CISSA

SSS

Intro to Programming

Beginner-friendly workshop in Python!

Tuesday 5th April, starting at 5:20 PM

CISSA

- Largest Tech Club in UoM.
- Representative of Computer Science / Data Science students in both undergrad and postgrad.
- Joint events and collaborations with…
  - Tech Companies (Google, Microsoft, Amazon, Canva etc.)
  - Consulting Firms (Deloitte)
  - Trading Companies (Optiver, Jane Street, IMC etc.)
  - Researchers in UoM
- Social events for you to make more friends in the tech community!

Join Our Club (it's FREE)!

- Access our Discussion Space on Facebook / Discord to connect with over 4000 people in the UoM tech community.
- Ask questions re. Subjects, Careers, Interviews etc.
- Events:
  - Industry Connect & Tech talks
  - Mock Interviews + Office Tours
  - Hackathons - Codebrew + Catalyst
  - Start-Up & Entrepreneurship Competitions
  - Subject Revision + Interview Workshops

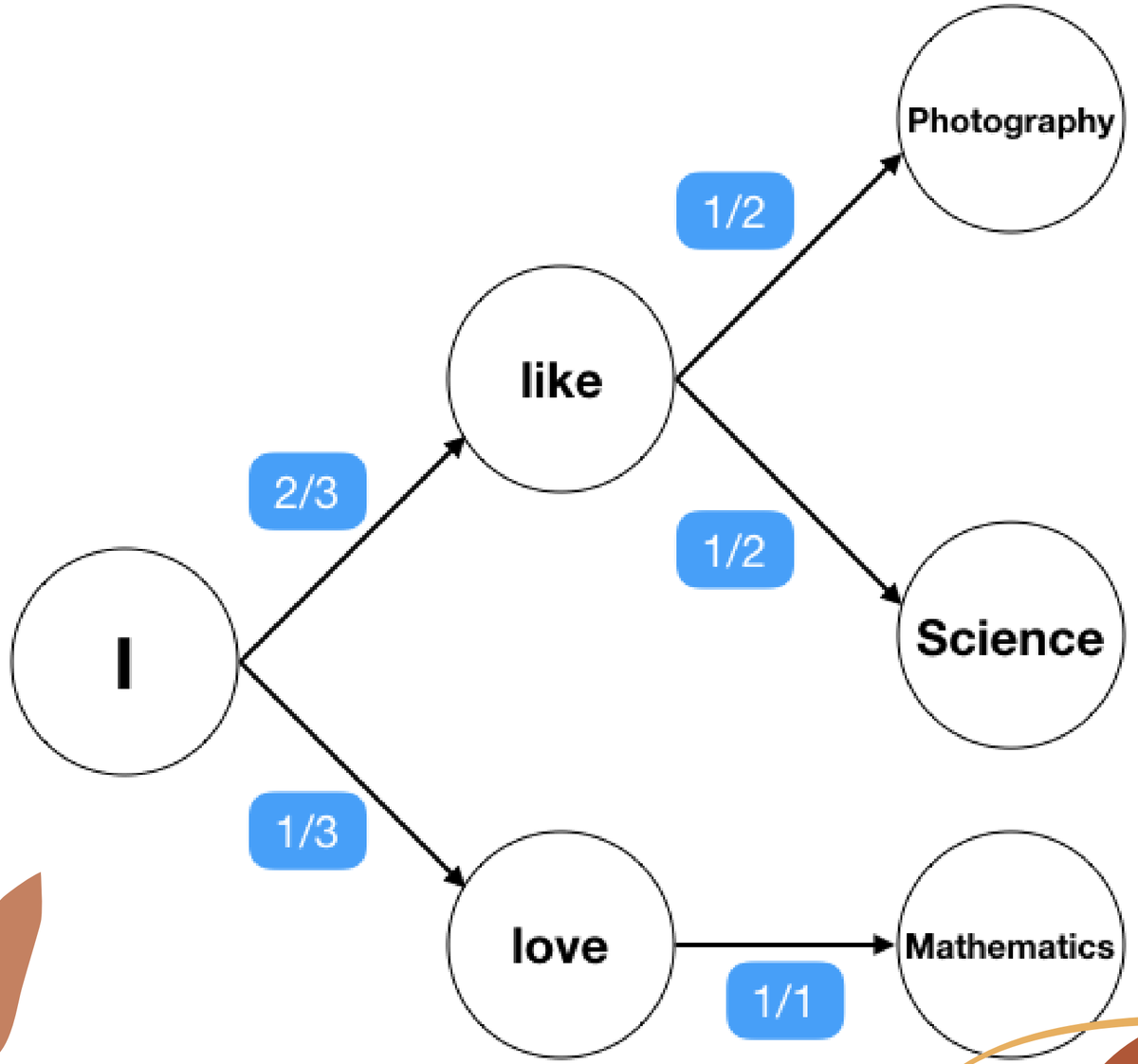
Sign-Up!

https://cissa.org.au/signup

Today's Project: Markov Chains!

Markov chains allow us to see all the different transitions between states.

Let's see an example!



Markov Chain Example

Using Markov Chains, we can generate some text in the style of Dr Seuss

Text: "one fish two fish red fish blue fish this one has a little car"

Which words follow each word in the text?

Hint: The last word in the text should connect to the first word so that we don't just stop when we get to car.

one	
two	
red	
blue	
this	
has	
a	
little	
car	
fish	

Markov Chain Example

Text: "one fish two fish red fish blue fish this one has a little car"

Choose a starting word e.g. "one"

- Randomly choose the next word from the possible options

- Can generate some fun text e.g.
- "one fish two fish this one has a little car one fish red fish blue has a little car..."

one	**fish, has**
two	**fish**
red	**fish**
blue	**fish**
this	**one**
has	**a**
a	**little**
little	**car**
car	**one**
fish	**two, red, blue, this**

Markov Chain Example

- Markov chains are exactly what we just did with the cups!
- This is what we'll be making the computer do in Python today!
- Here’s one we made from some Shakespeare!

```
doth stay! All days when I compare thee to unseeing eyes be
blessed made By chance, or eyes can see, For all the top of
happy show thee in dark directed. Then thou, whose shadow
shadows doth stay! All days when I compare thee in your self in
inward worth nor outward fair, Can make bright, How would thy
shade Through heavy sleep on the eye of life repair, Which this,
Time's pencil, or my pupil pen, Neither in the living day, When
in eternal lines of that fair from fair thou grow'st, So should
the lines to a summer's day?
```

What is Programming?

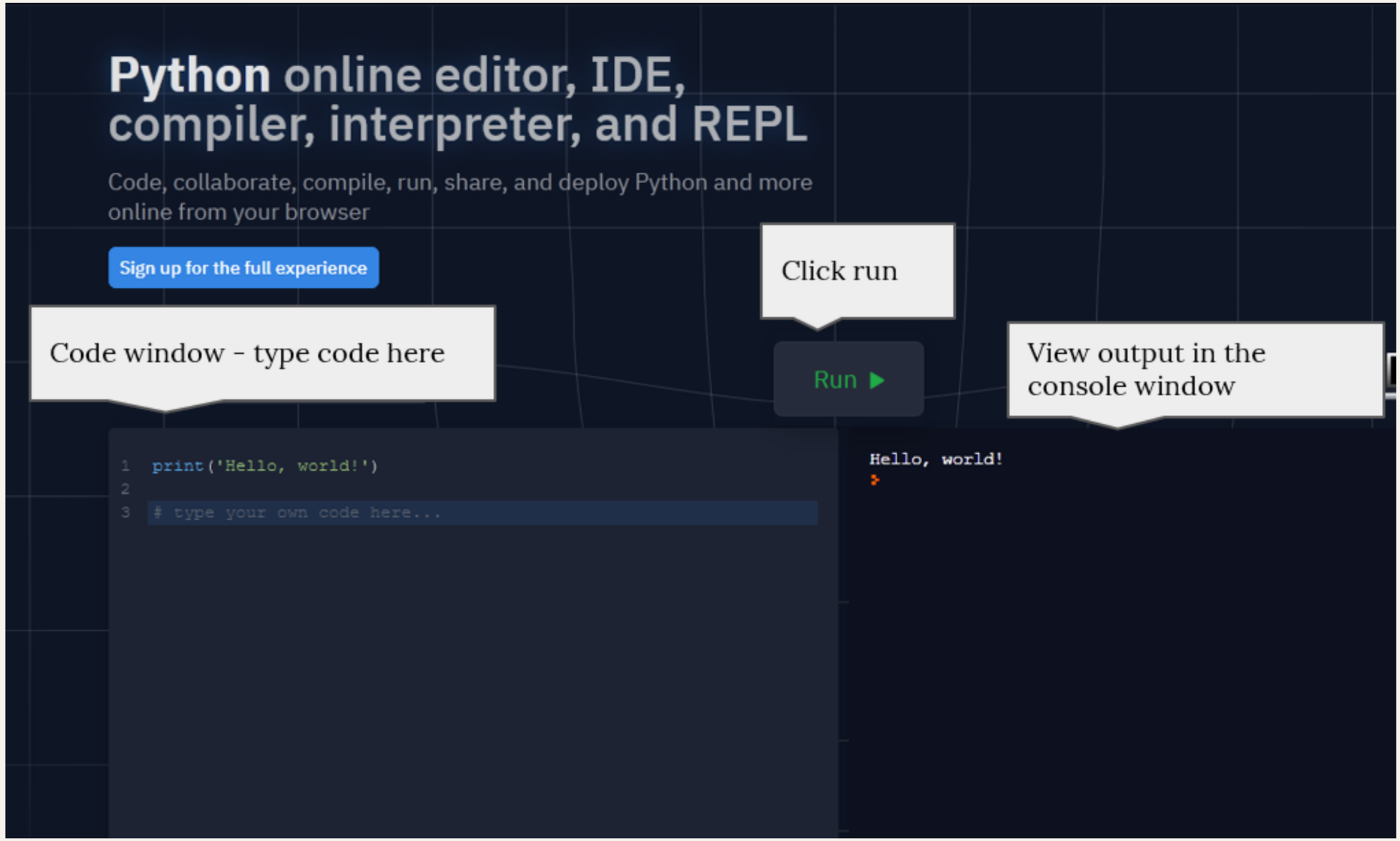
- Programming is giving computers a set of instructions!
- Very much like a recipe!
  - Computers will do exactly what you tell it to do, in that order

- We'll be learning Python!
- Very popular programming language
  - Can transfer many concepts to other languages e.g. C, Java

- We'll be coding here: **https://repl.it/languages/python3**

Python IDE

IDE is an Integrated Development Environment



Why learn to code?

- Can be used to automate tasks
- Very useful in pretty much any discipline and industry
  - Data analysis, simulations, creating websites and apps
- It's fun!
- You can solve a lot of problems by coding up a solution
  - This can require lots of logical thinking and creativity!

Make a mistake!

- Delete the content in the editor pane

- Type by button mashing the keyboard!
- Then press enter!

```
asdf asdjlkj;pa j;k4uroei
```

- Did you get a big red error message?

Mistakes are great!

- **Good work you made an error!**

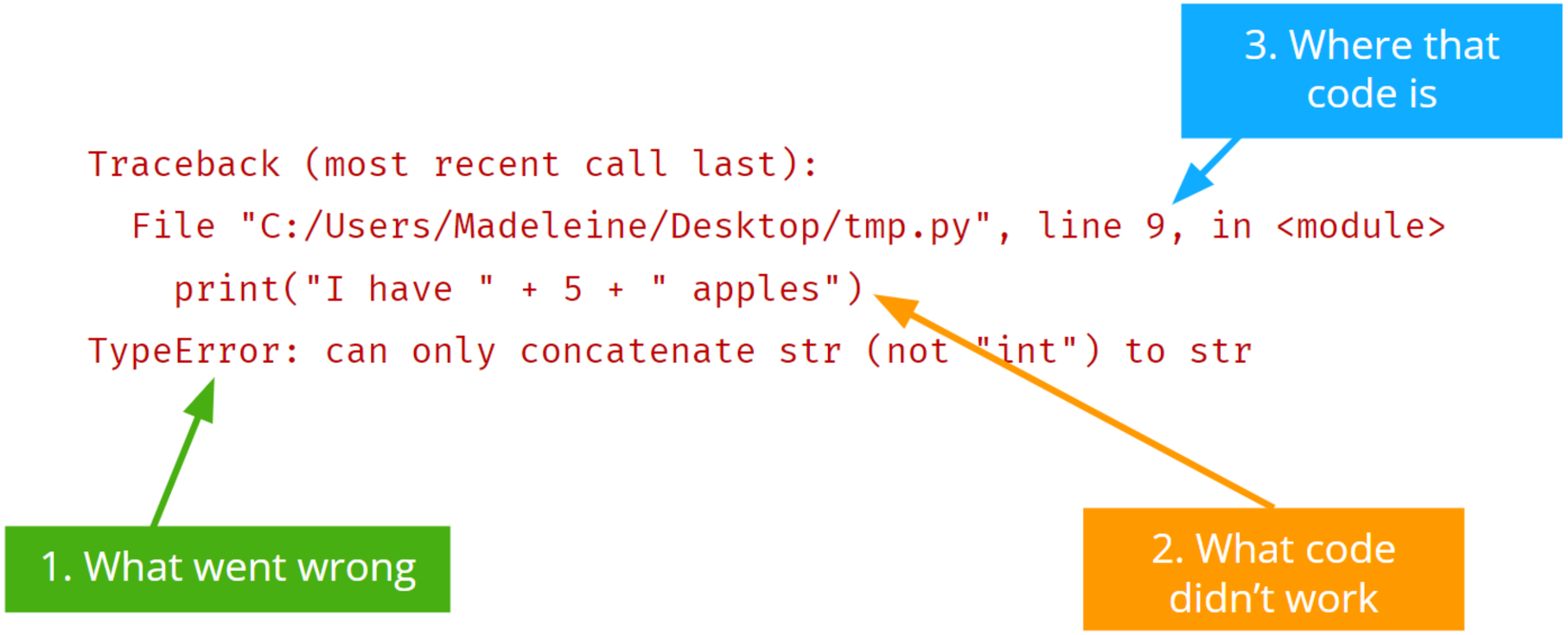
- Programmers make A LOT of errors!
- Errors give us hints to find mistakes
- Run your code often to get the hints!!
- Mistakes won’t break computers!

  - `ImportError: No module named randm`
  - `SyntaxError: Invalid Syntax`
  - `KeyError: ‘Sydny’`
  - `TypeError: Can't convert 'int' object to str implicitly`

We can learn from our mistakes!

Error messages help us fix our mistakes!

We read error messages from bottom to top



Hello World!

This is the first bit of code we will do. What do you think it does?

```
print('hello world')
```

Hello World!

This is the first bit of code we will do. What do you think it does?

```
print('hello world')
```

It prints the words “hello world” onto the screen!

Printing

We can `print` things in lots of different ways in python!

```
>>> print("Hello world!")
Hello world!
>>> print("Hello", "world!")
Hello world!
>>> print("Hello", "world", end="!")
Hello world!
```

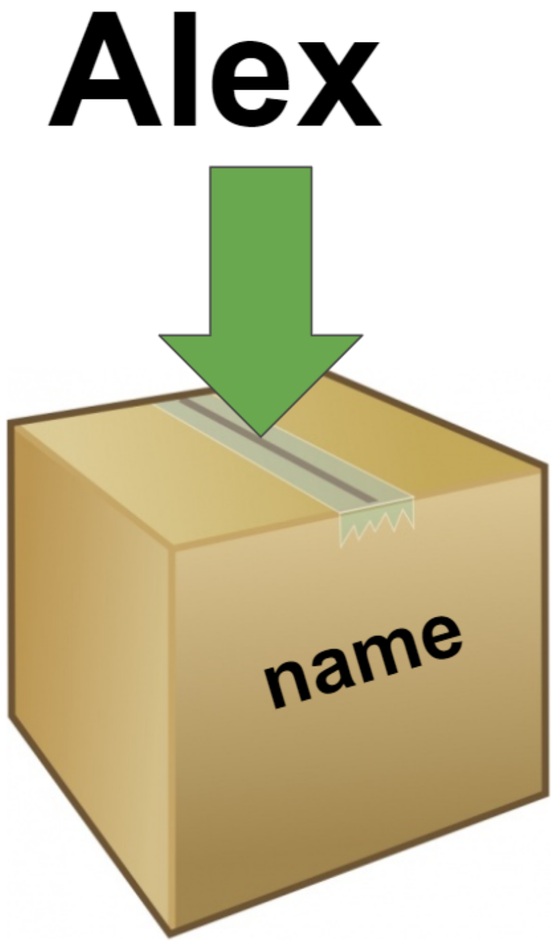
Note that this last one will not have a new line after it!

Variables

Variables are useful for storing things that change

(i.e. things that "vary" - hence the word "variable")

You can think of it like putting information in a box and giving it a name




```
name = "Alex"
```

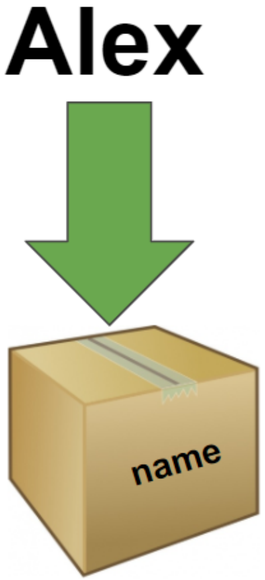
Variables

```
name = "Alex"
```

Instead of writing a name, we can use the name that is inside our variable! Here, we get the name out of the box.

```
print(name)
```

```
Alex
```



Reusing Variables

We can replace values in variables:

```
animal = "dog"
print("My favourite animal is a " + animal)
animal = "cat"
print("My favourite animal is a " + animal)
animal = animal + "dog"
print("My favourite animal is a " + animal)
```

What will this output?

Reusing Variables

We can replace values in variables:

```
animal = "dog"
print("My favourite animal is a " + animal)
animal = "cat"
print("My favourite animal is a " + animal)
animal = animal + "dog"
print("My favourite animal is a " + animal)
```

What will this output?

```
My favourite animal is a dog
My favourite animal is a cat
My favourite animal is a catdog
```

Asking a question!

It's more fun when we get to interact with the computer!

Let's get the computer to ask us a question!

```
my_name = input('What is your name? ')
print('Hello ' + my_name)
```

What do you think happens?

Asking a question!

It's more fun when we get to interact with the computer!

Let's get the computer to ask us a question!

```
my_name = input('What is your name? ')
print('Hello ' + my_name)
```

What do you think happens?

```
What is your name? Maddie
Hello Maddie
```

Asking a question!

Store the answer in the variable my_name

Writing input tells the computer to wait for a response

This is the question you want printed to the screen

```
my_name = input('What is your name? ')
print('Hello ' + my_name)
```

What do you think happens?

```
What is your name? Maddie
Hello Maddie
```

We can use the answer the user wrote that we then stored later!

Comments!

Sometimes we want to write things in code that the computer doesn't look at! We use **comments** for that!

Use comments to write a note or explanation of our code

Comments make code easier for humans to understand

```
# This code was written by Sheree
```

We can make code into a comment if we don't want it to run (but don't want to delete it!)

```
# print("Goodbye world!")
```

Coding Time!

You now know all about printing, variables and input!

Try attempting **Part 1 and 2**

The tutors will be around to help!

Task Sheet for Coding Parts:
shorturl.at/cuL13

If Statements

Conditions!

Conditions let us make decision.

First we test if the condition is met!

Then maybe we'll do the thing

<u>If it's raining</u> take an umbrella

Yep it's raining

..... take an umbrella

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	"Dog" == "dog"
3 + 2 == 5	"D" in "Dog"
5 != 5	"Q" not in "Cat"

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

Conditions

So to know whether to do something, they find out if it's `True`!

```
fave_num = 5
if fave_num < 10:
    print("that's a small number")
```

Conditions

So to know whether to do something, they find out if it's `True`!

```
fave_num = 5
if fave_num < 10:
    print("that's a small number")
```

That's the condition!

Is it `True` that fave_num is less than 10?

- Well, fave_num is 5
- And it's `True` that 5 is less than 10
- So it is `True`!

Conditions

So to know whether to do something, they find out if it's `True`!

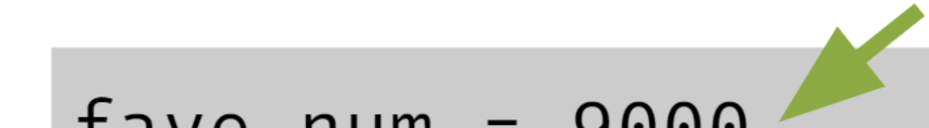
```
fave_num = 5
if True:
    print("that's a small number")
```

What do you think happens?

```
>>> that's a small number
```

Conditions

How about a different number???



```
fave_num = 9000
if fave_num < 10:
    print("that's a small number")
```

Conditions

Find out if it's `True`!

```
fave_num = 9000
if False:
    print("that's a small number")
```

Put in the answer to the question

Is it `True` that fave_num is less than 10?

- Well, fave_num is 9000
- And it's not `True` that 9000 is less than 10
- So it is `False`!

Conditions

How about a different number???

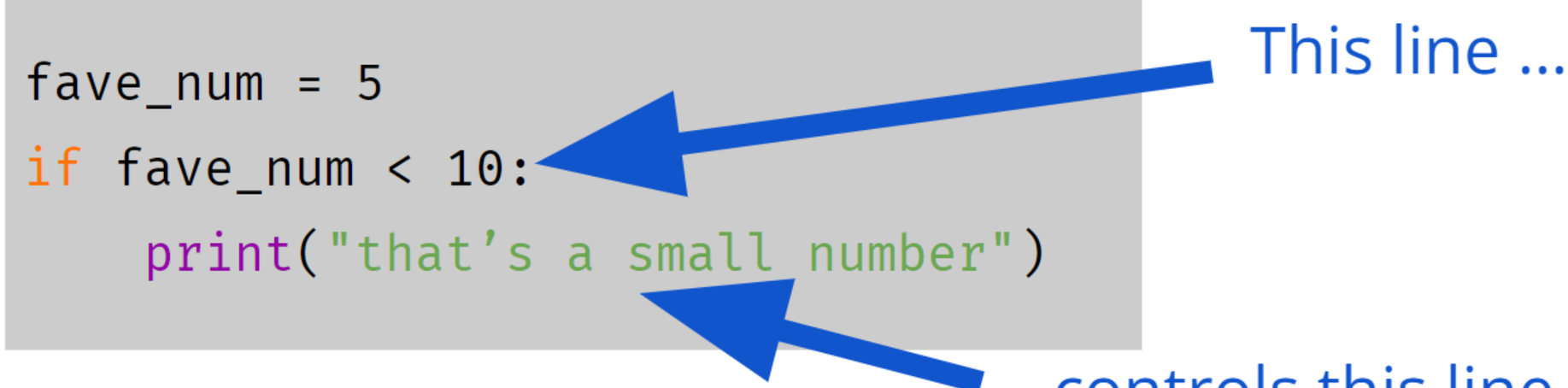
```
fave_num = 9000
if fave_num < 10:
    print("that's a small number")
```

What do you think happens?

```
>>>
```

If Statements

```
fave_num = 5
if fave_num < 10:
    print("that's a small number")
```



If Statements

Actually .....

```
fave_num = 5
if fave_num < 10:
    print("that's a small number")
    print("and I like that")
    print("A LOT!!")
```

This line ...

... controls anything below it
that is indented like this!

If Statements

Actually .....

```
fave_num = 5
if fave_num < 10:
    print("that's a small number")
    print("and I like that")
    print("A LOT!!")
```

This line ...

... controls anything below it
that is indented like this!

What do you think happens?

If Statements

All get printed out!

```
fave_num = 5
if fave_num < 10:
    print("that's a small number")
    print("and I like that")
    print("A LOT!!")
```

```
>>> that's a small number
>>> and I like that
>>> A LOT!!
```

If Statements

```
word = "CISSA & SSS"
if word == "CISSA & SSS":
  print("CISSA & SSS are awesome!")
```

What happens?

```
>>> CISSA & SSS are awesome!
```

If Statements

```
word = "CISSA & SSS"
if word == "CISSA & SSS":
  print("CISSA & SSS are awesome!")
```

What happens?

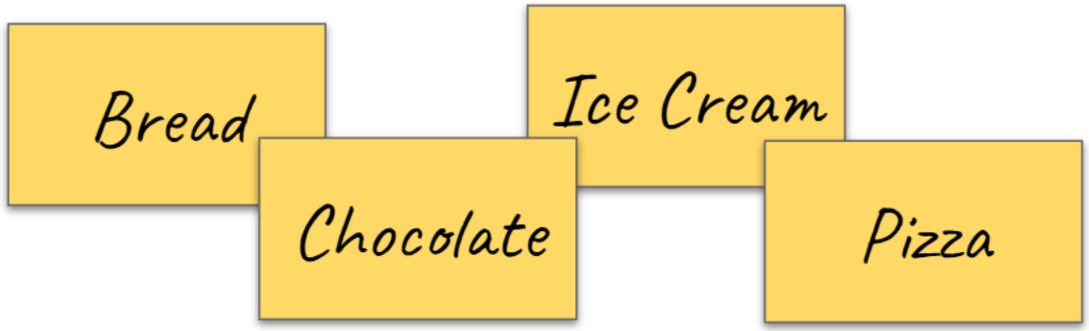
```
>>> CISSA & SSS are awesome!
```

Lists

Lists

When we go shopping, we write down what we want to buy!

But we don't store it on lots of little pieces of paper!



We put it in one big shopping list!

Lists

It would be annoying to store it separately when we code too

```
>>> shopping_item1 = "Bread"
>>> shopping_item2 = "Chocolate"
>>> shopping_item3 = "Ice Cream"
>>> shopping_item4 = "Pizza"
```

So much repetition!

Instead we use a python list!

```
>>> shopping_list = ["Bread", "Chocolate", "Ice Cream",
"Pizza"]
```

Lists

It would be annoying to store it separately when we code too

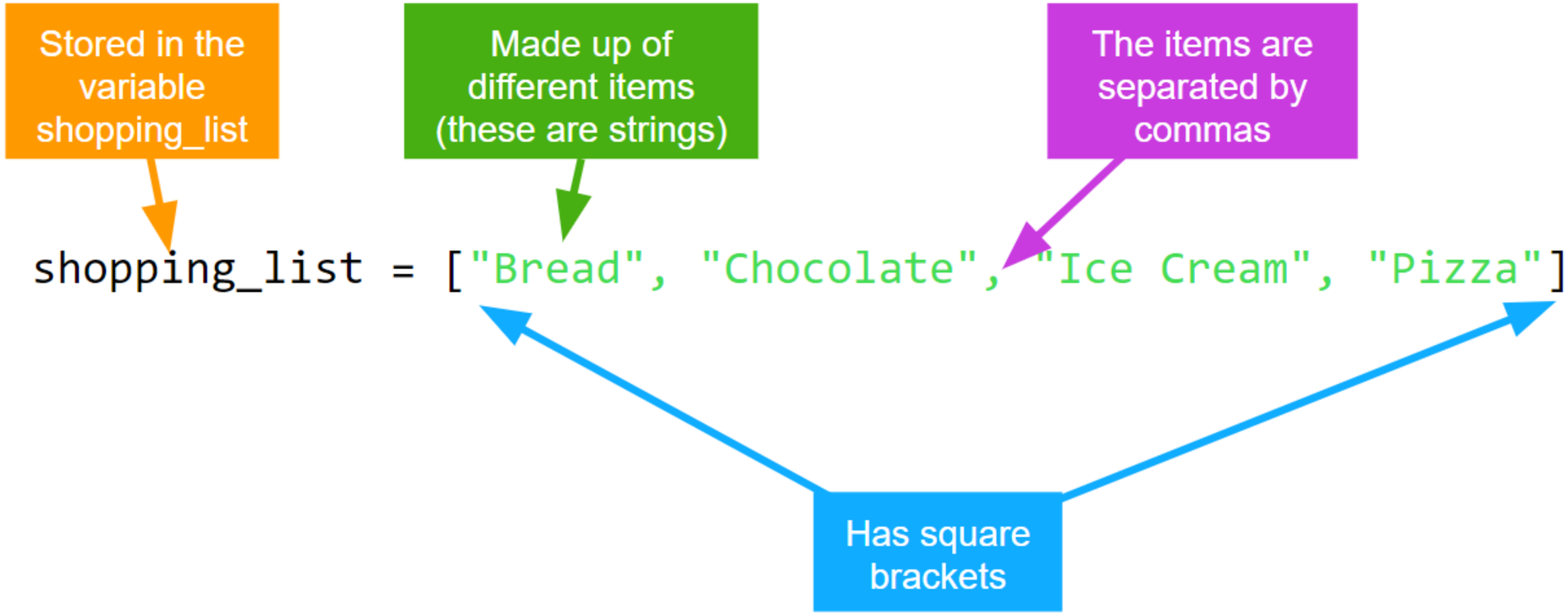
```
>>> shopping_item1 = "Bread"
>>> shopping_item2 = "Chocolate"
>>> shopping_item3 = "Ice Cream"
>>> shopping_item4 = "Pizza"
```

So much repetition!

Instead we use a python list!

```
>>> shopping_list = ["Bread", "Chocolate", "Ice Cream",
"Pizza"]
```

Lists


Stored in the variable shopping_list
Made up of different items (these are strings)
The items are separated by commas
shopping_list = ["Bread", "Chocolate", "Ice Cream", "Pizza"]
Has square brackets

Coding Time!

You now know all about if statements and lists!

Try attempting **Part 3**

The tutors will be around to help!

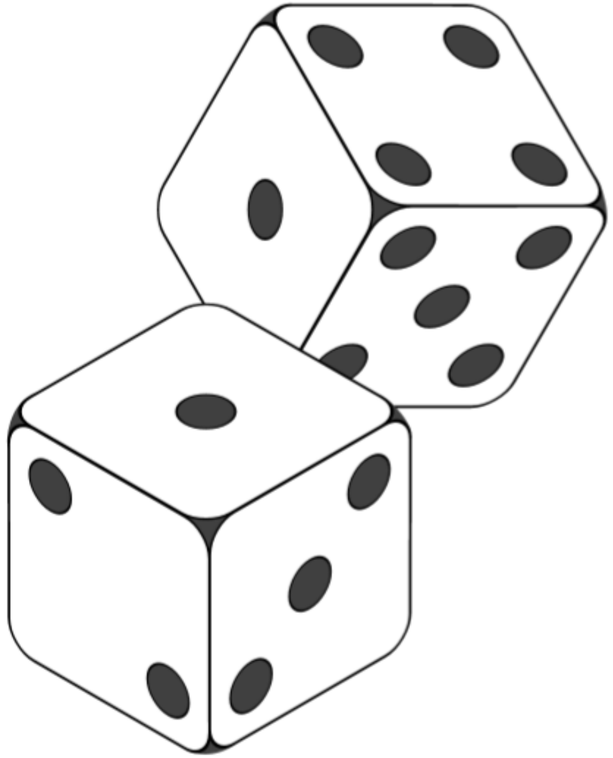
Coding Time!

You now know all about if statements and lists!

Try attempting **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!

We want the computer to be random sometimes!

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

Using the random module

Let’s choose something randomly from a list!

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

Here's an example!

1. **Import the random module!**

```
>>> import random
```

2. **Copy the shopping list into IDLE**

```
>>> 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)
```

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)
```

Coding Time!

You now know all about random!

Try attempting **Part 4**

The tutors will be around to help!

Coding Time!

You now know all about random!

Try attempting **Part 4**

The tutors will be around to help!

For Loops

For Loops

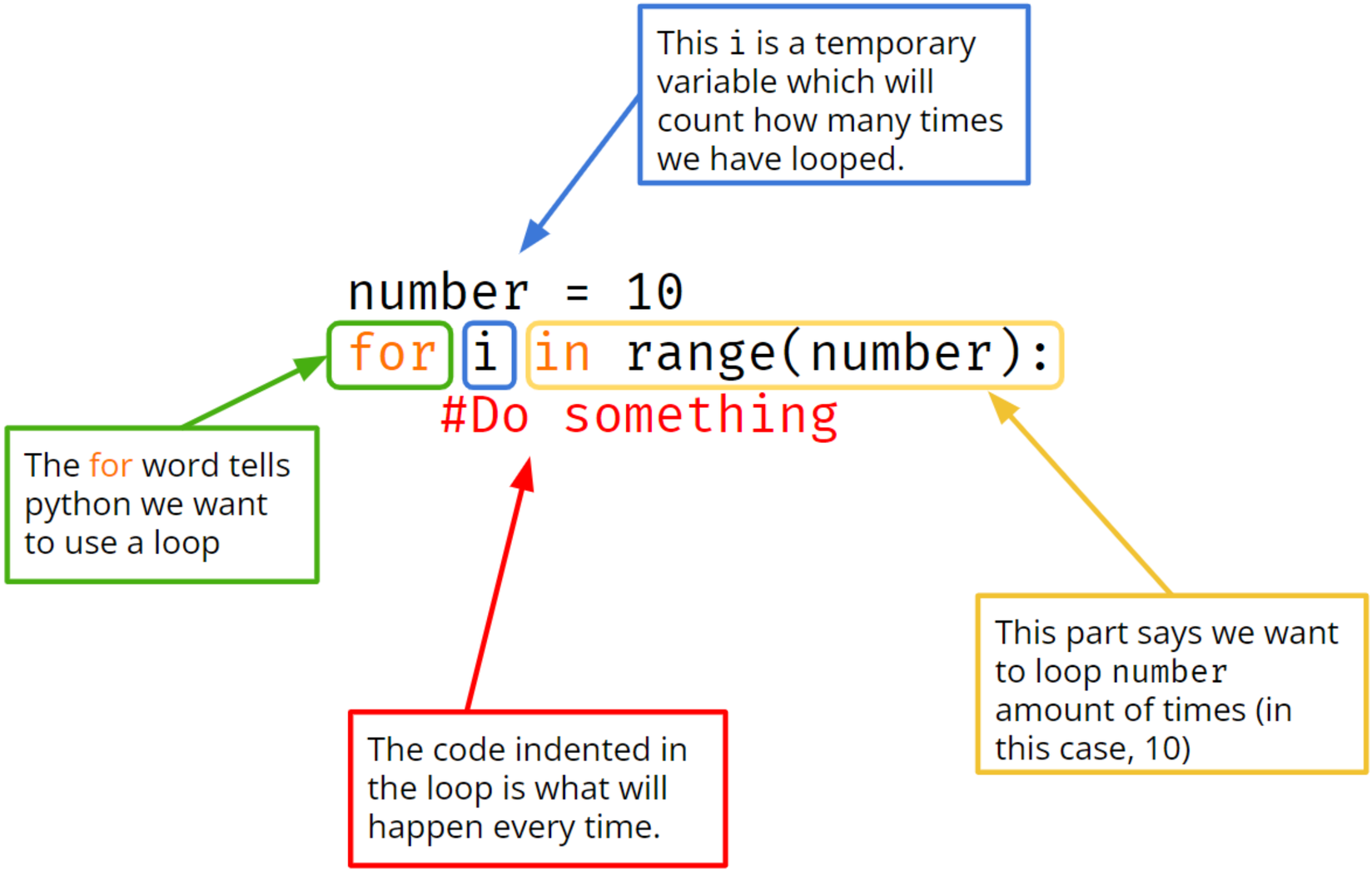
For loops allow you to do something a certain number of times.

We use them when we know exactly how many times we want to do something!

For Loops

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

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
The code indented in the loop is what will happen every time.
This part says we want to loop number amount of times (in this case, 10)

For Loops

We can loop through a list:

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

What’s going to happen?

For Loops

We can loop through a list:

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

What’s going to happen?

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

We do what’s in the for loop as many times as what is in the “range”

For Loops

We can loop through a list:

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

What’s going to happen?

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

We do what’s in the for loop as many times as what is in the “range”

Coding Time!

You now know all about for loops!

Try attempting **Part 5**

The tutors will be around to help!

Coding Time!

You now know all about for loops!

Try attempting **Part 5**

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

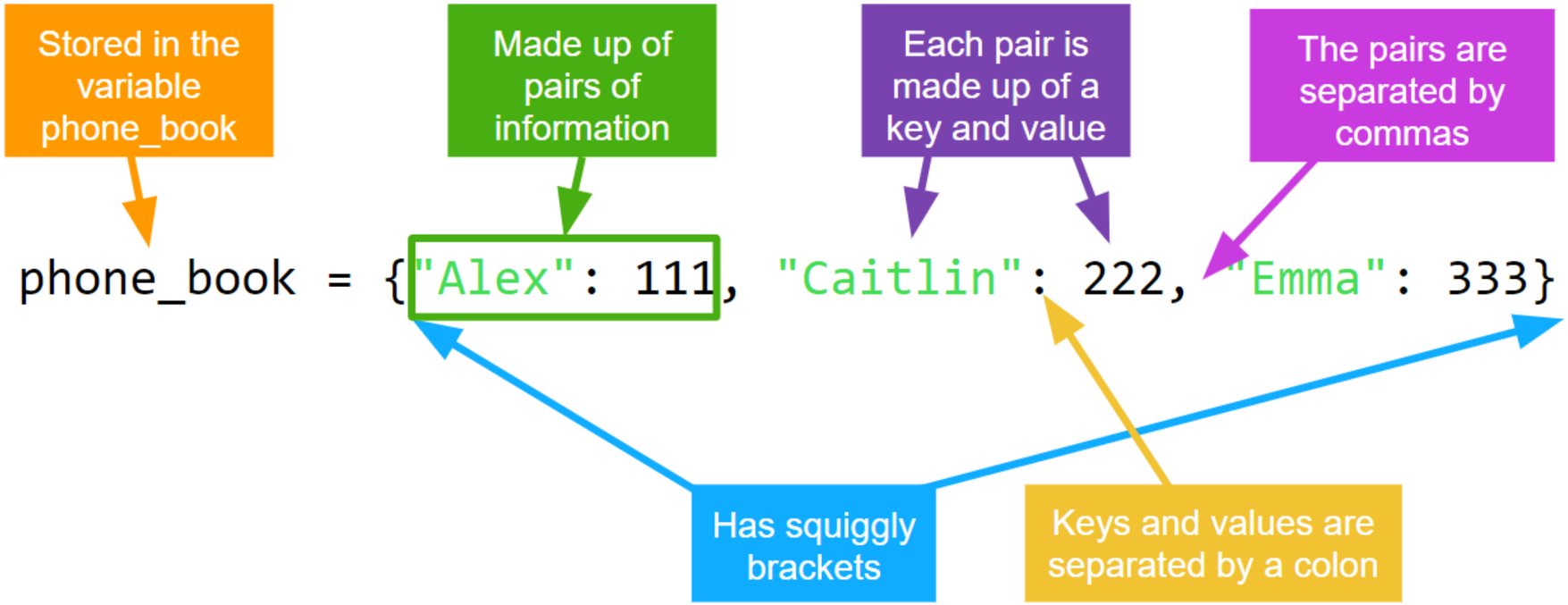
Key

Value

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

Dictionaries in Python

This is a python dictionary!



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

Dictionaries in Python

Let’s try using the phone book!

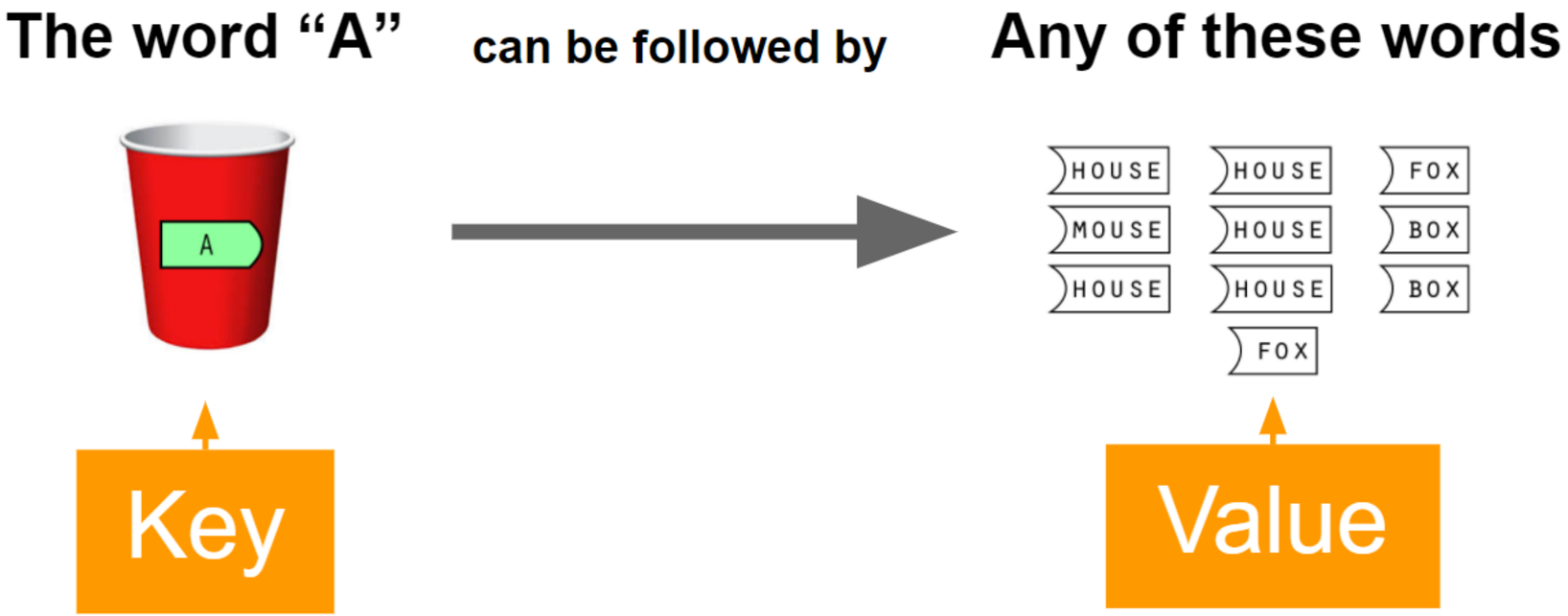
- Let’s create the phonebook

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

- Let’s get Alex’s number from the phonebook

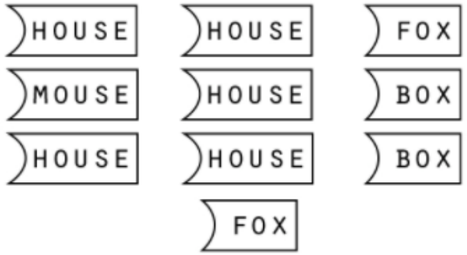
```
>>> phone_book["Alex"]
111
```

Revisiting the Markov Chain Example



Revisiting the Markov Chain Example

The word “A” **can be followed by** **Any of these words**

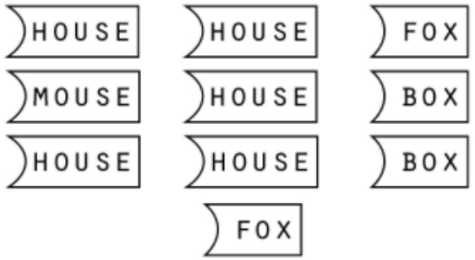


We can store the slips of paper as a python list!

```
['house', 'mouse', 'house',
'mouse', 'box', 'fox', 'box',
'fox', 'house', 'mouse']
```

Revisiting the Markov Chain Example

The word “A” **can be followed by** **Any of these words**

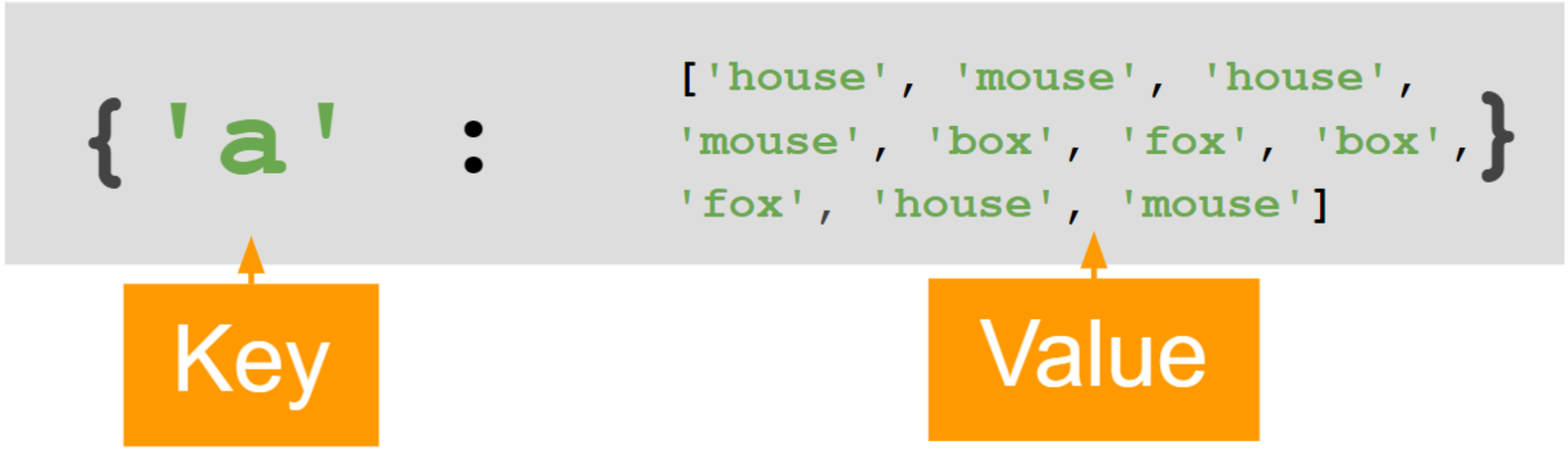


We want to look up the word “a” and get back the list!

```
{'a' : ['house', 'mouse', 'house',
'mouse', 'box', 'fox', 'box',
'fox', 'house', 'mouse']}
```

Dictionaries!

So we get a Dictionary with a List value!



If you look up “A” you get back a list of all the words that can follow “a”

Entire Dictionary

```
cups = {'am': ['Sam', 'That'],
        'In': ['a', 'a', 'a'],
        'a' : ['house', 'mouse',
               'house', 'mouse',
               'box', 'fox', 'box',
               'fox', 'house',
               'Mouse']
        ....}
```

Coding Time!

You now know all about for loops!

Try attempting **Part 6**

The tutors will be around to help!

Thank You!

Other Resources:

- Codecademy
- Grok
- Trinket
- University Subjects