Guess Who!

Welcome to the Labs

Thank you to our Sponsors!

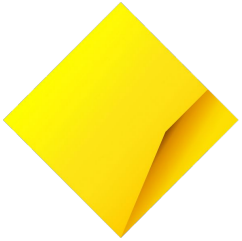
Platinum Sponsors:

ATLASSIAN

Australian Government

Australian Signals Directorate

Gold Sponsor:



Commonwealth Bank

Welcome to the Labs

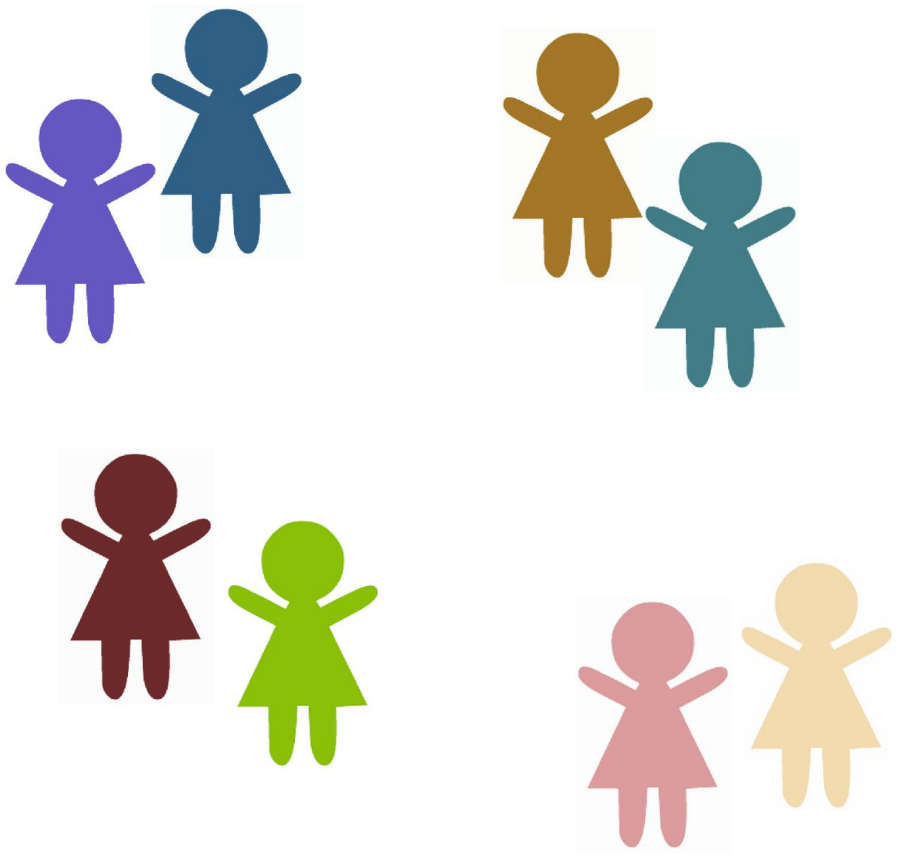
Guess Who!

Who are the tutors?

Who are you?

Introduce your partner

1. Find a partner (someone you've never met before)
2. Find out:
   a. Their name
   b. What (school) year they are in
   c. A fun fact about them!
3. Introduce them to the rest of the group!



Log on

Log on and jump on the GPN website

girlsprogramming.network/workshop

Click on your node location

Click on your room.

From this page you can see:

- These **slides** (to take a look back or go on ahead).
- A link to your **workbook** in EdStem
- Other helpful bits to use through the day!

Tell us you're here!

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

Start of Day Survey

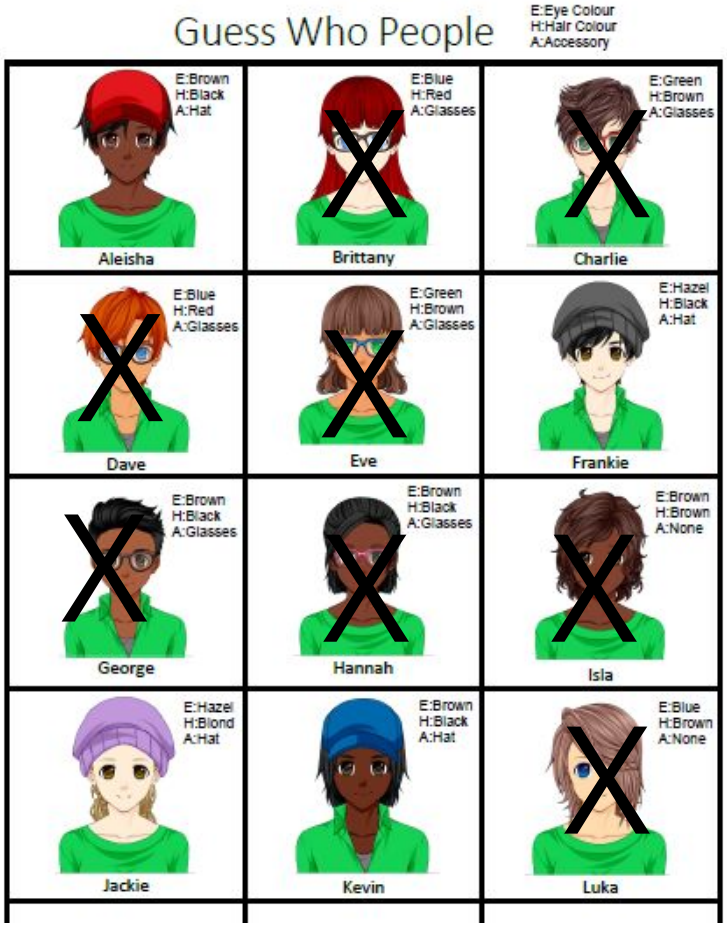
Today’s project!

Guess Who?

What is Guess Who?

- Fun Board Game
- We're going to code our own version
- One player picks a character - keep it secret!
- The other player asks questions to figure out who it is

What is Guess Who?



Q: Do they have a hat?

A: YES

Eliminates everyone without a hat

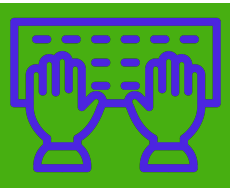
Keep asking questions until figure out who it is

Introduction to Edstem

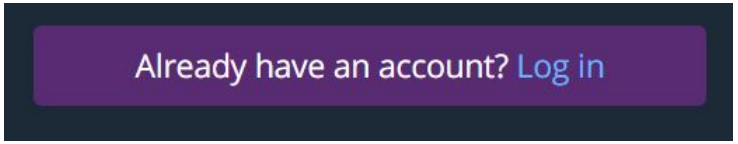
Log on

Click on your **Workbook** link to take you into EdStem

Signing up to Edstem

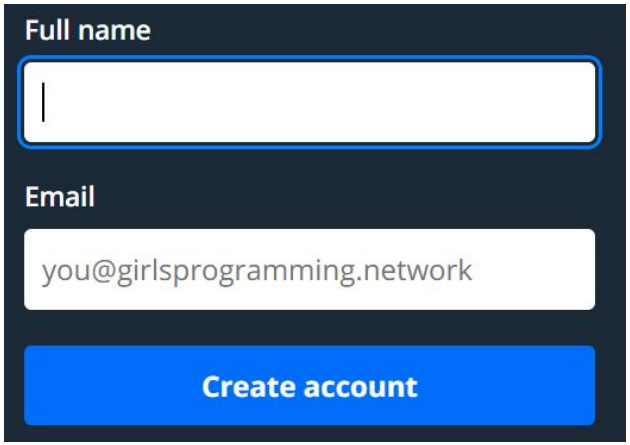


Log in if you already have a an “Edstem” account from a past GPN

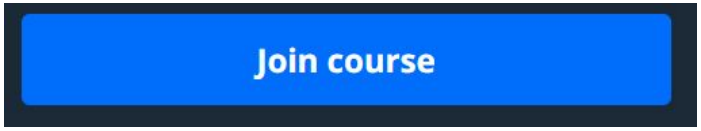


If you haven’t got an account, let’s make one:

1. Type in your Full Name
2. Type in your personal email
3. Click Create Account
4. Go to your email and verify your new account
5. Create a password



Click Join Course



The name of your course will be at the top : →

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 set up of the workbook

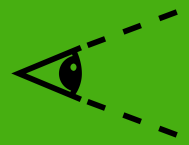
The main page:

1. Heading at the top that tells you the project you are in

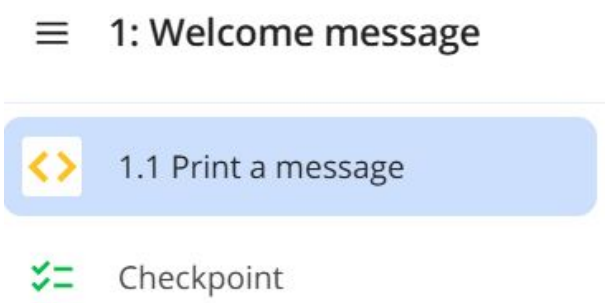
2. List of “Chapters” called something like **1:Welcome Message**
   They have an icon that looks like this:

3. 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:



- **Lessons** where you will do your coding.
  - They have this icon: 
- **Checkpoints** 

  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.

There may also be **Bonus Lessons** to try if you want to or if you are waiting for the next lecture

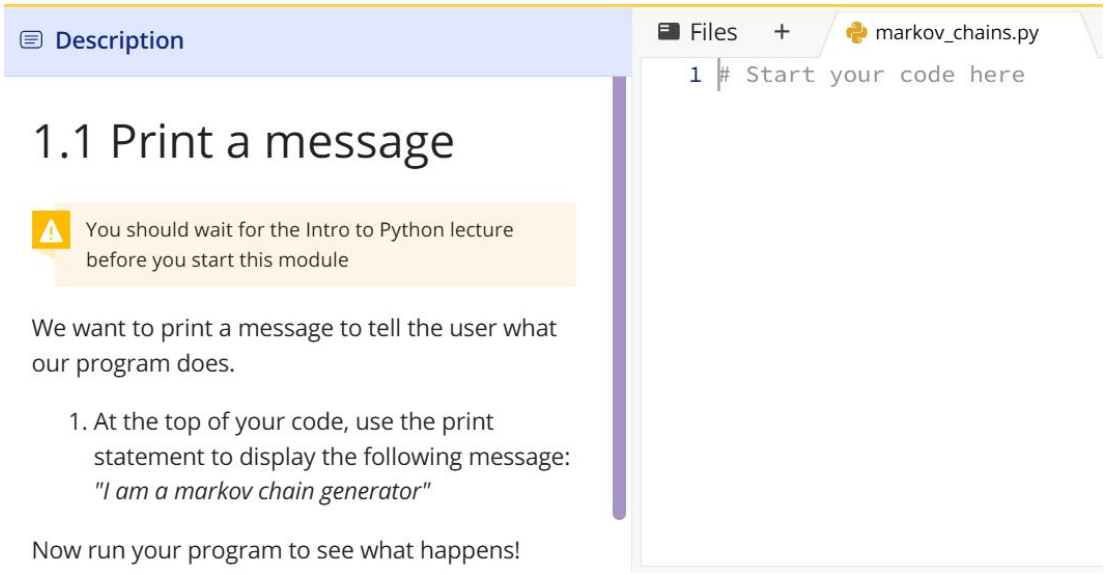
How to do the work

In each Lesson there is:

1. A section on the left with instructions
2. A section on the 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



Hints

Sometimes in a lesson, there's some code we want you to do that might be a bit tricky, to help you out we've added some hints. They look like this:

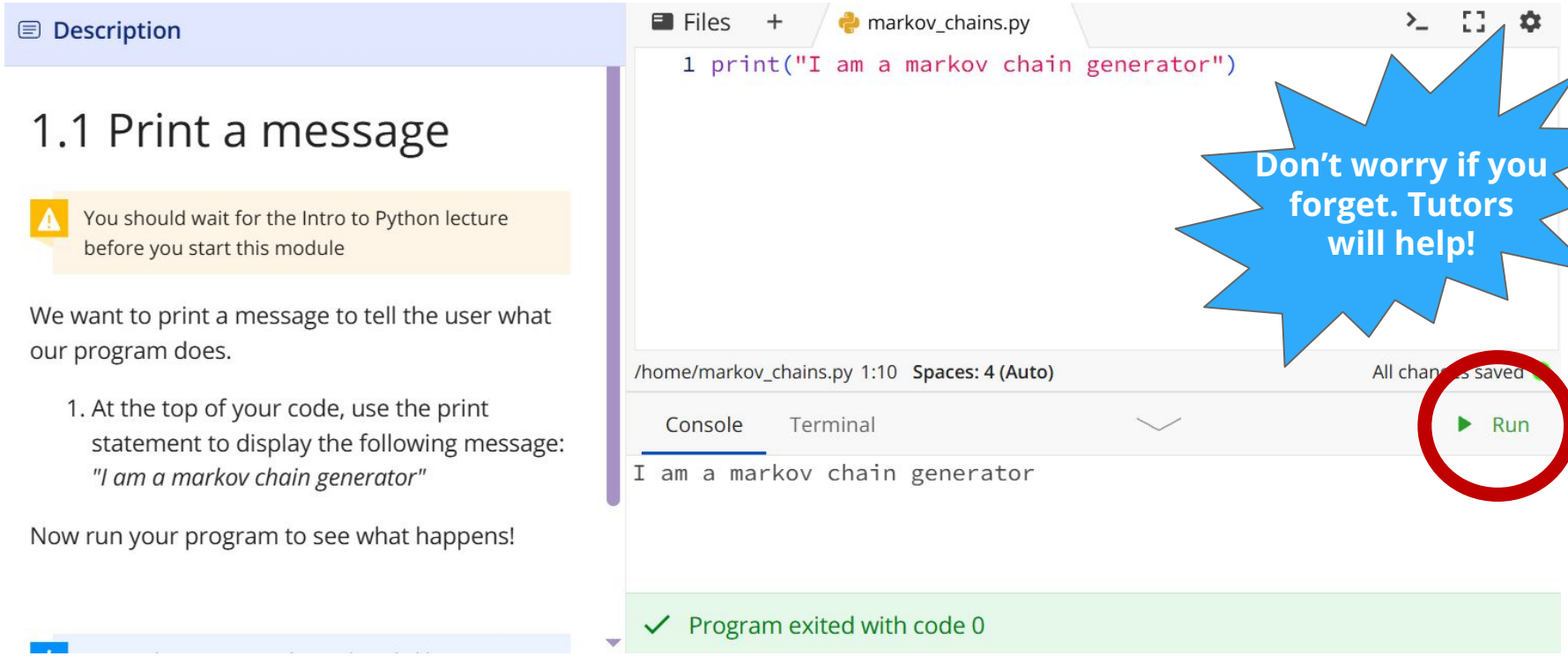
If you press the blue run button it will show you what that code does, you can even change the code to see if/how it changes.

These are **just hints** make sure you're not copying the hint into your code as it will likely end up breaking. They are just to show you the kinds of things you can do.

Running your code...

Click ▶ Run in the bottom right hand corner

Your code will run and any output will display in the Console



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

On Macs use Command (⌘) instead of Ctrl

Project time!

You now know all about the EdStem!

You should now sign up and join our EdStem class if you haven't done this already.

Remember the tutors will be around to help!

Classes

What is an object?

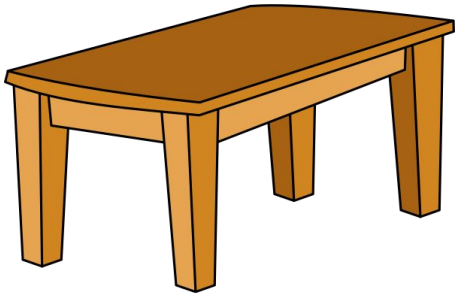
What do you think an object is?

What is an object?

What do you think an object is?

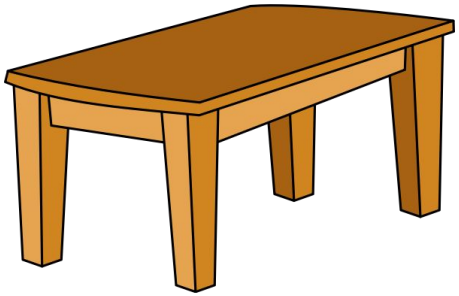
What is an object?

What do you think an object is?



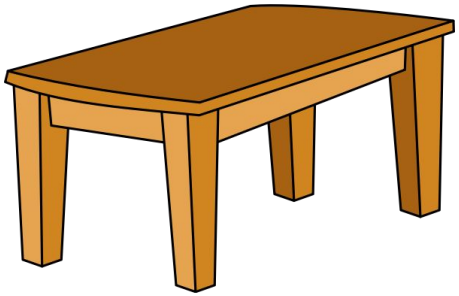
What is an object?

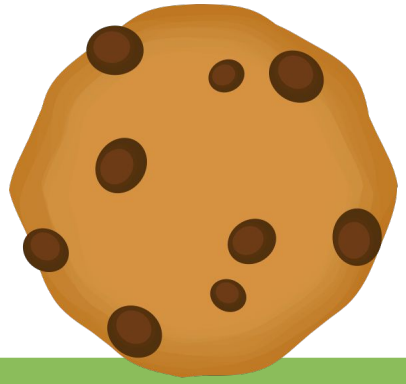
What do you think an object is?



What is an object?

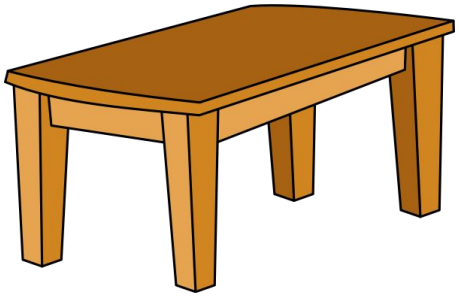
What do you think an object is?

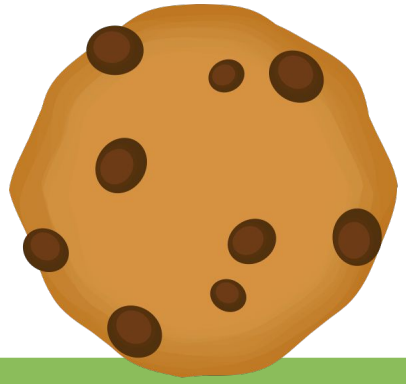




What is an object?

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

Owner

Age

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 information might we know about a cat?

Name

Owner

Age

Weight

Colour

Microchip #

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

Sleep

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

Sleep

Eat

Purr

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

Sleep

Eat

Purr

Scratch

Jump

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
```

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
```

Files

Filing it away!

What happens if we want to use different data in our program? What if that data is too big to write in with the keyboard?

We'd have to change our code!!

It would be better if we could keep all our data in a file and just be able to pick and choose what file we wanted to play today!

people.txt

```
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
```

Opening files!

To get access to the stuff inside a file in python we need to **open** it!

That doesn't mean clicking on the little icon!

```
with open("test.txt", "r") as f:
```

You'll now be able to read the things in f

If your file is in the same location as your code you can just use the name!

A missing file causes an error

Here we try to open a file that doesn't exist:

```
with open("missing.txt", "r") as f:
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IOError: [Errno 2] No such file or
directory: 'missing.txt'
```

You can read in one line at a time

You can use a for loop to read 1 line at a time!

```
with open("haiku.txt", "r") as f:
    for line in f:
        print(line)
```

```
Wanna go outside.

Oh NO! Help! I got outside!

Let me back inside!
```

Why is there an extra blank line each time?

Chomping off the newline

The newline character is represented by '\n':

```
print('Hello\nWorld')
Hello
World
```

We can remove it from the lines we read with .strip()

```
x = 'abc\n'
x.strip()
'abc'
```

x.strip() is safe as lines without newlines will be unaffected

Reading and stripping!

```
with open("haiku.txt", "r") as f:
    for line in f:
        line = line.strip()
        print(line)
```

```
Wanna go outside.
Oh NO! Help! I got outside!
Let me back inside!
```

No extra lines!

Project time!

I hope you **`file`**d that knowledge away

Use it in the next section of the project!

Try to do Part 1 and 2

The tutors will be around to help!

Random!

Using the random module

Let's choose something randomly from a list!

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

We use **random.choice** to randomly select something from a list

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

Each time we run this we would probably get a different answer

```
eggs OR bread OR apples OR milk
```

Using the random module

You can also assign your random choice to a variable and then use that variable in your code

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

The variable **`random_food`** contains the random choice that was made from the list

Methods

This is how we make our classes DO things

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)
```

4

Syntax cheatsheet

```
class MyClassName:
  staticVariable = someValueForEveryInstance
  def __init__(self, param1, param2...):
     # Set the instance variables
     self.myParam1 = param1
     self.someOtherValue = param2
  def someFunc(self, otherParam1, otherParam2...):
     # Do stuff here
     # You can even return values if you like!
```

Syntax cheatsheet

```
# Access static variables
MyClassName.staticVariable

# Create new instance of a class
mine = MyClassName(param1, param2...)

# Access an instance variable or function
mine.myParam1
mine.someFunc(otherParam1, otherParam2...)

# Store values from functions that return something
someValue = mine.someFunc(otherParam1, otherParam2...)
```

Project time!

I hope you now know the `method` to do the rest

Use it in the next section of the project!

Try to do Part 3 onwards

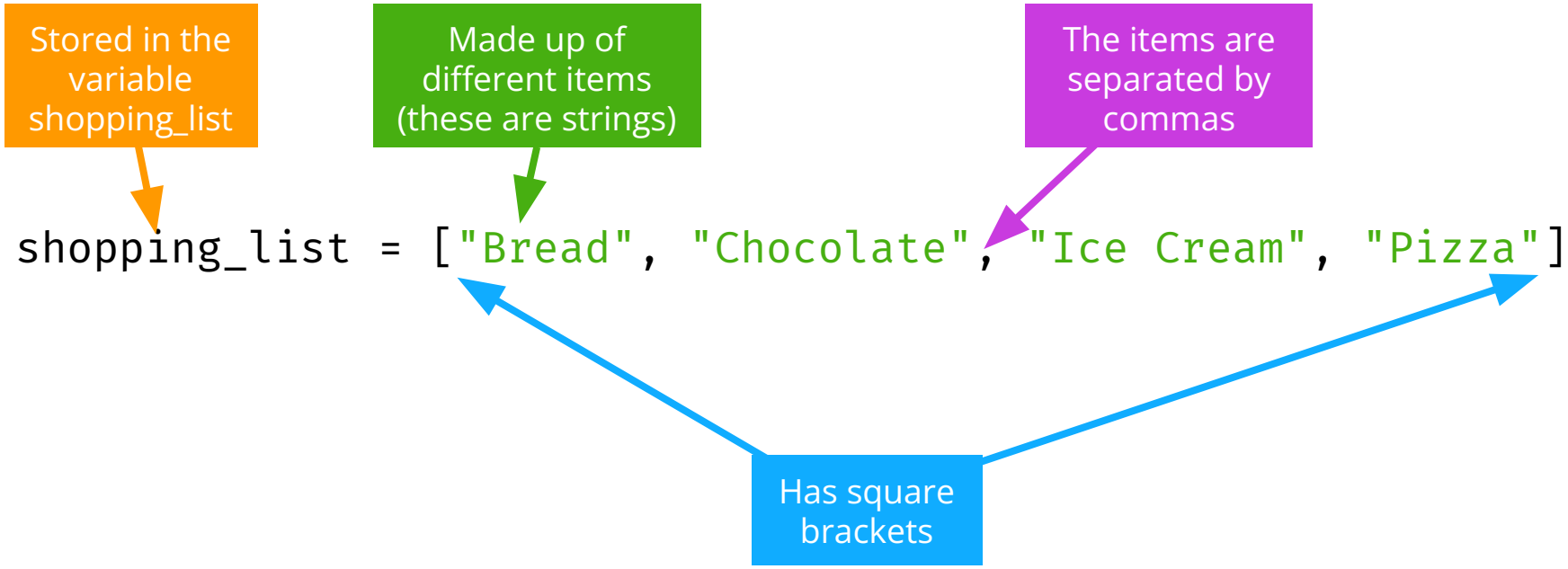
The tutors will be around to help!

Revision slides

Here are some handy concepts to jog your memory in case you've forgotten

The tutors will also be around to help!

List anatomy



Removing items!

We can remove items from the list if they're no longer needed!

What if we decided that we didn't like butterflies anymore?

```
>>> faves
['books', 'butterfly', 'lollipops', 'skateboard']
>>> faves.remove('butterfly')
```

What does this list look like now?

```
['books', 'lollipops', 'skateboard']
```

List of lists!

You really can put anything in a list, even more lists!

We could use a list of lists to store different sports teams!

```
tennis_pairs = [
    ["Alex", "Emily"], ["Kass", "Annie"], ["Amara", "Viv"]
]
```

Get the first pair in the list

```
>>> first_pair = tennis_pairs[0]
>>> ["Alex", "Emily"]
```

Now we have the first pair handy, we can get the first the first player of the first pair

```
>>> fist_player = first_pair[0]
>>> "Alex"
```

Booleans (True and False)

Computers store whether a condition is met in the form of

`True` and `False`

To figure out if something is `True` or `False` we do a comparison

5 < 10	True	"Dog" == "dog"	False
3 + 2 == 5	True	"D" in "Dog"	True
5 != 5	False	"Q" not in "Cat"	True

Else statements

`else`
Statements means something still happens if the `if` statement was `False`

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

```
What happens??
>>> The word isn’t GPN :(
```

Elif statements

elif
Means we can give specific instructions for other words

```
word = "Chocolate"
if word == "GPN":
  print("GPN is awesome!")
elif word == "Chocolate":
  print("YUMMM Chocolate!")
else:
  print("The word isn’t GPN :(")
```

```
What happens??
>>> YUMMM Chocolate!
```

Looping over a list of ints

Strings are lists of letters!

```
word = "cat"
for i in word:
    print(i)
```

What's going to happen?

Looping over a list of ints

Strings are lists of letters!

```
word = "cat"
for i in word:
    print(i)
```

What's going to happen?

```
>>> c
>>> a
>>> t
```

How does it work??

Somehow it knows how to get one fruit out at a time!!

It's like it knows english!

```
fruits = ['apple', 'banana', 'mango']
for fruit in fruits:
    print('yummy ' + fruit)
```

But fruit is just a variable! We could call it anything! Like dog!

```
fruits = ['apple', 'banana', 'mango']
for dog in fruits:
    print('yummy ' + dog)
```

```
>>> Yummy apple
>>> Yummy banana
>>> Yummy mango
```