 2) **Ctrl + C**      Pressing these keys together will copy anything that's selected
- 3) **Ctrl + V**      Pressing these keys together will paste anything you've copied



# Need help with EdStem?



There is a section at the top of your workbook that explains how to use EdStem if you get stuck and need a reminder!

**It's called 0: Intro to EdStem**

## Secret Diary Chatbot



◦ 0: Intro to EdStem



◦ 1: Who are you?

**Go to Part 0 and have a look!**



# Classes



# What is an object?

## What do you think an object is?



# What is an object?

## What do you think an object is?





# What is an object?

## What do you think an object is?



# What is an object?

What do you think an object is?



# What is an object?

What do you think an object is?



# What is an object?

What do you think an object is?



# What is an object in code?

An object is something that we know information about and that can sometimes do things



# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!

What information might we know about a cat?



# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What information might we know about a cat?

**Name**



# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What information might we know about a cat?

**Name**

**Age**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What information might we know about a cat?

**Name**

**Age**

**Colour**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What information might we know about a cat?

**Name**

**Owner**

**Age**

**Colour**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What information might we know about a cat?

**Name**  
**Age**  
**Colour**

**Owner**  
**Weight**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What information might we know about a cat?

**Name**

**Owner**

**Age**

**Microchip #**

**Weight**

**Colour**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!

What things might a cat do?



# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!

What things might a cat do?



**Meow**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What things might a cat do?

**Meow**  
**Eat**



# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What things might a cat do?

**Meow**  
**Eat**  
**Scratch**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What things might a cat do?

**Meow**  
**Eat**  
**Scratch**

**Sleep**

# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What things might a cat do?

**Meow**  
**Eat**  
**Scratch**

**Sleep**  
**Purr**



# What is an object in code?

An object is something that we know information about and that can sometimes do things

Like a cat!



What things might a cat do?

**Meow**

**Eat**

**Scratch**

**Jump**

**Sleep**

**Purr**



# What does that look like in Python?

Let's have a look at how we might make a Cat object in Python code!



# What does that look like in Python?

Let's have a look at how we might make a Cat object in Python code!

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour
```

Here we tell python that we are making a new type (or class) of object called Cat

# What does that look like in Python?

Let's have a look at how we might make a Cat object in Python code!

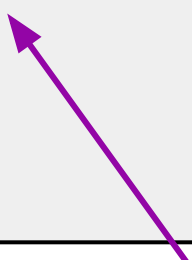
`__init__` is how we tell Python how to make a new Cat

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour
```

# What does that look like in Python?

Let's have a look at how we might make a Cat object in Python code!

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour
```



Here we tell Python what information we need to know about the Cat

Note: self is special and we always need it






# What does that look like in Python?

Let's have a look at how we might make a Cat object in Python code!

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour
```



Here we save the  
information we got so  
we can use it again

# What does that look like in Python?

## How do we make a new Cat?

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
emmy = Cat("Emmy", 3, "Dark brown")
```



# What does that look like in Python?

What does this print out?

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
emmy = Cat("Emmy", 3, "Dark brown")  
print(emmy.name)  
print(emmy.age)  
print(emmy.colour)
```



# What does that look like in Python?

What does this print out?

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
emmy = Cat("Emmy", 3, "Dark brown")  
print(emmy.name)  
print(emmy.age)  
print(emmy.colour)
```

Emmy

3

Dark Brown



# What about doing things?

We said an object was something with information that could sometimes do things. Our Cat object doesn't do anything right now - let's add a way for it to meow!



# What about doing things?

We said an object was something with information that could sometimes do things. Our Cat object doesn't do anything right now - let's add a way for it to meow!

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
    def meow(self):  
        print("Meow")
```

# What about doing things?

What does this code do?

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
    def meow(self):  
        print("Meow")  
  
emmy = Cat("Emmy", 3, "Dark brown")  
emmy.meow()
```



# What about doing things?

What does this code do?

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
    def meow(self):  
        print("Meow")  
  
emmy = Cat("Emmy", 3, "Dark brown")  
emmy.meow()
```

Meow





# What else can it do?

Let's have our cat have a Birthday that makes it get older by 1 year!



# What else can it do?

Let's have our cat have a Birthday that makes it get older by 1 year!

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
    def meow(self):  
        print("Meow")  
  
    def birthday(self):  
        self.age = self.age + 1
```



# What else can it do?

What does this code do?

```
class Cat():  
    def __init__(self, name, age, colour):  
        self.name = name  
        self.age = age  
        self.colour = colour  
  
    def meow(self):  
        print("Meow")  
  
    def birthday(self):  
        self.age = self.age + 1  
  
emmy = Cat("Emmy", 3, "Dark brown")  
emmy.birthday()  
print(emmy.age)
```



# What else can it do?

What does this code do?

```
class Cat():
    def __init__(self, name, age, colour):
        self.name = name
        self.age = age
        self.colour = colour

    def meow(self):
        print("Meow")

    def birthday(self):
        self.age = self.age + 1

emmy = Cat("Emmy", 3, "Dark brown")
emmy.birthday()
print(emmy.age)
```



# I have more than 1 cat!

Emmy has a little sister, Saphira! Let's add her to our code too!

```
cat1 = Cat("Emmy", 3, "Dark brown")  
cat2 = Cat("Saphira", 1, "Grey")
```



# Cat Crime!

There has been a cat crime!

One of the cats has gotten on the kitchen counter and eaten some of my lunch!

They both look innocent but they left a hair behind at the scene of the crime! Let's write some code to work out who did it



# Cat Crime

Who did it??

```
cat1 = Cat("Emmy", 3, "Dark brown")
cat2 = Cat("Saphira", 1, "Grey")

hair_colour = "Grey"

if hair_colour == cat1.colour:
    print("That hair belongs to", cat1.name)
elif hair_colour == cat2.colour:
    print("That hair belongs to", cat2.name)
```



# Cat Crime

Who did it??

```
cat1 = Cat("Emmy", 3, "Dark brown")
cat2 = Cat("Saphira", 1, "Grey")

hair_colour = "Grey"

if hair_colour == cat1.colour:
    print("That hair belongs to", cat1.name)
elif hair_colour == cat2.colour:
    print("That hair belongs to", cat2.name)
```

That hair belongs to Saphira





# Project time!

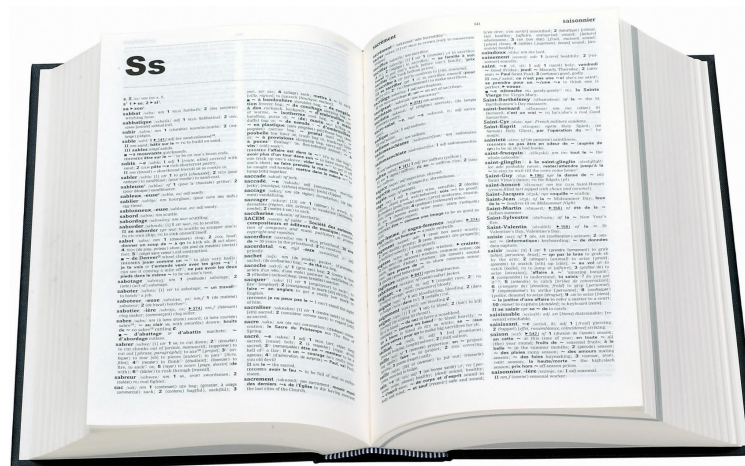
You now know all about **classes**!

**Let's put what we learnt into our project**  
**Try to do Parts 0-2**

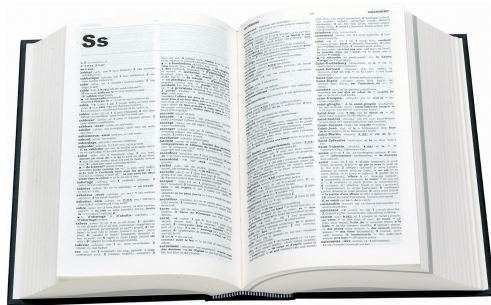
The tutors will be around to help!



# Dictionaries



# Dictionaries!

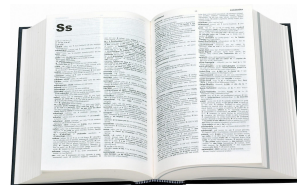


***You know dictionaries!***

**They're great at looking up thing  
by a word, not a position in a list!**

Look up

***Hello***



Get back

***A greeting (salutation) said  
when meeting someone or  
acknowledging someone's  
arrival or presence.***



# Looking it up!

**There are lots of times we want to look something up!**



**Competition registration**

Team Name → List of team members



**Phone Book**

Name → Phone number



**Vending Machine**

Treat Name → Price

Looking it up!



## Phone Book

Name → Phone number

↑  
Key

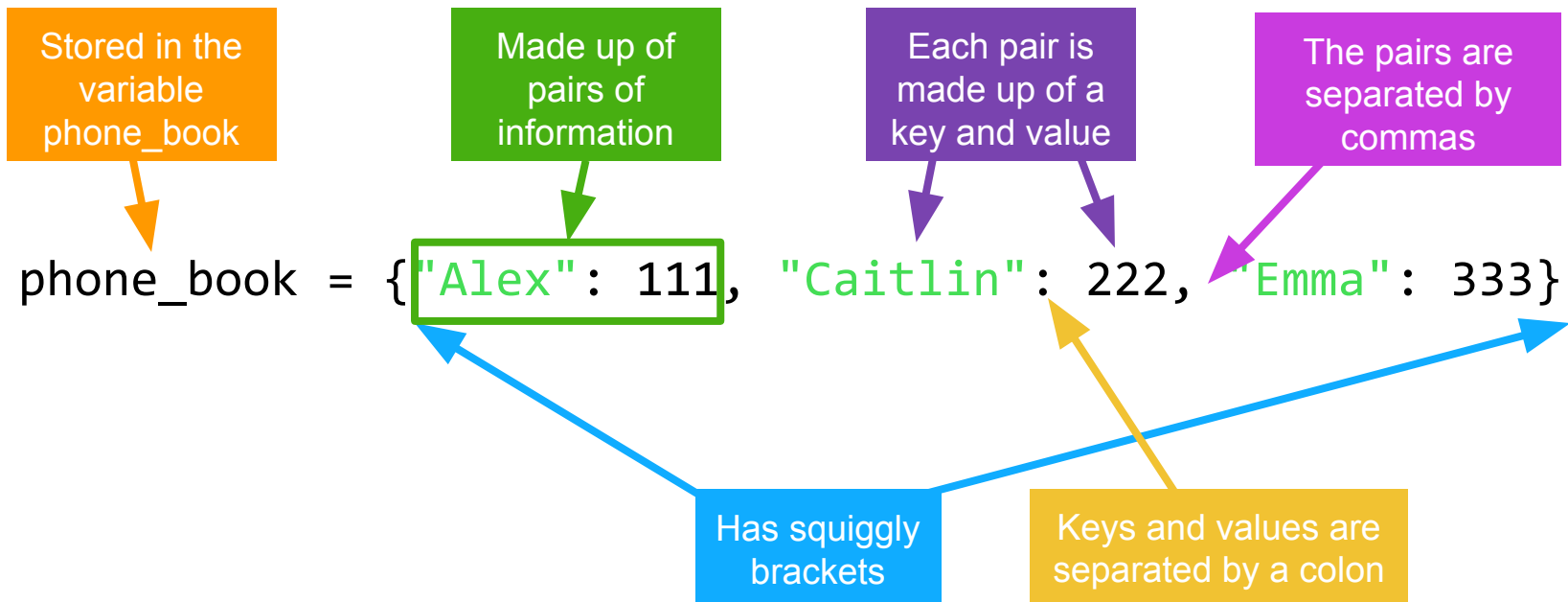
↑  
Value

**We can use a dictionary for anything with a  
key → value pattern!**



# Dictionaries anatomy!

**This is a python dictionary!**



**This dictionary has Alex, Caitlin and Emma's phone numbers**



# Playing with dictionaries!



Let's try using the phone book

1. Copy in the dictionary! Add your own made up phone number!

```
phone_book = {"Alex": 111, "Caitlin": 222, "Emma": 333}
```

2. Try this: `phone_book["Alex"]`

3. How would you look up Emma's phone number?

4. Look up the name of someone who is not in the phone book? What happens?



# Save it for later!



Sometimes we don't need the info right now.

Let's store it in a variable and use it later!

1. Look up Alex's phone number and store it in a variable

```
alexs_number = phone_book["Alex"]
```

2. Print out a message using alexs\_number

```
print("Alexs number is: ", alexs_number)
```

3. Repeat task 1 and 2 for another person in the phone book!





# Tuples!

## Some data sticks together!

Tuples are like lists that you can't edit or add too!

**It's a:**

- **list of items**
- **in round brackets**
- **separated by commas**

**Tuples are a way of grouping data!**

("January", "1st")

("December", "25th")

("April", "25th")



# Tuples in dictionaries!



## We can use tuples as the key to a dictionary

1. Copy in the dictionary! Add your own made up phone number!

```
phone_book = {("January", "1st"): "New Years",  
              ("December", "25th"): "Christmas Day",  
              ("April", "25th"): "ANZAC Day"}
```

2. Try this: `phone_book[("January", "1st")]`
3. How would you look up what happens on the 25th of April
4. What happens if you we do: `phone_book[("25th", "December")]`



# Project time!

You now know all about dictionaries!

**Let's put what we learnt into our project**  
**Try to do Part 3**

The tutors will be around to help!



Random!

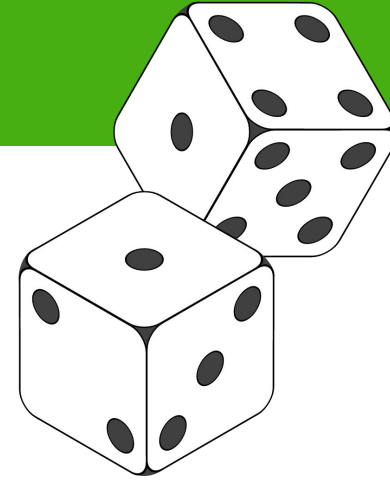


# That's so random!

There's lots of things in life that are up to chance or random!



Python lets us **import** common bits of code people use! We're going to use the **random** module!



We want the computer to be random sometimes!



# Using the random module



Let's choose something randomly from a list!

This is like drawing something out of a hat in a raffle!

## Try this!

### 1. Import the random module!

```
>>> import random
```

### 2. Copy the shopping list into your script

```
>>> shopping_list = ["eggs", "bread", "apples", "milk"]
```

### 3. Choose randomly! Try it a few times!

```
>>> random.choice(shopping_list)
```



# Using the random module



## You can also assign your random choice to a variable

```
>>> import random
>>> shopping_list = ["eggs", "bread", "apples", "milk"]
>>> random_food = random.choice(shopping_list)
>>> print(random_food)
```



# Project Time!

## **Raaaaaaaaaandom! Can you handle that?**

Let's try use it in our project!

Try to do Part 4

The tutors will be around to





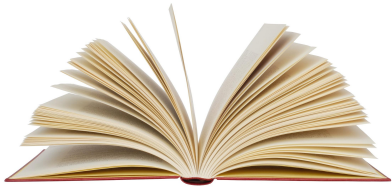
# For Loops



# For Loops

For loops allow you to do something **for a number of times or for each item in a group**

There are many real world examples, like:



**For each page in this book:  
Read**



**For each chip in this bag of chips:  
Eat**



# For Loops

```
number = 10  
for i in range(number):  
    #Do something
```



# For Loops

```
number = 10
for i in range(number):
    #Do something
```

The `for` word tells python we want to use a loop



# For Loops

This `i` is a temporary variable which will count how many times we have looped.

```
number = 10  
for i in range(number):  
    #Do something
```

The `for` word tells python we want to use a loop



# For Loops

```
number = 10  
for i in range(number):  
    #Do something
```

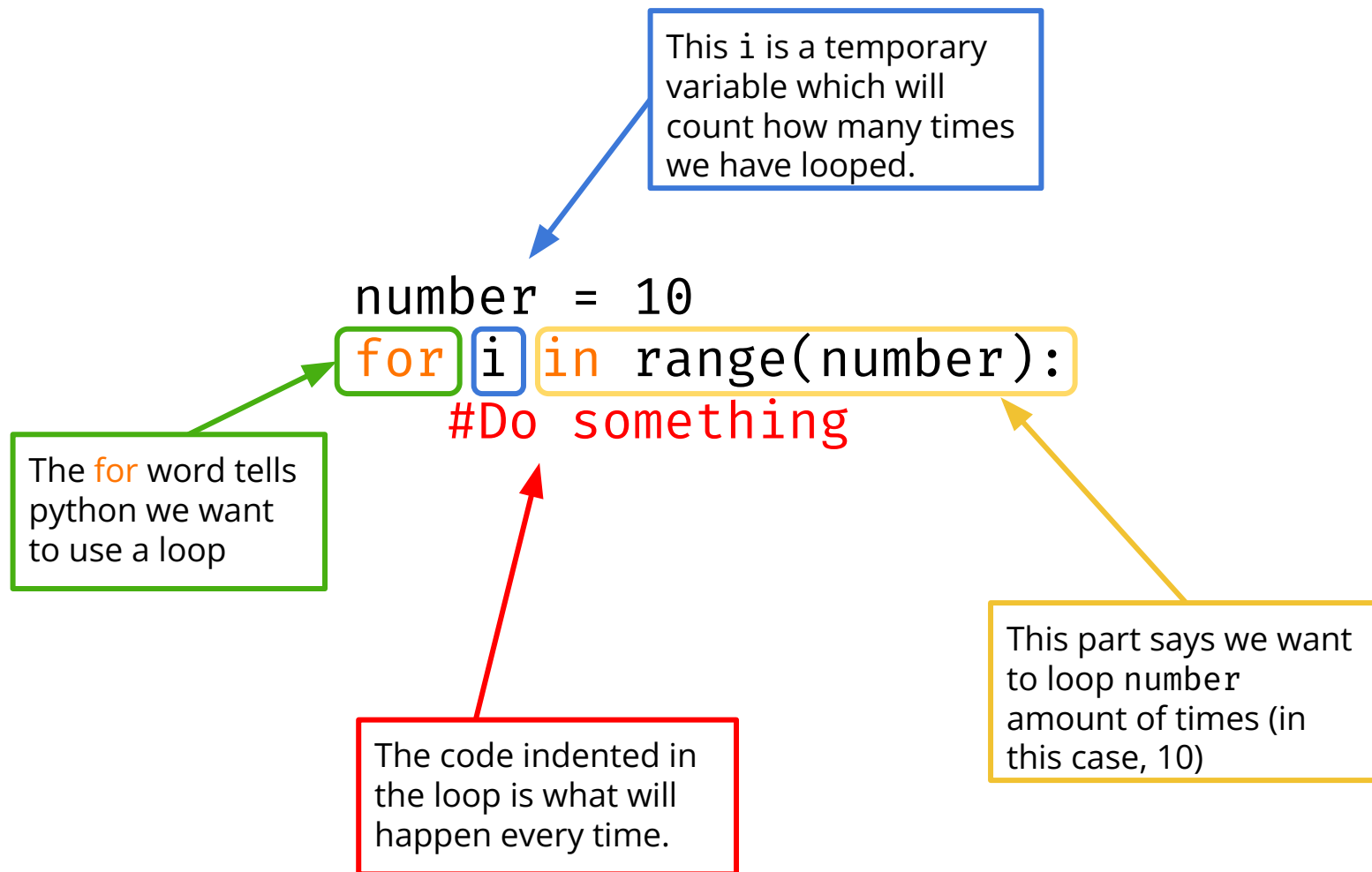
This i is a temporary variable which will count how many times we have looped.

The **for** word tells python we want to use a loop

This part says we want to loop number amount of times (in this case, 10)



# For Loops



# Looping how many times?

**We can loop through a list:**

```
friends = 4  
for i in range(friends):  
    print("Hello friend!")
```

What's going to happen?





# Looping how many times?

## We can loop through a list:

```
friends = 4  
for i in range(friends):  
    print("Hello friend!")
```

What's going to happen?

We do what's in the for loop as many times as what is in the "range"



# Looping how many times?

## We can loop through a list:

```
friends = 4  
for i in range(friends):  
    print("Hello friend!")
```

What's going to happen?

We do what's in the for loop as many times as what is in the "range"

```
>>> Hello friend!  
>>> Hello friend!  
>>> Hello friend!  
>>> Hello friend!
```



# Asking a question with a number answer!

It's common to ask the user to enter a number

**Input** always gives us a string of text

We need to turn the **string** into a number before we can use it as a range in a for loop

We do this by using **int()**

```
no_of_turns = int(input("How many times: " ))  
for i in range(no_of_turns)  
    Do something
```



# Project Time!

**Now you know how to use a for loop!**

**Try to do Parts 5 and 6  
...if you are up **for** it!**

The tutors will be around to help!



# While Loops



# Introducing ... while loops!

## What do you think this does?

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```



# Introducing ... while loops!

## What do you think this does?

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
i is 1
i is 2
>>>
```



# Introducing ... while loops!


Stepping through a while loop...



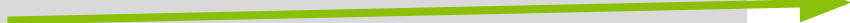


# Introducing ... while loops!

## One step at a time!



```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```



MY VARIABLES

i = 0

Set the  
variable



# Introducing ... while loops!

## One step at a time!

0 is less  
than 3!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

i = 0



# Introducing ... while loops!

## One step at a time!

Print!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
```

MY VARIABLES


```
i = 0
```



# Introducing ... while loops!

## One step at a time!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```



MY VARIABLES

~~i = 0~~  
i = 1

UPDATE  
TIME!

i is 0



# Introducing ... while loops!

## One step at a time!

Take it  
from the  
top!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
```

### MY VARIABLES

```
i = 0
i = 1
```



# Introducing ... while loops!

## One step at a time!

i is less  
than 3!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

```
i = 0
i = 1
```

```
i is 0
```



# Introducing ... while loops!

## One step at a time!

Print!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
i is 1
```

MY VARIABLES


```
i = 0
i = 1
```



# Introducing ... while loops!

## One step at a time!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```



### MY VARIABLES

~~i = 0~~  
~~i = 1~~  
i = 2

UPDATE  
TIME!

```
i is 0
i is 1
```





# Introducing ... while loops!

## One step at a time!

Take it  
from the  
top!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
```

```
i is 1
```

### MY VARIABLES

~~i = 0~~

~~i = 1~~

i = 2



# Introducing ... while loops!

## One step at a time!

2 is less  
than 3!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

```
i = 0
i = 1
i = 2
```

```
i is 0
i is 1
```



# Introducing ... while loops!

## One step at a time!

Print!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
i is 1
i is 2
```

### MY VARIABLES


```
i = 0
i = 1
i = 2
```



# Introducing ... while loops!

## One step at a time!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```



### MY VARIABLES

```
i = 0
i = 1
i = 2
i = 3
```

UPDATE  
TIME!

```
i is 0
i is 1
i is 2
```



# Introducing ... while loops!

## One step at a time!

Take it  
from the  
top!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

```
i is 0
i is 1
i is 2
```

### MY VARIABLES

```
i = 0
i = 1
i = 2
i = 3
```



# Introducing ... while loops!

## One step at a time!

3 IS NOT  
less than  
3!

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

MY VARIABLES

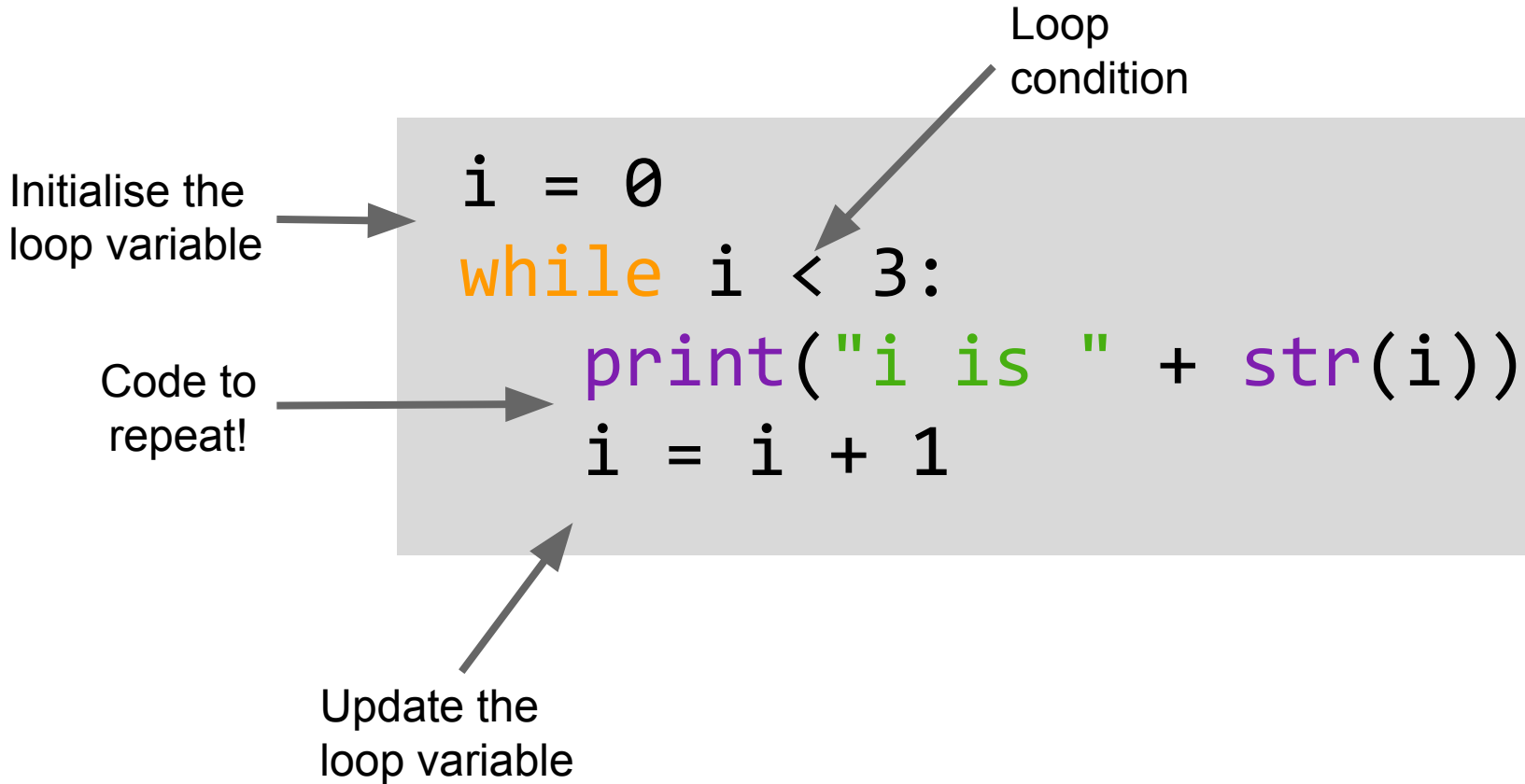
~~i = 0~~  
~~i = 1~~  
~~i = 2~~  
i = 3

We are  
done  
with this  
loop!

```
i is 0
i is 1
i is 2
```



# Introducing ... while loops!



The diagram illustrates the components of a while loop in Python. It features a light gray rectangular box containing the following code:

```
i = 0
while i < 3:
    print("i is " + str(i))
    i = i + 1
```

Four annotations with arrows point to specific parts of the code:

- Initialise the loop variable:** Points to the line `i = 0`.
- Code to repeat!:** Points to the indented block of code inside the loop: `print("i is " + str(i))` and `i = i + 1`.
- Loop condition:** Points to the condition `i < 3:` in the `while` statement.
- Update the loop variable:** Points to the line `i = i + 1` at the end of the loop block.

# What happens when.....

What happens if we forget to update the loop variable?

```
i = 0
while i < 3:
    print("i is " + str(i))
```





# What happens when.....

What happens if we forget to update the loop variable?

```
i = 0
while i < 3:
    print("i is " + str(i))
```

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0

i is 0



# Infinite loop!

## Sometimes we want our loop to go forever!

So we set a condition that is always True!

We can even just write True!

```
while True:  
    print("Are we there yet?")
```



# Project Time!

**while** we're here:

**Try to do Part 7!**

And the extensions

The tutors will be around to help!



Tell us what you think!

Click on the  
**End of Day Form**  
and fill it in now!