TEACHER'S HANDBOOK

Lesson Plans, Guides & Support

3-B

DECISION MAKING

Using Python

CONTENTS PAGE

BREAKDOWN OF LESSON PACK

Learners who have started to learn about Python using Marty the Robot will be introduced to different programming constructs to create small chatbots that can be used in conjunction with Marty. Students will be introduced to approaches to getting user input, using if statements and logical operators to make decisions and manipulating strings to analyse user input.

Core Concepts

- ✓ Sequences
- ✓ Debugging
- ✓ Algorithms
- ✓ Application of Engineering
- ✓ Variables
- ✓ Conditions
- ✓ Operators

Core Skills

- ✓ Collaboration
- ✓ Computational Thinking
- ✓ Problem Solving
- ✓ Communication
- ✓ Digital Literacy
- ✓ Creative Thinking
- ✓ Critical Thinking

3.5 GETTING USER INPUT

Students will be continuing to explore different movements that Marty can do by creating a simple chatbot that can be used alongside Marty. Students will learn to gather input from the user by asking questions.

3.6 IF STATEMENTS

Using if statements, students will be able to analyse what information the user is entering into the program and coding Marty and their chatbot to respond accordingly.

3.7 VALIDATING USER INPUT

Continuing to extend the chatbot, students will begin to analyse and validate the information that the user enters into the program to ensure that any values entered are sensible and won't cause problems when used with Marty.

3.8 PROCESSING COMPLEX USER INPUT

Instead of asking multiple questions to gather values for one Marty movement, they will be asking the user to enter all values in one go and using string manipulation techniques in Python they will extract the required values.

DEVELOPING DIGITAL LEADERS

Our students are living in a digital world. We introduce core concepts throughout our lessons such as algorithms and if statements but we want to emphasise the importance of developing digital leaders who are engaged and can thrive in a connected, digital world. This is why in each of our lesson packs, we will refer to the *ISTE Standards for Students** to highlight how our lessons help guide the way.

Empowered Learner	
1-a Set Learning Goals	✓
1-b Customise Learning	✓
1-c Seek Feedback	✓
1-d Explore Technology	✓

Digital Citizen	
2-a Digital Identity	
2-b Online Behaviours	✓
2-c Respect & Sharing	
2-d Personal Data	

Knowledge Constructor	
3-a Effective Research	✓
3-b Evaluate Resources	
3-c Curate Information	✓
3-d Explore Real Issues	✓

Innovative Designer	
4-a Use a Design Process	✓
4-b Select Digital Tools	
4-c Prototype Iteration	✓
4-d Problem Solving	✓

Computational Thinker	
5-a Formulate Problems	✓
5-b Data Analysis	
5-c Decomposition	✓
5-d Algorithms	✓

Creative Communicator	
6-a Tool Selection	
6-b Original vs Remix	✓
6-c Communicate Ideas	✓
6-d Presentation	✓

Global Collaborator	
7-a Diverse Teams	✓
7-b Collaborative Tech	✓
7-c Giving Feedback	✓
7-d Investigate issues	✓

HOW TO USE THIS HANDBOOK

We have created this handbook so that you have everything you need to deliver our lesson packs. For each lesson you will have the following information,

LESSON PLANS

Outcomes, Resources & Learning Plans

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

TEACHER GUIDES

Prompts & Questions for Delivery of Lessons

POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

SOLUTIONS

Sample Solutions & Activity Guides

GETTING USER INPUT

Outcomes, Resources & Learning Plans

EDUCATION LEVEL: Third/Fourth Level (Ages 11-15) **PRE-REQUISITES:** 3.1-3.4
LESSON DURATION: 45 minutes **DEVICE COMPATIBILITY:** Laptop or PC
CROSS-CURRICULAR LINKS: Technologies/Literacy/Numeracy

LESSON OVERVIEW

After an introduction to Marty using the Python library *martypy*, students will be asking users for input to create a chatbot interface for Marty and have control of movements. To do this, they will need to use variables to store the replies from users and use this input to alter how Marty moves.

LEARNING OBJECTIVES

- Explore chatbots and describe different use cases
- Describe what variables are and how they are used
- Understand that values stored in a program may be of different types
- Use sequence and variables to design and create a chatbot program in Python

KEY VOCABULARY

- Python
- Chatbot
- Variables
- String
- Number/Integer/Float
- User Input

RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 1)
- Devices with a Python editor and *martypy* installed

ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

LEARNING PLAN & ACTIVITIES

1. Discussion of what a chatbot is with students, exploring different chatbots available online
   A. What are some of the different use cases for chatbots?
   B. What features make a chatbot feel realistic when you're chatting to it?
   C. What ethics would you have to consider when designing and creating a chatbot?
2. Demonstrate example script for Marty chatbot to show students what they will be working on today
3. Ask students what they think they will need to create this kind of program? What programming constructs and concepts? Students will need to gather user input and use variables to store and explore the input
4. Discussion with students about variables
   A. Have they seen or used variables before? What is the definition of a variable?
5. Showing students a basic example script (in PPT slides) ask them to highlight where they think the variables are
6. Students begin to design and code their simple chatbot, asking the user questions and using Marty movements to compliment the chatbot responses
7. Ask students what *type* of values/data they have been gathering from the user so far?
   A. Values/data that is made up of text are called *strings*
   B. Highlight that Python assumes each input from the user will be in *string* form so they will need to change it to be a number or integer in order to process a number (for example, if asking the user how many steps Marty should take)
8. Challenge students to ask the user for values/data that they can change into integers and include that in their chatbot programs
9. Students should complete the end of lesson reflection section of their workbooks

GETTING USER INPUT

Outcomes, Resources & Learning Plans

EDUCATION LEVEL: Third/Fourth Level (Ages 11-15)
PRE-REQUISITES: 3.1-3.4
LESSON DURATION: 45 minutes
DEVICE COMPATIBILITY: Laptop or PC
CROSS-CURRICULAR LINKS: Technologies/Literacy/Sciences

EXTENSIONS & CHALLENGES

- Ask other student groups to test out the chatbot and give feedback on how they could make it better or more realistic (*Literacy/Technologies*)
- Explore the history of chatbots further by researching artificial intelligence and studies including The Turing Test and Chinese Room debating whether computers can think and create a short video/presentation/animation (*Literacy/Technologies/Social Studies*)
- Create a poster with instructions and tips on creating a realistic chatbot (*Arts/Literacy/Technologies*)

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.5
Digital Literacy	TCH 0-01a	●
Technological Developments in Society and Business	TCH 0-05a	●
	TCH 1-05a	●
	TCH 2-05a	○
	TCH 3-05a	○
Craft, Design, Engineering and Graphics	TCH 0-09a	○
	TCH 0-12a	●
	TCH 1-12a	●
	TCH 2-12a	●
	TCH 3-12a	○
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	●
	TCH 3-13a	○
	TCH 3-13b	○
	TCH 0-14a	●
	TCH 2-14a	●
	TCH 3-14a	○
	TCH 4-14a	○
	TCH 1-14b	●
	TCH 0-14b	●
	TCH 2-14b	○
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	●

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

National Curriculum - Computing, Design & Technology

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.5
Computing	1-a	●
	1-b	●
	1-c	●
	1-e	●
	2-a	●
	2-b	○
	2-c	●
	2-f	○
	3-a	●
	4-a	○
	4-b	○
	4-c	○
Design & Technology	1.3-a	●
	2.3-a	○
	2.3-b	●
	3.3-b	●

Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.5
Digital Technologies	ACTDIK001	●
	ACTDIK002	●
	ACTDIP004	●
	ACTDIK008	●

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Australian F-10 Curriculum Continued...

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Digital Technologies	ACTDIP010	●
	ACTDIP012	○
	ACTDIP016	○
	ACTDIP019	○
	ACTDIP029	●
	ACTDIP030	○
Design & Technologies	ACTDEP008	●
	ACTDEP009	●
	ACTDEP015	○
	ACTDEP018	●

CSTA K-12 - Computer Science

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.5
Computing Systems	1A-CS-01	○
	1A-CS-02	●
	1B-CS-01	●
	1B-CS-02	●
	2-CS-02	○
Algorithms & Programming	1A-AP-08	○
	1A-AP-09	○
	1A-AP-11	●
	1A-AP-12	○
	1A-AP-15	○

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

CSTA K-12 - Computer Science Continued...

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.5
Algorithms & Programming	1B-AP-09	●
	1B-AP-11	●
	1B-AP-12	●
	1B-AP-13	○
	1B-AP-15	●
	1B-AP-17	○
	2-AP-11	●
	2-AP-13	○
	2-AP-16	○

TEACHER GUIDES

Prompts & Questions for Delivery of Lessons

GETTING USER INPUT

WHAT IS A CHATBOT?

In this lesson pack, students will be creating a Marty chatbot. It is important that they research and use some of the chatbots that are available to use online. Here are some online examples that you may wish to ask students to look at and evaluate,

https://www.cleverbot.com/
https://www.eviebot.com/en/

- How realistic is this chatbot?
- What different scenarios could you use a chatbot in?
- What would you change about the chatbots that you have used? What would make them better?

VARIABLES

One of the main features of a chatbot is getting user input. We will be doing this in our programs in Python but to store and remember what the user has typed in, we will need to use a variable. Students should recall using a variable in previous classes or even whilst using Scratch.

- Has anyone used a variable before? What did you use it for?
- What are variables useful for?
- What kind of things can we store in our variables for our chatbot programs?

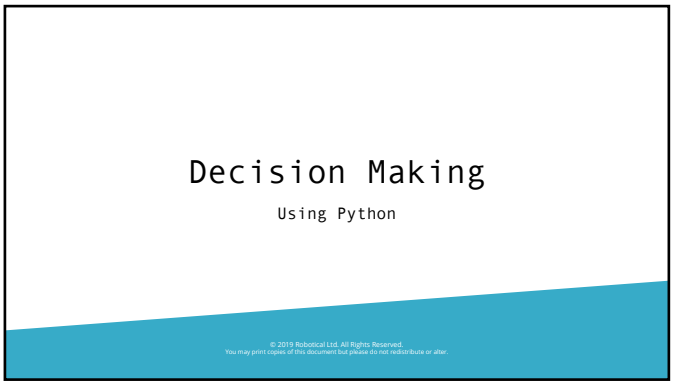
DATA TYPES IN PYTHON

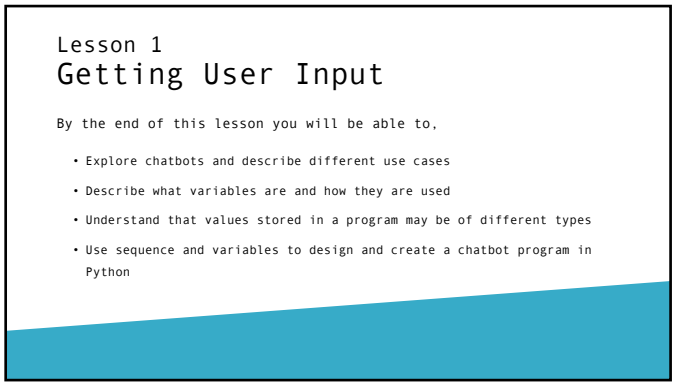
Whilst getting user input in our programs, students will realise that Python will automatically assume that any input will be in *string* format, that is text instead of numbers. To use the input as a value for a Marty movement, they will need to *convert* it into a number data type such as an *integer*.

- Has anyone experienced using different data types in Python before?
- Why do we need different data types whilst coding? What does it tell the computer?
- What type of data do we need to tell Marty the number of steps to walk?

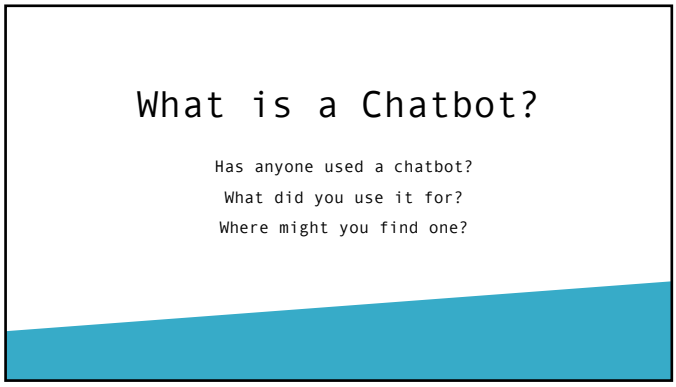
POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

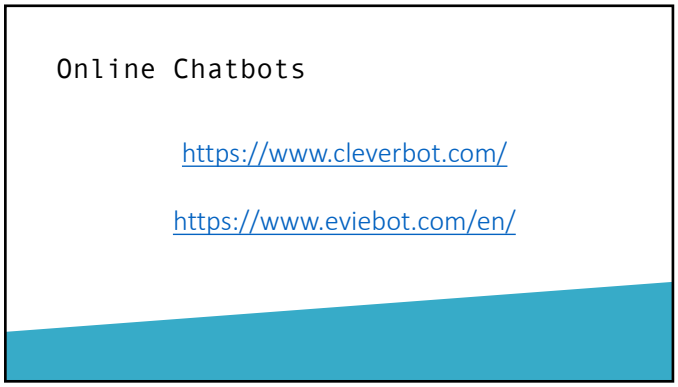




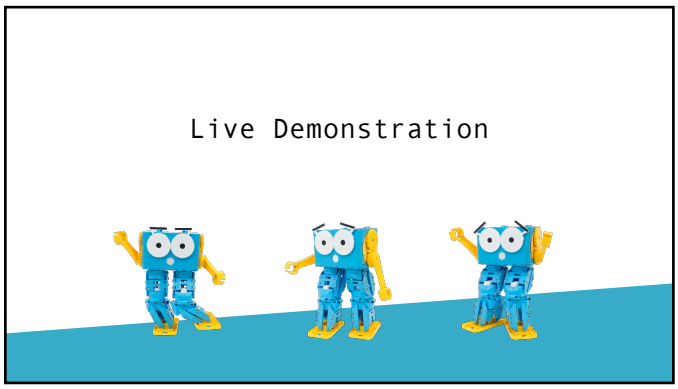
Introduction to lesson outcomes



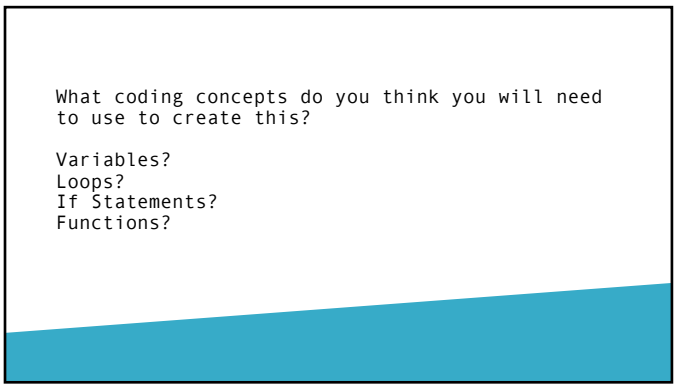
Class discussion of what a chatbot is, exploring that some students might have used one in the past and considering where



Spend some time trying out the example chatbots online to see how human-like they are



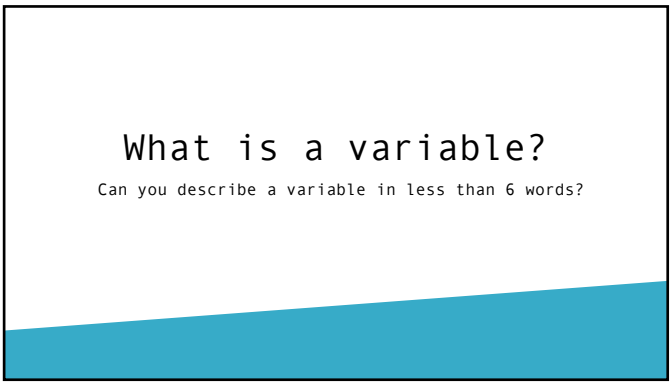
Demonstrate what students will be creating today



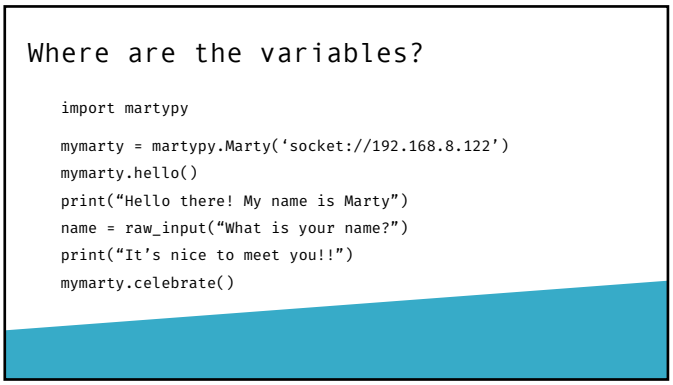
What coding concepts do students think they will need to get the first draft of a chatbot?

POWERPOINT NOTES

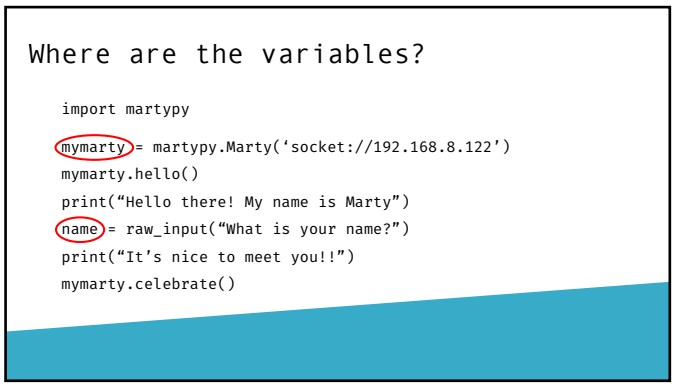
Slide Notes to Deliver our PowerPoints



Can anyone describe what a variable is in less than 6 words?

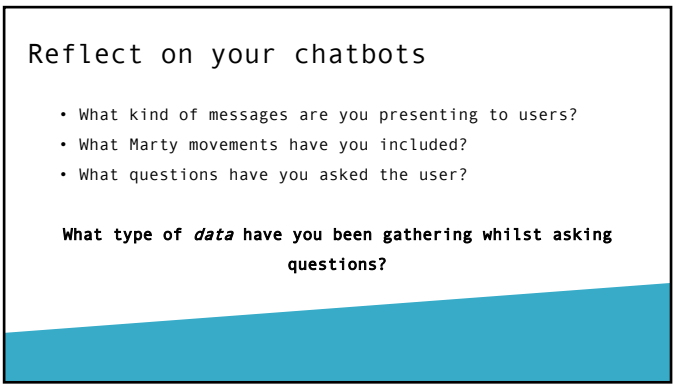


Can students highlight where the variables are in the above example script?

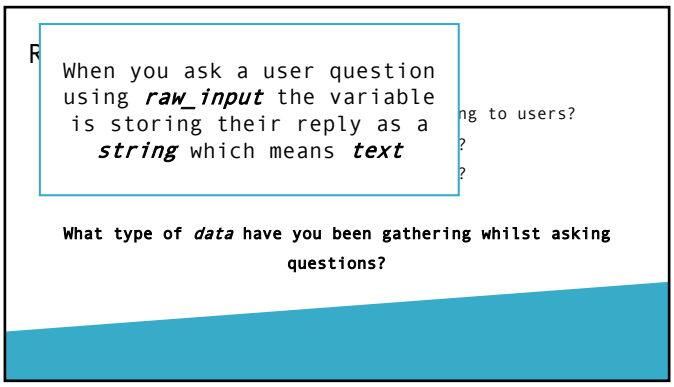


Variables are highlighted but what do they have in common? Hint: they are followed by an = sign

Start coding your chatbot



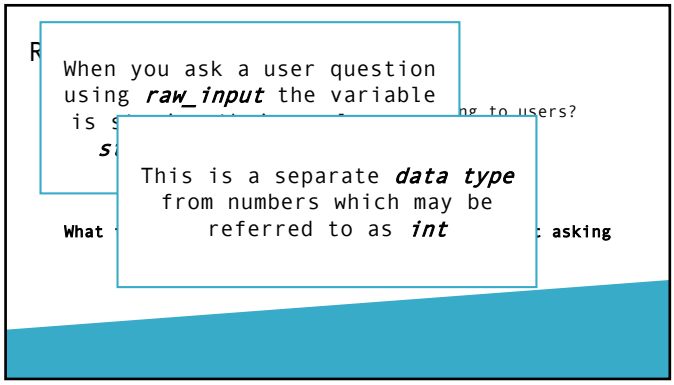
Reflection on what students have created so far - how would they like to improve it so far?



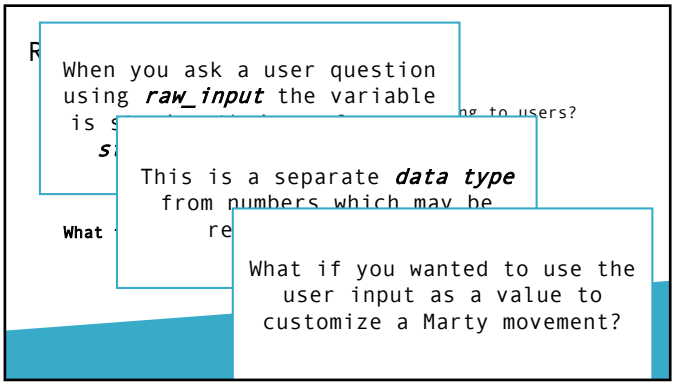
The input from the *raw_input* questions are stored as variables as a string which is text

POWERPOINT NOTES

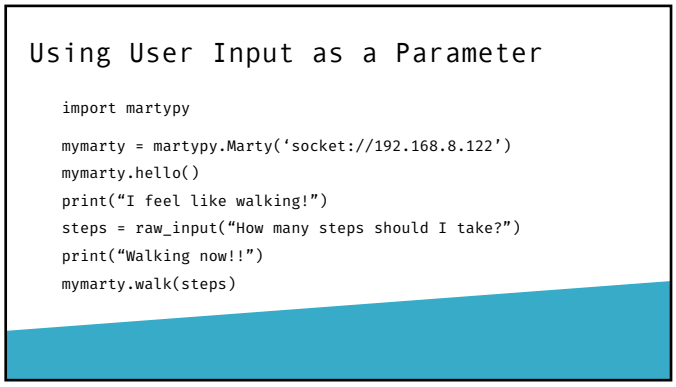
Slide Notes to Deliver our PowerPoints



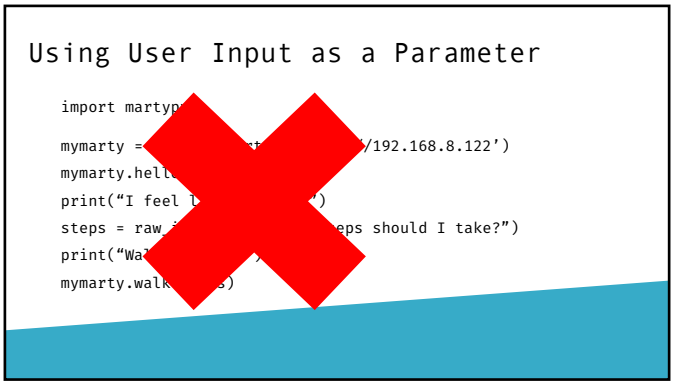
This is different from numbers which is referred to as an int in coding

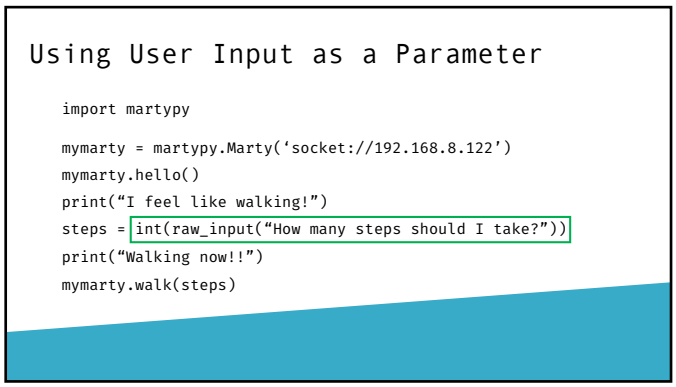


What could we do if we wanted to use this value to customise a Marty movement? Like decide how many steps Marty should walk?

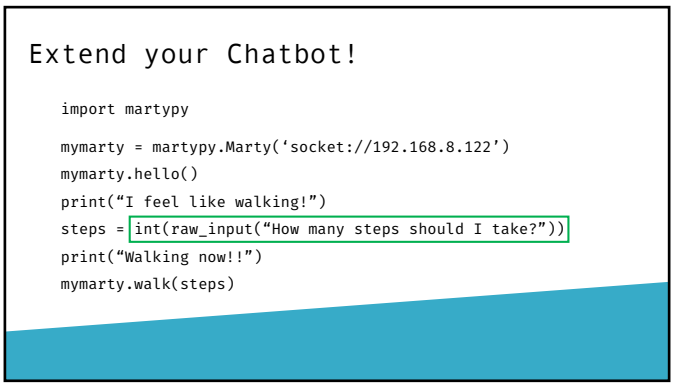


Right now this would get an error message





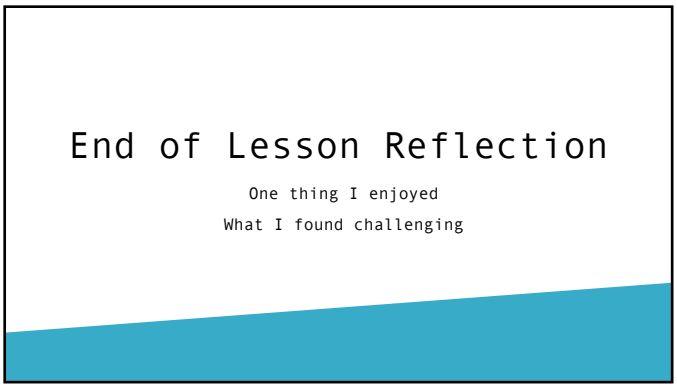
If we make a small change by wrapping the input around an *int* command then this will change the input from text into a number



Extend the chatbot to include questions that take an input and change it into a number

POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints



Complete reflections in student workbooks

SOLUTIONS

Sample Solutions & Activity Guides

1 Write down your own definition of a *chatbot*

There are no specific right or wrong answers here as long as the students understand that chatbots are usually found online or on a computer/device that uses technology to appear like they are speaking to another human.

It is important that students understand that it is not another human at the other end of the chatbot, but it is being run solely by the computer. Once they have had a chance to explore some of the online examples they will see the flaws with some chatbots not feeling realistic.

2 Plan out your own chatbot

Students should use this space to plan out what they want their chatbot thinking about what the chatbot will say and what movements Marty will do to complement them. They should consider the timing of the print messages as well as the movements to complete so that they are in time.

Any challenges that they come across during development should be noted so that they can easily reflect back to these later on. It is good for students to see when they have overcome challenges, even something that might seem like a small thing like connecting to Marty.

3 Ask your classmates for feedback on your chatbot

During the lesson, students will be asked to test out other chatbots created by classmates and leave feedback including aspects that they liked and areas that they think the chatbot could be improved and how.

It is important that students leave critical feedback but are not mean in what they are writing. You might want to incorporate this by asking students to leave feedback in the form of two stars (two things that was done well) and one wish (something to work on in the future).

IF STATEMENTS

Outcomes, Resources & Learning Plans

EDUCATION LEVEL: Third/Fourth Level (Ages 11-15)
LESSON DURATION: 45 minutes
CROSS-CURRICULAR LINKS: Technologies/Literacy

PRE-REQUISITES: 3.1-3.5
DEVICE COMPATIBILITY: Laptop or PC

LESSON OVERVIEW

Students have started to create their chatbots using variables and understand the difference between *strings* and *integers*. In this lesson, we will introduce *if statements* to students to incorporate into the chatbots so that they can begin to make decisions their programs based on the user input to create a more responsive chatbot for Marty.

LEARNING OBJECTIVES

- Describe when you might use an if statement
- Use sequence, variables and if statements to extend the chatbot to make decisions based on the user input
- Explore different approaches to responding to user input through text and Marty movements

KEY VOCABULARY

- Decision Making
- Chatbot
- Variables
- If Statements
- Control
- User Input

RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 2)
- Devices with a Python editor and *martypy* installed

ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

LEARNING PLAN & ACTIVITIES

1. Recap on what the students did in the last lesson
   A. What did they like about the chatbots? What did they want to try and improve about their chatbots?
   B. What programming constructs and concepts were used?
2. Discuss with students in more detail what might make the chatbots better, for example, being able to respond to different keywords that the user has entered
3. Demonstrate example script to show students what they will be working on in the lesson
4. Explain that they will need to use *if statements* to check what the user has entered
   A. What are if statements? Have the students used them before? What for?
   B. Discuss examples of if statements in real life with students using examples from the workbook
5. Students continue work on their chatbots, incorporating if statements and regularly testing with other groups
6. Students should complete the end of lesson reflection section of their workbooks

EXTENSIONS & CHALLENGES

- Ask students to list ways to make their chatbots feel more realistic *(Technologies/Literacy)*
- Can students think of a way to accept different synonyms for *walk* so that their chatbot can respond to a variety of words? *They might want to think about lists in Python (Technologies/Literacy)*

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.6
Digital Literacy	TCH 0-01a	●
Technological Developments in Society and Business	TCH 0-05a	●
	TCH 1-05a	○
	TCH 2-05a	○
Craft, Design, Engineering and Graphics	TCH 0-11a	○
	TCH 0-12a	○
	TCH 1-12a	○
	TCH 2-12a	○
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	●
	TCH 3-13a	○
	TCH 0-14a	●
	TCH 1-14a	●
	TCH 2-14a	●
	TCH 3-14a	○
	TCH 0-14b	●
	TCH 1-14b	●
	TCH 2-14b	○
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	●
	TCH 3-15a	○
	TCH 4-15a	○

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

National Curriculum - Computing, Design & Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.6
Computing	1-a	●
	1-b	●
	1-c	●
	1-e	●
	2-a	●
	2-b	○
	2-c	●
	3-a	○
	3-c	○
	3-g	○
	4-a	●
	4-b	●
Design and Technology	1.3-a	○
	1.3-b	●
	2.3-a	○
	2.3-b	●
	3.3-c	●

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.6
Digital Technologies	ACTDIK001	●
	ACTDIK002	●
	ACTDIP004	●
	ACTDIK008	○
	ACTDIP010	●
	ACTDIP013	●
	ACTDIP019	○
	ACTDIP030	○
	ACTDIP040	○
Design & Technologies	ACTDEK001	○
	ACTDEP005	●
	ACTDEP008	○
	ACTDEP009	●
	ACTDEP015	○

CSTA K-12 - Computer Science

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.6
Computing Systems	1A-CS-01	○
	1A-CS-02	●
	1A-CS-03	○
	1B-CS-02	○
	2-CS-02	○

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

CSTA K-12 Continued...

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Algorithms & Programming	1A-AP-08	●
	1A-AP-09	●
	1A-AP-10	○
	1A-AP-11	●
	1A-AP-12	●
	1A-AP-13	○
	1A-AP-14	○
	1A-AP-15	●
	1B-AP-09	●
	1B-AP-10	○
	1B-AP-11	●
	1B-AP-12	○
	1B-AP-13	○
	1B-AP-15	●
	2-AP-11	●
	2-AP-16	○

IF STATEMENTS

EVALUATING CHATBOTS

Students have started to develop their own chatbot to accompany Marty. Throughout, they will be receiving feedback from other groups in the class and can use what they researched online to guide the changes and development. They should consider the following,

- What makes a chatbot realistic?
- What would make your chatbot better? Easier to use?
- How could you incorporate the physical movements that Marty can make to enhance the chatbot?

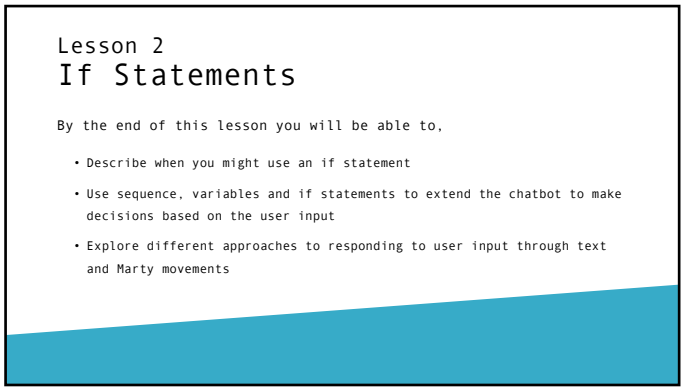
IF STATEMENTS

For their chatbots to make decisions, students will need to incorporate *if statements* into their programs. Students might find it useful to relate if statements to real life decisions that they make where they wait for a condition to become true before doing certain actions.

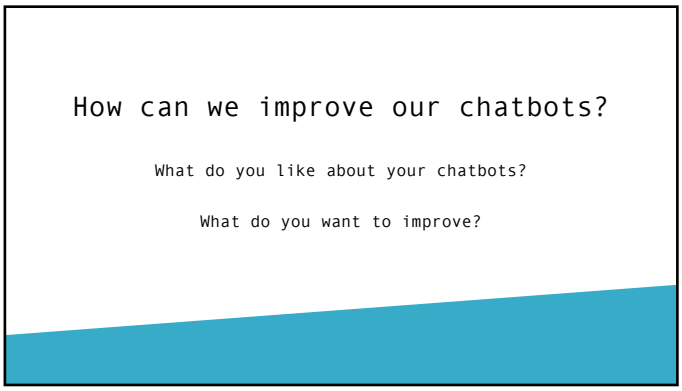
- How do you make decisions? Like deciding when to cross the road?
- What kind of decisions could your chatbot make?
- How would an if statement help make decisions in your chatbot program?

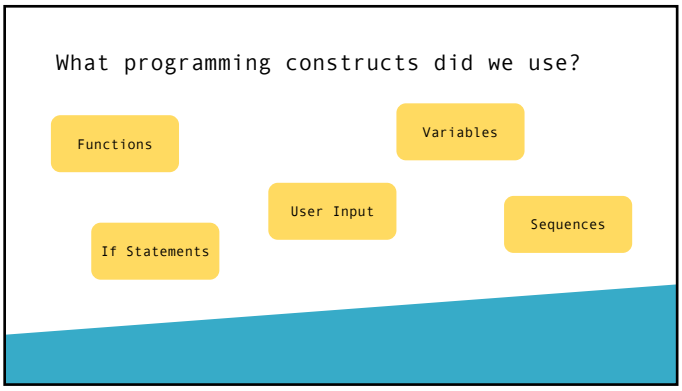
POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

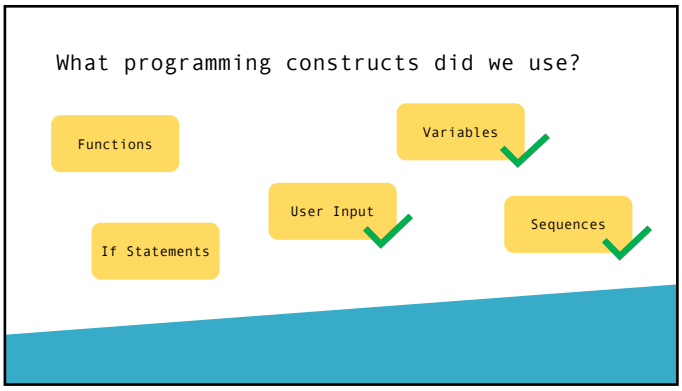


Introduction to lesson outcomes

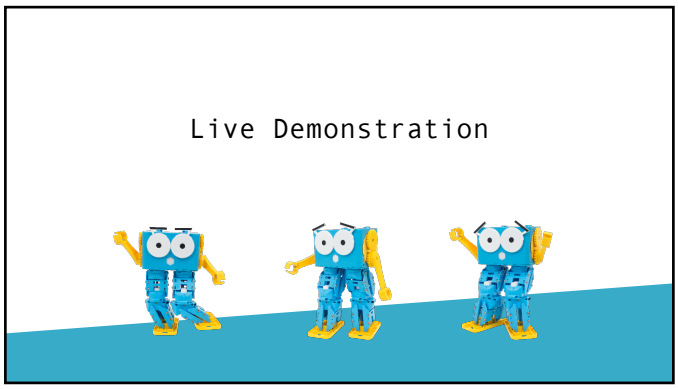




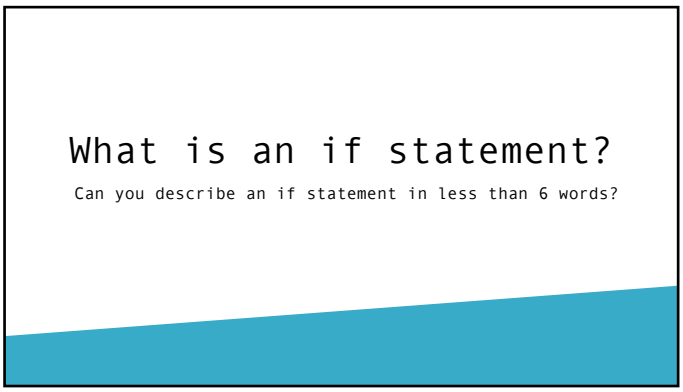
Ask students what programming constructs they think that they have used so far for their chatbots.



They should all agree that they used variables, user input and sequencing to complete their programs in the last lesson.



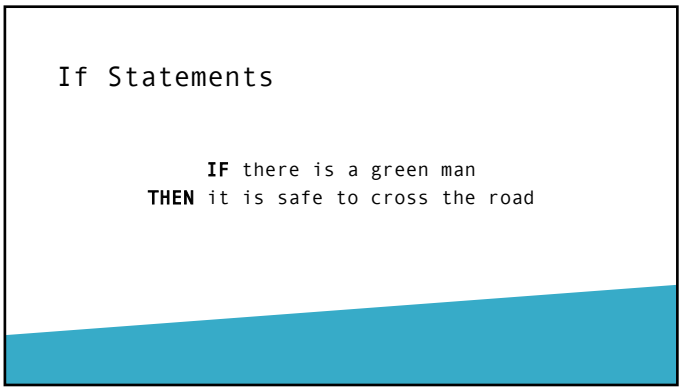
Demonstrate what students will be creating today

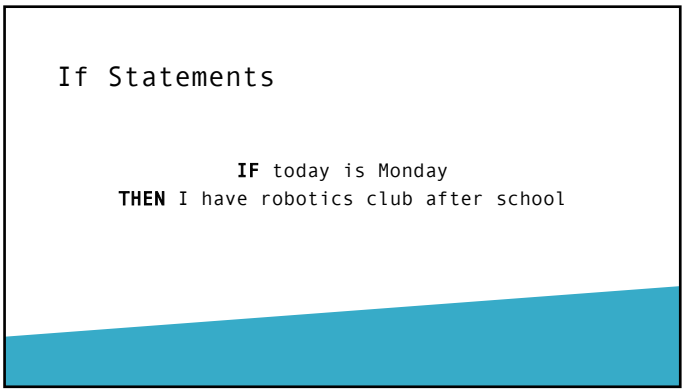


Can anyone describe what an if statement is in less than 6 words

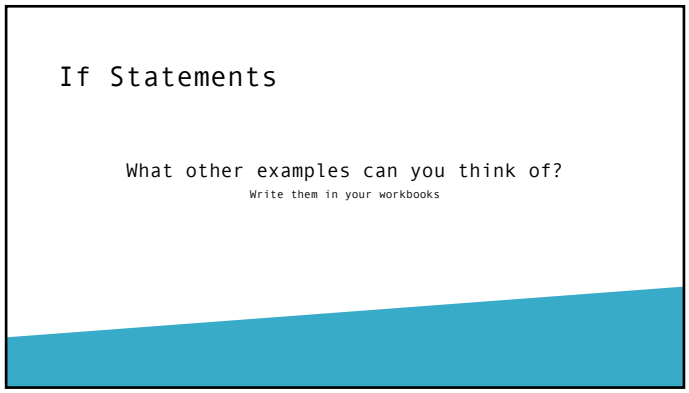
POWERPOINT NOTES

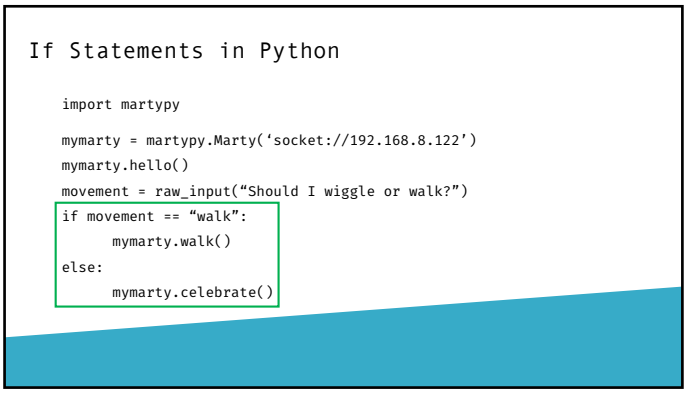
Slide Notes to Deliver our PowerPoints



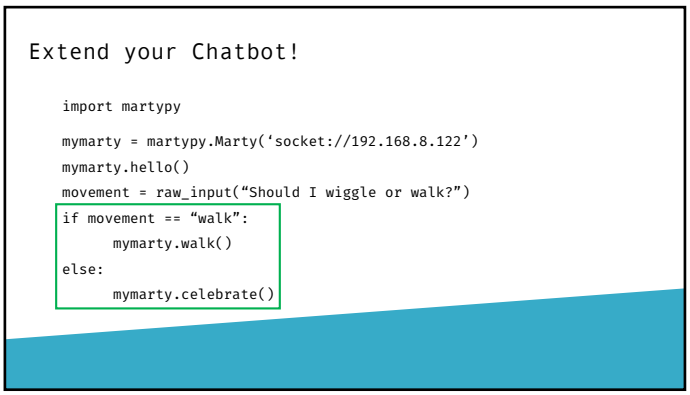


Examples of if statements in real word situations/ scenarios





We ask the user to decide whether Marty should wiggle or walk and use an if statement to decide what to do



Extend your chatbot to include an example like this

Can anyone describe what an if statement is in less than 6 words

SOLUTIONS

Sample Solutions & Activity Guides

1 List ideas to improve your chatbot

Students should reflect on their chatbots by thinking about what they like about their chatbots and how they think they could improve it. They might want to test out or explore their classmates chatbots to give them some ideas on things that they could do to improve their chatbot.

From the previous lesson, students should also have some feedback from their classmates that might give them some inspiration for new ideas.

2 Describe an *if statement* in 6 words

The main concept that students should be exploring in their descriptions is that if statements are used to help make decisions. Using logic to make decisions should be at the core of their answers.

Remember that the students have a word bank at the start of their workbooks that they can look back to at anytime if they are unsure what a word means, as long as they have remembered to fill it out as they go along.

SOLUTIONS

Sample Solutions & Activity Guides

3 Examples of *if statements*

Here are some examples of what students could write. The main thing to look out for is that the students have an IF section and a THEN section.

IF there is a green man **THEN** it is safe to cross the road
IF today is Saturday **THEN** there is football on
IF tomorrow is Friday **THEN** it is almost the weekend
IF I have done the dishes **THEN** I can watch TV

VALIDATING USER INPUT

Outcomes, Resources & Learning Plans

EDUCATION LEVEL: Third/Fourth Level (Ages 11-15) **PRE-REQUISITES:** 3.1-3.6
LESSON DURATION: 45 minutes **DEVICE COMPATIBILITY:** Laptop or PC
CROSS-CURRICULAR LINKS: Technologies/Literacy/Numeracy

LESSON OVERVIEW

After incorporating if statements into their chatbots, students now have programs that are more responsive based on what the user has typed in. We can use this input to decide what actions Marty should do but also to set the parameters for these moves. It's important that we check that the values that have been entered are sensible and we need logical operators to do this.

LEARNING OBJECTIVES

- Describe when you might use logical operators
- Use sequence, variables, if statements and logical operators to create a chatbot that is responsive to and validates the user input
- Explore what real-world systems might incorporate validating user input and why

KEY VOCABULARY

- Decision Making
- Chatbot
- If Statements
- Logical Operators
- Validation
- User Input

RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 3)
- Devices with a Python editor and *martypy* installed

ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

LEARNING PLAN & ACTIVITIES

1. Recap on what the students were working on last time and reflect on their chatbots
   A. What did you add to your chatbot last lesson?
   B. What is your favourite part about your chatbot?
   C. Do you allow the user to set any values for Marty movements? (*number of steps or circle dance timing*)
2. Discussion of why we need to validate user input on different systems - can students think of any examples?
3. Demonstration of how we can do this using logical operators
4. Students should explore logical operators by considering when each should be used in workbook examples
5. Extend the chatbot to ask the user for values to use for Marty movements that get validated using a combination of if statements and logical operators

EXTENSIONS & CHALLENGES

- When the user enters a value that doesn't pass the validation check, let the user know why it has failed and give them another chance to re-enter *(Technologies/Literacy/Numeracy)*
- At the start, could the user type in the name of the Marty they would like to speak to and, using this information along with the *discover* method, connect to that Marty? *(Technologies/Literacy/Numeracy)*

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.7
Digital Literacy	TCH 0-01a	●
	TCH 1-01a	○
	TCH 2-01a	○
Technological Developments in Society and Business	TCH 0-05a	●
	TCH 1-05a	○
Craft, Design, Engineering and Graphics	TCH 0-12a	●
	TCH 1-12a	●
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	○
	TCH 3-13a	●
	TCH 3-13b	○
	TCH 4-13a	●
	TCH 0-14a	●
	TCH 1-14a	●
	TCH 2-14a	●
	TCH 0-14b	●
	TCH 1-14b	●
	TCH 2-14b	○
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	○
	TCH 3-15a	○
	TCH 4-15a	○

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

National Curriculum - Computing, Design & Technology

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.7
Computing	1-a	●
	1-b	●
	1-c	●
	1-e	●
	2-a	●
	2-b	○
	2-c	●
	3-a	●
	3-d	○
	3-g	○
	4-a	●
	4-b	●
Design & Technology	1.3-a	●
	1.3-b	●
	2.3-b	●
	3.3-c	●

Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.7
Digital Technologies	ACTDIK001	●
	ACTDIK002	○
	ACTDIP003	○
	ACTDIP004	●

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Australian F-10 Curriculum Continued...

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.7
Digital Technologies	ACTDIP010	●
	ACTDIP013	○
	ACTDIK014	○
	ACTDIP019	○
	ACTDIP029	○
	ACTDIP030	○
Design & Technologies	ACTDEK001	○
	ACTDEP005	●
	ACTDEP008	○
	ACTDEP009	●

CSTA K-12 - Computer Science

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.7
Computing Systems	1A-CS-01	●
	1A-CS-02	●
	1A-CS-03	●
	1B-CS-01	●
	1B-CS-02	●
	1B-CS-03	○
	2-CS-02	○
	3A-CS-02	○

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

CSTA K-12 - Computer Science Continued...

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.7
Algorithms and Programming	1A-AP-08	●
	1A-AP-09	●
	1A-AP-10	○
	1A-AP-11	●
	1A-AP-12	●
	1A-AP-13	○
	1A-AP-14	○
	1A-AP-15	●
	1B-AP-08	●
	1B-AP-09	●
	1B-AP-10	○
	1B-AP-11	○
	1B-AP-12	○
	1B-AP-13	●
	1B-AP-15	●
	1B-AP-17	○
	2-AP-11	●
	2-AP-12	○

VALIDATING USER INPUT

VALIDATION OF USER INPUT

Throughout this series of lessons, students have been asking the users for input to decide what movement Marty should do and which parameters we should pass in such as speed or number of steps. Students may have started to realise that they need to validate what information is being entered by the user so that Marty can complete the move successfully with the given values.

- Why do you think we need to validate user input? What values might users enter that wouldn't work with Marty?
- What should we do when the value entered isn't accepted/won't work with Marty?
- Can you think of any situations where you've entered information that needs to be checked or validated? *(passwords, checking of age for entry into different places, ...)*

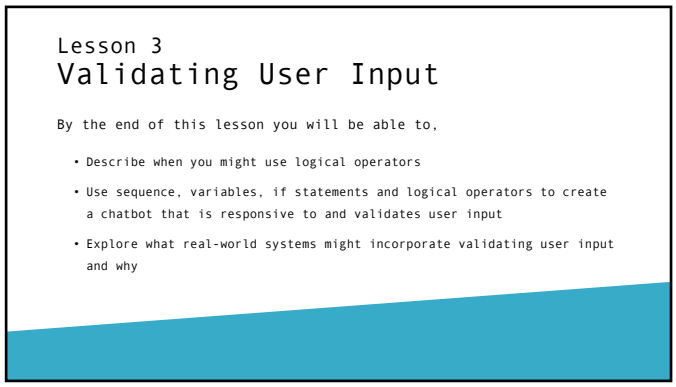
LOGIC OPERATORS

Students will be using logical operators to help them write programs that check the user input. There are three different kind of operators that they will be introduced to and students will need to practice when each should be used.

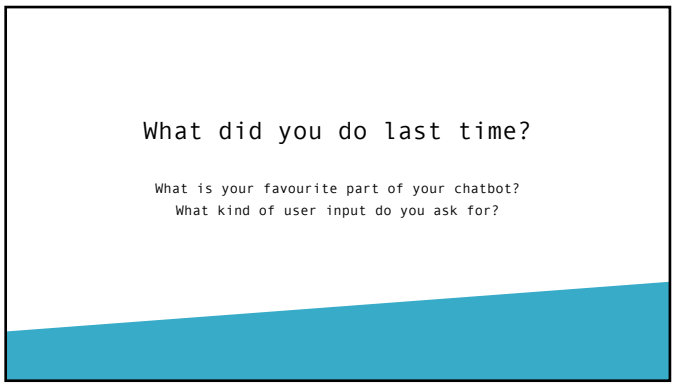
- When would you use AND/OR/NOT operators?
- Why would you use these?
- Do you think you would ever need to use more than one of these at the same time? Can you think of an example?

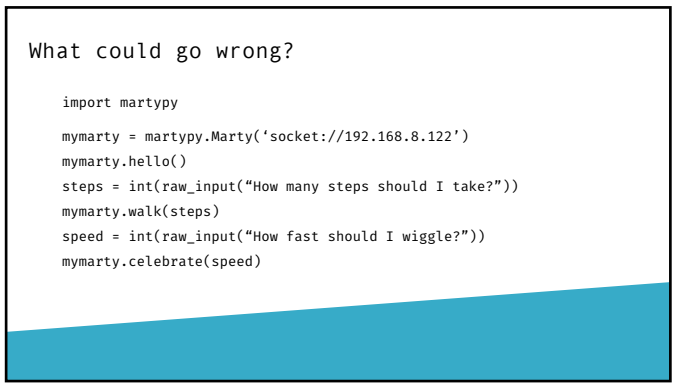
POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

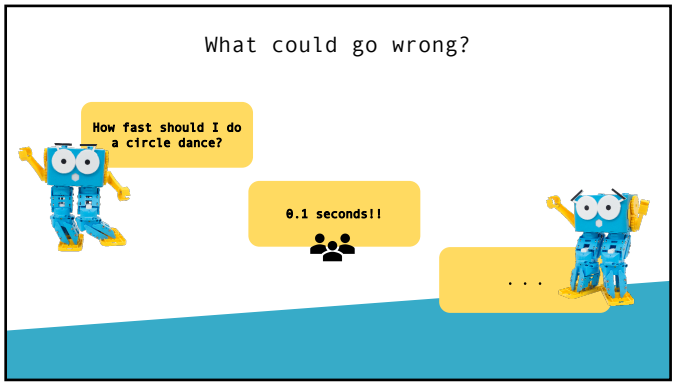


Introduction to lesson outcomes

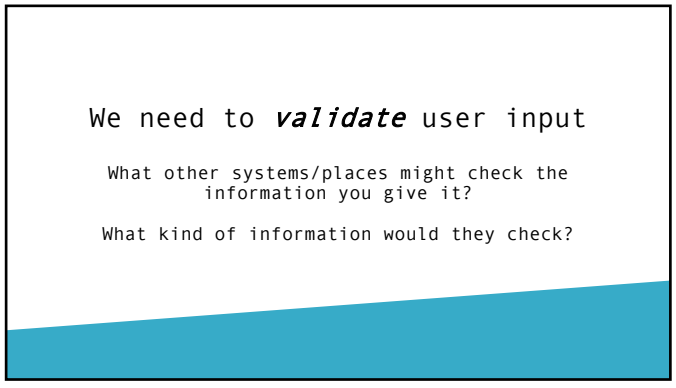




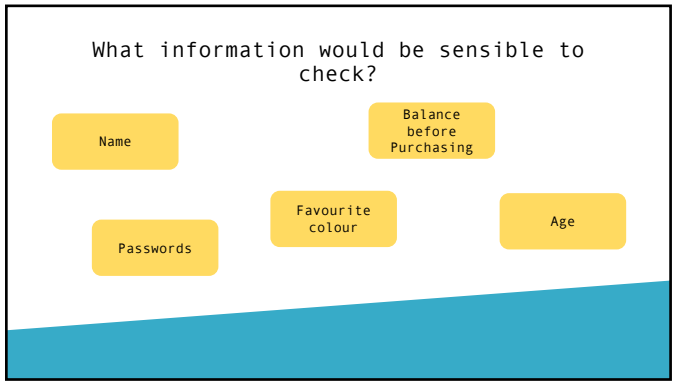
What could go wrong if we ask users to pick the values for Marty's movements like speed or number of steps?



Lets imagine we ask the user how fast we should do a circle dance and they respond with something too fast like 0.1 seconds - this would result in Marty falling!



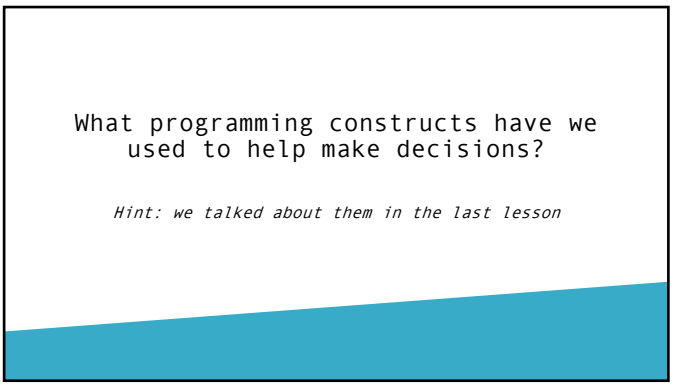
What other places might need to validate information or check it over?



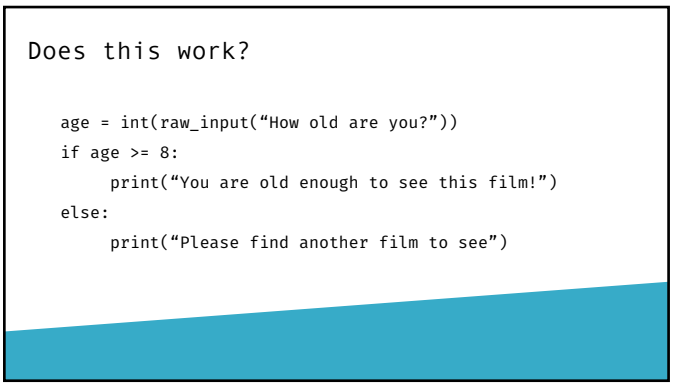
Which pieces of information would be sensible to validate? What other pieces would you check and why?

POWERPOINT NOTES

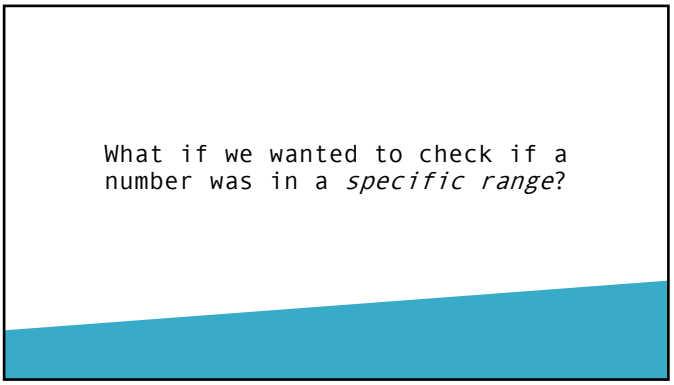
Slide Notes to Deliver our PowerPoints



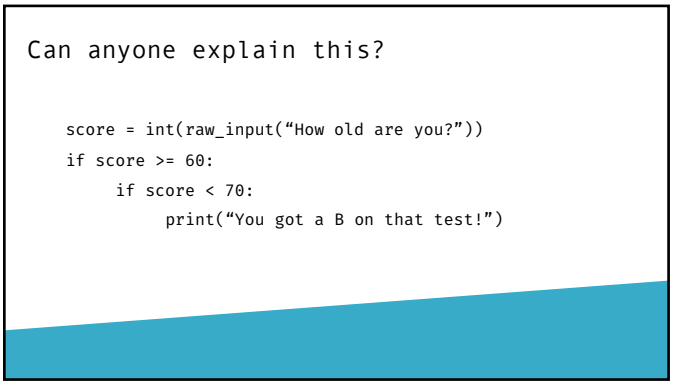
If Statements!!



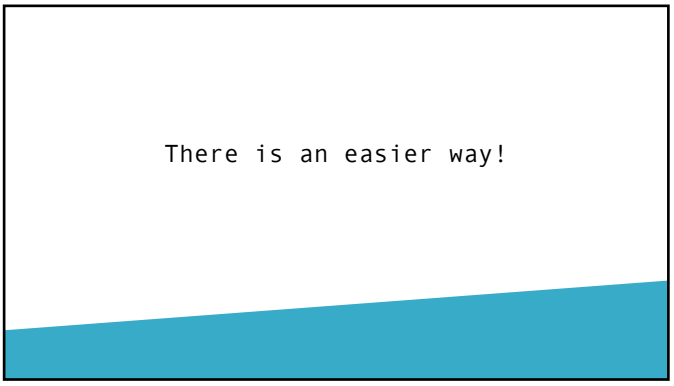
We could use if statements to check if a number is low/ high enough

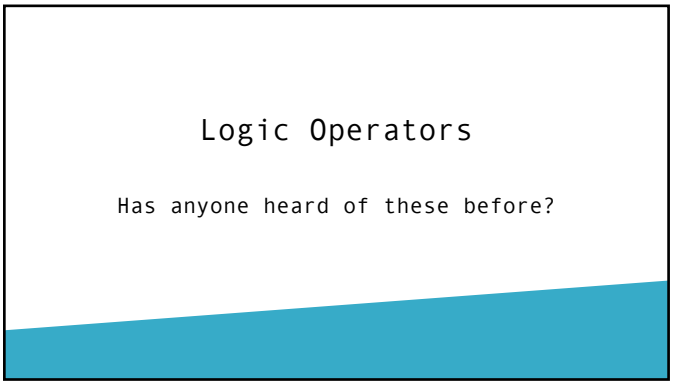


But what if we wanted to check if it was between 2 numbers?



This could get too complicated!

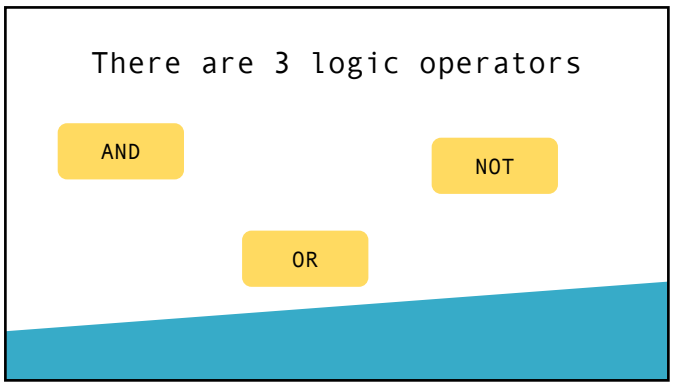




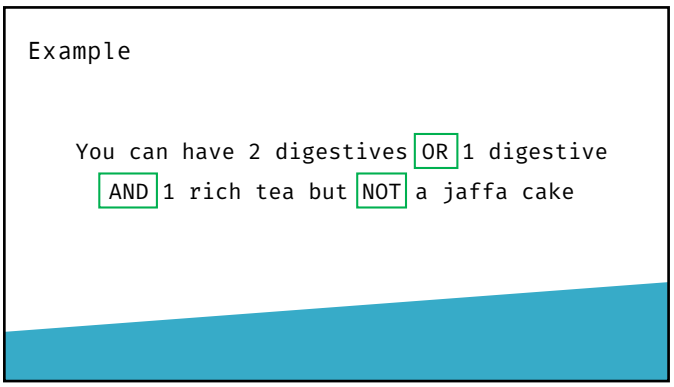
We can use logic operators to make our if statements more complex and easier to write!

POWERPOINT NOTES

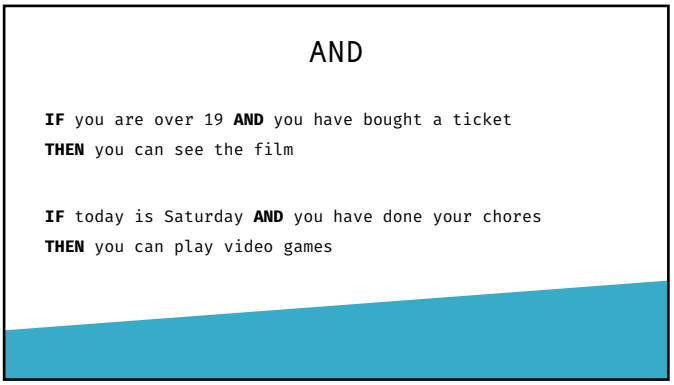
Slide Notes to Deliver our PowerPoints



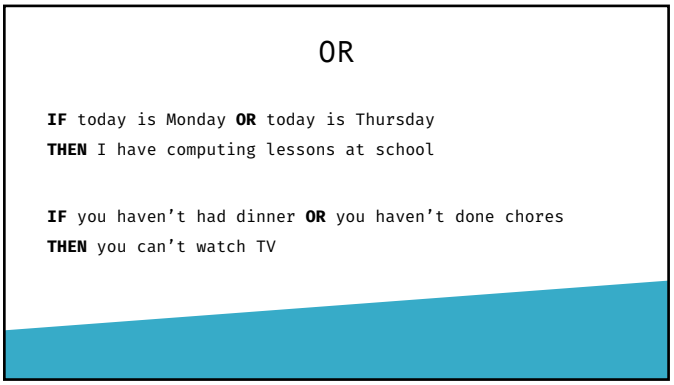
There are three that we will be using - AND, NOT, OR



Here is an example in English using all three logic operators



Examples for AND



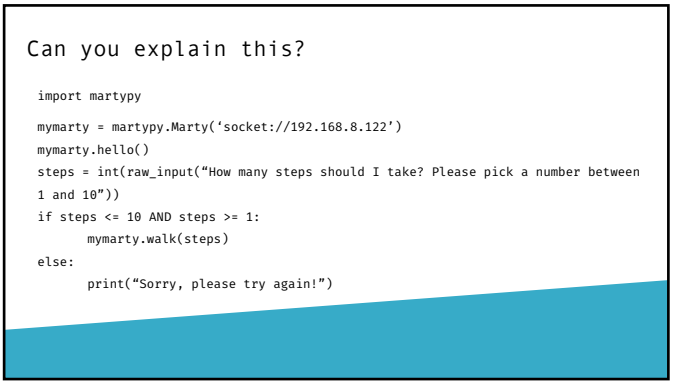
Examples for OR

Examples for NOT

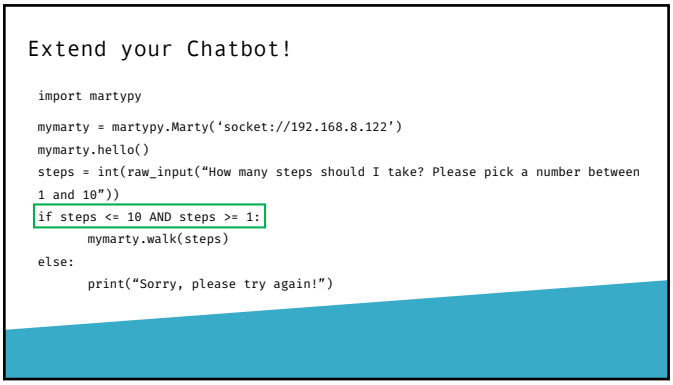
Try the examples in the workbooks

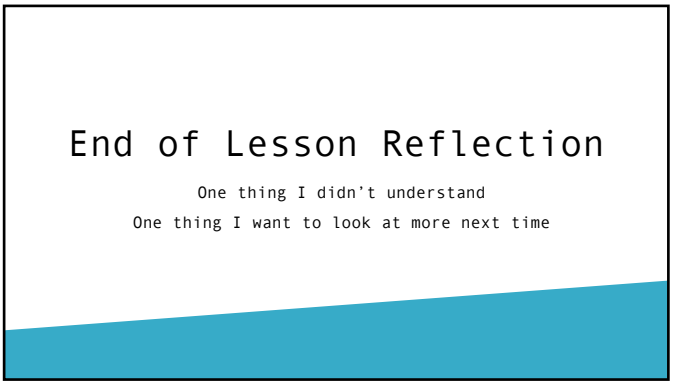
POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints



Who can explain what will happen here? When will Marty walk? What input is needed?





SOLUTIONS

Sample Solutions & Activity Guides

1 Examples of validating user input/data

Please note that students might have some different examples here.

Information Validation Examples

- Checking someone's age at a cinema
- Checking the entered password
- Check bank balance before confirming purchase
- Make sure a text entry is under the limit
- Make sure the entered value is in the correct range
- Checking a PIN number at checkout
- Checking customer details before showing sensitive info
- Checking there are enough seats on the bus before selling a ticket

2 Fill in the blanks

IF **NOT** October

THEN you will not find pumpkins in the shop

IF score >= 60 **AND**

score < 70

THEN the student got a B

IF today is Saturday

OR today is Sunday

THEN it is the weekend

PROCESSING COMPLEX USER INPUT

Outcomes, Resources & Learning Plans

EDUCATION LEVEL: Third/Fourth Level (Ages 11-15) **PRE-REQUISITES:** 3.1-3.7
LESSON DURATION: 45 minutes **DEVICE COMPATIBILITY:** Laptop or PC
CROSS-CURRICULAR LINKS: Technologies/Literacy/Numeracy

LESSON OVERVIEW

So far, students have created a chatbot for Marty that is responsive and validates user input. When asking the user to suggest values for Marty movements, it would be nice if they could give all the values needed for one move in one go. To do this we will need *string manipulation* to pull out the required information.

LEARNING OBJECTIVES

- Describe when you might need to use string manipulation whilst coding
- Use string manipulation of user input to breakdown the text to be used as parameters for Marty movements
- Explore values needed for different Marty moves

KEY VOCABULARY

- Chatbot
- Decision Making
- Control
- User Input
- String Manipulation
- Variable

RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 4)
- Devices with a Python editor and *martypy* installed

ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

LEARNING PLAN & ACTIVITIES

1. Discuss with students what different values/parameters are needed for Marty movements (examples in workbooks)
    1. Reflect on individual chatbots and whether this movement is used
    2. Do students ask the user to customise the values for this movement through user input?
    3. Ask students how they would ask users for values to create the movement
2. Demonstrate that we can ask users for all values in one go that we can then split up into individual values
3. Class to work together to complete the example given in the ppt
4. Continue to work on the chatbot to include this approach and ask users to customise the values for walking

EXTENSIONS & CHALLENGES

- Students should rotate around the different group chatbots and leave feedback for that chatbot including things that they liked and ways they think it could be improved *(Technologies/Literacy)*
- Write a how-to guide for creating a chatbot in Python that other students could follow *(Technologies/Literacy)*

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.8
Digital Literacy	TCH 0-01a	●
	TCH 1-01a	○
	TCH 2-01a	○
Technological Developments in Society and Business	TCH 0-05a	●
Craft, Design, Engineering and Graphics	TCH 0-12a	○
	TCH 1-12a	○
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	●
	TCH 3-13b	●
	TCH 0-14a	●
	TCH 1-14a	●
	TCH 2-14a	●
	TCH 3-14a	○
	TCH 4-14a	○
	TCH 0-14b	●
	TCH 1-14b	●
	TCH 2-14b	○
	TCH 3-14b	○
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	●
	TCH 3-15a	○
	TCH 4-15a	○

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

National Curriculum - Computing, Design & Technology

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.8
Computing	1-a	●
	1-b	●
	1-c	●
	2-a	●
	2-b	○
	2-c	●
	3-a	●
	3-b	○
	3-c	○
	3-d	○
	3-e	●
	4-a	●
	4-b	●
Design and Technology	1.1-a	●
	1.3-b	●
	2.3-b	●
	3.3-c	●

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.8
Digital Technologies	ACTDIK001	●
	ACTDIK002	●
	ACTDIP004	●
	ACTDIK008	○
	ACTDIP010	●
	ACTDIP012	○
	ACTDIP013	●
	ACTDIP016	○
	ACTDIP019	○
	ACTDIP028	○
	ACTDIP029	○
Design and Technologies	ACTDEK001	●
	ACTDEK002	○
	ACTDEP005	●
	ACTDEP008	○
	ACTDEP009	●
	ACTDEP018	●

LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

CSTA K-12 - Computer Science

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.8
Computing Systems	1A-CS-01	●
	1A-CS-02	●
	1A-CS-03	○
	1B-CS-01	●
	1B-CS-02	●
	2-CS-02	○
Data and Analysis	1A-DA-05	●
Algorithms & Programming	1A-AP-08	●
	1A-AP-09	●
	1A-AP-10	○
	1A-AP-11	●
	1A-AP-12	●
	1A-AP-14	○
	1A-AP-15	●
	1B-AP-09	●
	1B-AP-10	○
	1B-AP-13	●
	1B-AP-15	●
	2-AP-11	●
	2-AP-12	○
	3A-AP-14	○

PROCESSING COMPLEX USER INPUT

GATHERING COMPLEX INPUT

Students will be challenged in this lesson to ask the user for all of the values for one Marty move through a single input/string which they will then have to manipulate to pick out different pieces of information.

- How do you think you are going to keep track of the different values that are being entered by the user? What coding concept can be used to remember pieces of information that we assign?

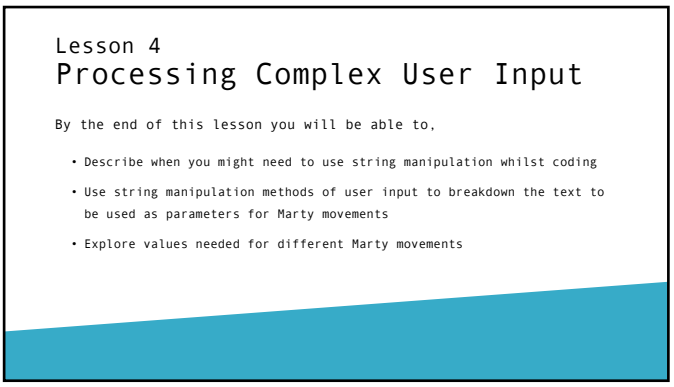
SPLITTING THE USER INPUT BY COMMAS

If the information is entered by the user with each value separated by a comma then that means students can use a built-in function called *split* and we can ask the function to split our long string of values where each comma appears. This will result in a *list* of values, something that students might not have come across yet. Here are some important things to remember,

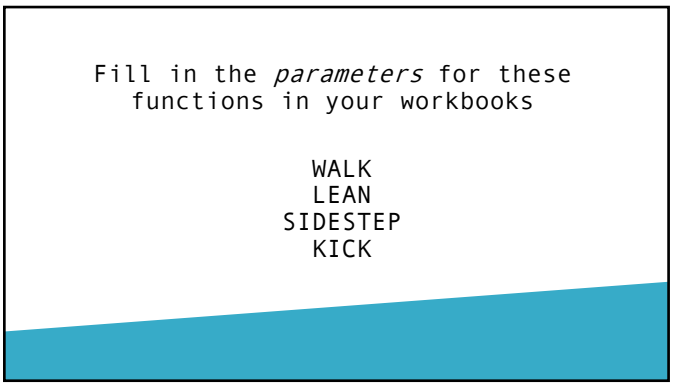
- It is important to ask the user to enter the values separated using a comma because this will make it easier to break into separate values later in our program
- Using the *split* function the values will be broken up and put into a *list* where each value is a different entry
- Lists start at 0 so when students are going to retrieve information from a list they will need to start with 0 to get the very first element in the list
- Students might want to think about *trimming* whitespace off the inputs...

POWERPOINT NOTES

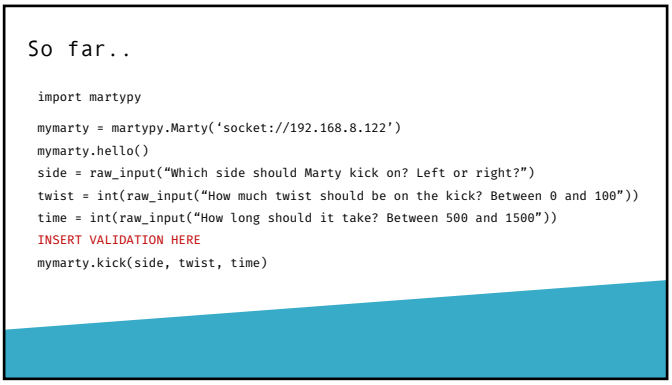
Slide Notes to Deliver our PowerPoints



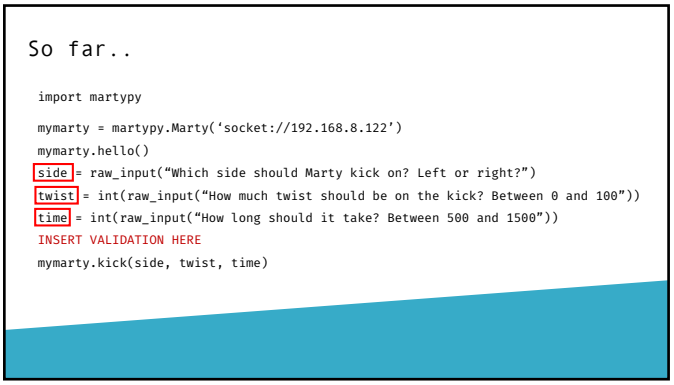
Introduction to lesson outcomes



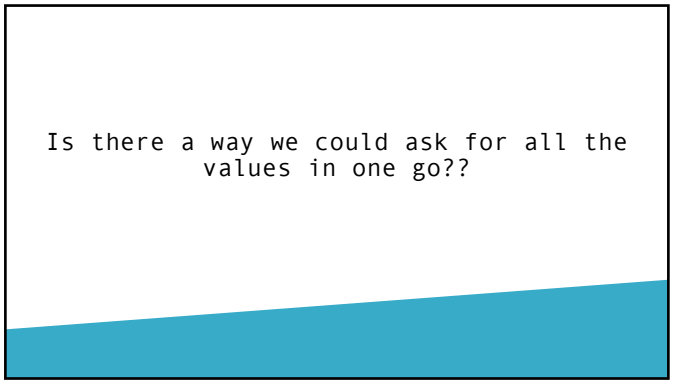
Using MartyPy documentation, can students fill in the blanks for the required parameters for these moves

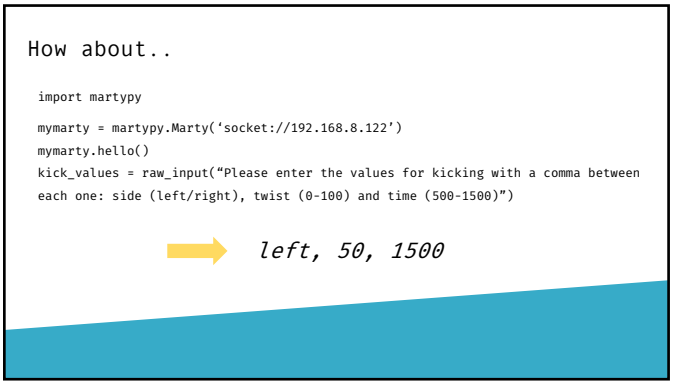


We have to use several variables and input questions to keep track of the required values for a Marty movement



This could get too complicated!

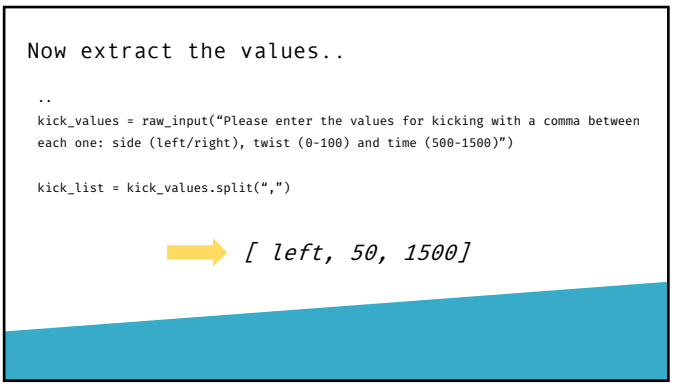




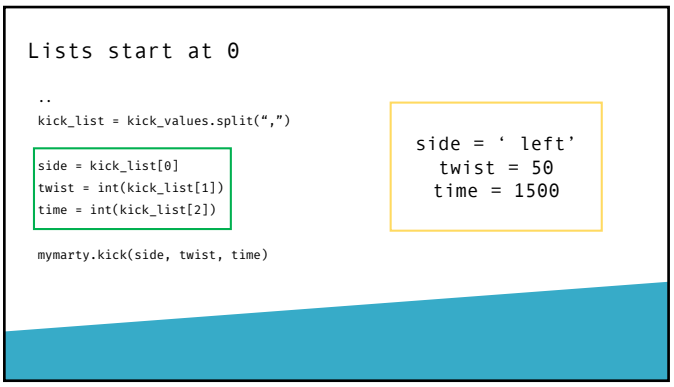
We now have a list of values as a STRING

POWERPOINT NOTES

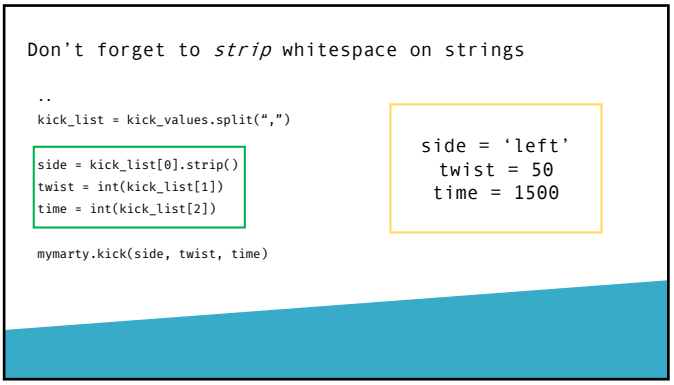
Slide Notes to Deliver our PowerPoints



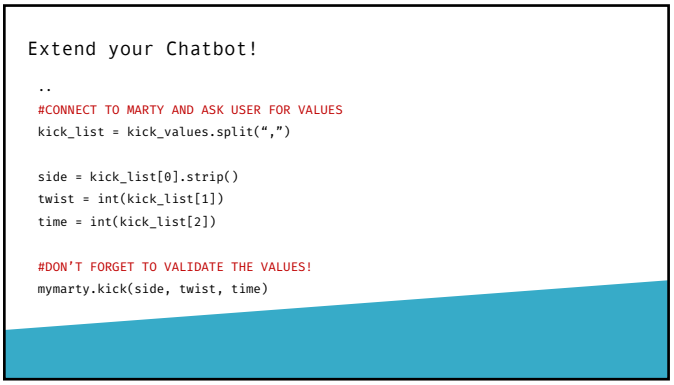
We now have a formal list of STRING entries



We can pick out each value and enter it into a variable, making sure to change number ones into an int first!



If you have a text entry (called a String) then you might want to strip any extra whitespace off the string



Extend your chatbot using this example!

SOLUTIONS

Sample Solutions & Activity Guides

1 Fill in the blanks

marty.walk(	Num Steps	Start Foot	Turn	Step Length	Move Length	**)**
marty.lean(	Direction	Amount	Move Time			**)**
marty.sidestep(	Side	Num Steps	Step Length	Move Time		**)**
marty.kick(	Side	Twist	Move Time			**)**