

DT Challenge Year 5/6 Blockly – Chatbot

Module 1: Getting Started with Code

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/getting-started/>

Previous: None

Next: Data Types: Numbers and Strings

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Representing data as a string
Specification, algorithms, implementation	Simple Algorithms, user input
Digital Systems	
Interactions	Interaction (input/output)
Impact	

Objectives (Content Descriptions)

ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps
ACTDIP020	Implement digital solutions as simple visual programs involving user input.

What are we learning? (Abstract)

Learning to write programs includes getting to know the instructions that a computer can understand. By combining them logically and creatively, we can write a great number of different programs. Therefore, programming is not too dissimilar from building things with Lego. Once we know the parts and how to use them, we can combine them in many different ways.

Module outline

In this module, students are carefully guided in taking their first steps in programming with the Blockly programming language. The module consists of five activities:

In the first activity, students write a simple “Hello, World!” program. Students learn the pattern to insert a string block into a print block to accomplish this.

In the second activity, students are introduced to strings. They learn that strings can contain any kind of information and how to output strings via one or multiple print-statements.

In the third activity, students are introduced to variables, which can be thought of as containers that hold content.

Students learn that the contents of variables can be assigned via a set block.

In the fourth activity, students are introduced to the concept of input via the ask block. Step by step, the concepts of input, variables and output are combined, which enables students to write a variety of little programs that prompt the user for input, store it in a variable and then print it - often in combination with some other text, such as a greeting.

In the fifth and final activity of this module, students are introduced to the concept of more than one variable in a program.

Guiding Question

How do we make a computer do exactly what we want?

How can we input and output information with a computer?

How can we store information in a computer (variables)?

Elements

Representations

Sequencing

User / Environmental Input

Output

Visual programming

Purpose/Hook - The world is changing...



Video:

Computer Science is Changing Everything: <https://www.youtube.com/watch?v=xJqSu1IbcHg>

The video talks about the role computer science plays in a range of industries, and the impact it has had on our modern society. It provides students with some context about why it's important we learn about and understand programming and related concepts, and can serve as a starter for a discussion about how it is computers are able to do the things that they do.

Students should be guided to the conclusion below through a class discussion, starting with how it is computers "learn" what they need to do, and the role computer programmers play in formulating the required algorithms and code.













Discussion:

How do we know what to do in a bunch of situations? How do we know how to tie our shoes in the morning? How do we know how to read a book? Or even which bus to get on at the bus stop? It's because we've been taught by someone older or more knowledgeable than us.

Our job as computer scientists is to “teach” a computer by giving really clear instructions that the computer is very good at following... it never gets on the wrong bus unless we told it to.

Structure of the lesson:

Types of component:			
 Discussion	 Worksheet	 Plugged Activity	
 Group Activity	 Unplugged Activity	 Video	
 Animation	 Reflection	 Game	 App

Lesson Components:

Why do we need to learn programming?

 **Discussion:** As described above

Step 1: Writing your first program

Plugged Activity

In the first activity, students write a simple “Hello, World!” program. Students learn the pattern to insert a string block into a print block to accomplish this.

Start at Module 1, slide 0

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/getting-started/0/>

Step 2: Strings and Printing

Plugged Activity

In this activity, students are introduced to strings. They learn that strings can contain any kind of information and how to output strings via one or more print blocks.

Start at Module 1, slide 4

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/getting-started/4/>

Step 3: Variables

Video: CS Principles: Intro to Variables - Part 1

https://www.youtube.com/watch?v=G41G_PEWfJE



Plugged Activity

In this activity, students are introduced to variables, which can be thought of as containers that hold information. Students learn that the contents of a variable can be set via a set block and retrieved with a get block.

Start at module 1, slide 8

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/getting-started/8/>

Step 4: Reading User Input



Plugged Activity

In this activity, students are introduced to the concept of input via the ask block. Step by step, the concepts of input, variables and output are combined, which enables students to write a variety of small programs that prompt the user for input, store it in a variable and then print it - often in combination with some other text, such as a greeting.

Start at Module 1, slide 13

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/getting-started/13/>

Step 5: Multiple Variables



Plugged Activity

In the final activity of this module, students are introduced to the concept of more than one variable in a program and how to combine them, for example in order to get and print the full name of a student, consisting of first, middle and last name.

Start at Module 1, slide 17

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/getting-started/18/>

Review:

Variables are encountered everywhere



Group Activity

Students share some examples where they think examples similar to the code they've been writing are used (particularly variables). Things might include:

- Have you had your name appear in a high score list in a game?
- Substituting the player's chosen name throughout a game's interface
- Bus destinations or train stations appearing in an electronic transport display
- Usernames appearing on websites

But in all of these cases, the same data isn't always displayed – things can change depending on the situation. We'll learn about that later – *this can be the teaser for future lessons*.

Resources and Links

Printables

Digital Resources

Computer Science is Changing Everything: <https://www.youtube.com/watch?v=xJqSu1IbcHg>

CS Principles: Intro to Variables - Part 1 https://www.youtube.com/watch?v=G41G_PEFWfjE

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 2: Data types: numbers and strings

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/data-types/>

Previous: Getting started with code

Next: Strings: Working with Words

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as integer and string
Specification, algorithms, implementation	Simple Algorithms
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps
ACTDIP020	Implement digital solutions as simple visual programs involving user input.

What are we learning? (Abstract)

Computer programs use different types of data for their operation. Some of them are strings, others are integers, float, or boolean. In this module, students learn to use and manipulate integer numbers through basic arithmetic operations (+, -, *, /).

Module outline

The module consists of three activities: The first activity introduces the concept of numbers (integers) and arithmetic operators. In the second activity, students learn to read user input as numbers, store it into variables and apply arithmetic operators. We discuss why strings need quotes and how strings and numbers are different. In the third activity students learn how to multiply strings by applying a multiplication operator to a string.

Guiding Questions

How do we get the computer to perform simple arithmetics (+, -, *, /)?

How does a computer treat strings and numbers differently?

Elements

Representations

Types of data

Whole numbers represent data

Sequencing

User / Environmental Input

Output

Visual programming

Arithmetic operators (+, -, *, /)

Purpose/Hook - Data Types in Python



Worksheet: Data Types

Enclosed at the end of this document.



Discussion:

How do we perform arithmetics? How do we count, add, subtract, multiply and divide numbers? Is it always easy? What happens when the numbers get large or when we have to process many numbers? Do we sometimes make mistakes when performing arithmetics?

By recognising that arithmetics is hard and especially repeated arithmetics is time-consuming, students identify that some type of helper can take over some of the laborious and repetitive arithmetic tasks.

Structure of the lesson :

Types of component:



Discussion



Worksheet



Plugged Activity



Group Activity



Unplugged Activity



Video



Animation



Reflection



Game



App

Lesson Components:

Step 1: Numbers



Video

The role of computer arithmetics

https://www.youtube.com/watch?v=O5nskjZ_GoI



Plugged Activity

This activity introduces the concept of numbers (integers) and arithmetic operators

Start at Module 2, slide 0

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/data-types/0/>

Step 2: Data Types



Plugged Activity

In this activity, students learn to read user input as numbers, store it into variables and apply arithmetic operators. In Blockly, there are two kinds of ask-blocks. The green ask-block is used for string input, whilst the blue ask-block is used for the input of numbers.

Start at Module 2, slide 2

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/data-types/2/>

Step 3: Multiplying Strings



Plugged Activity

In this activity, students learn how to multiply strings by applying the multiplication operator to a string.

Start at Module 2, slide 7

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/data-types/7/>

Review:



Group Activity

Students share some examples where they think the computer could be used to do arithmetics and string operation for them.

Resources and Links

Printables

None

Digital Resources

The role of computer arithmetic: https://www.youtube.com/watch?v=O5nskjZ_GoI

Feedback Link: ★★★★★

Data Types Worksheet

Background on data types:

A data type is a kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it.

Just like there are different kinds of animals in biology, there do exist different types of data in computing. Each of them has a particular purpose and can store different things.

Almost all programming languages include the notion of data type, though different languages may use different terminology. Some of the common data types include:

String: Can store anything, such as '18 Sunshine Road, 1234 Moon Colony'

Integer: Can be a whole number, such as 12345

Boolean: Can be a Yes/No decision, often expressed as 'true' or 'false'

Float: Can be any number, such as 1.23478

Select which data types you would use to store the following information

- 1) An entry in a phone book
- 2) Your friend's mobile phone number
- 3) A shopping list
- 4) The result of rolling a dice
- 5) The result of 3 divided by 2
- 6) The decision of going to the cinema tonight, or not.
- 7) Your favourite colour
- 8) The number of people in the world
- 9) The number of relatives a person has
- 10) The height (in metres) of the Eiffel Tower in Paris

Solutions

- 1) String, because a phone book entry consists of a name and a number
- 2) String, because of the international prefix (+61) or because of leading zero, such as 02. An Integer would not be able to represent the '+' or the leading zero.
- 3) String
- 4) Integer, because a dice will only produce whole numbers
- 5) Float
- 6) Boolean, because it is sufficient to store yes or no
- 7) String, because a colour has a name
- 8) Integer
- 9) Integer
- 10) Integer

DT Challenge Year 5/6 Blockly – Chatbot

Module 3: Strings: Working with Words

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/strings/>

Previous: Data Types: Numbers and Strings
Project 1

Next:

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as string
Specification, algorithms, implementation	Simple Algorithms
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps
ACTDIP020	Implement digital solutions as simple visual programs involving user input.

What are we learning? (Abstract)

Computer programs spend a lot of their effort in modifying strings. Some of the most common operations are to make strings lowercase, uppercase and replacing letters in a given string.

Module outline

This module is about simple string manipulation: It consists of three activities: The first activity introduces the concept of string manipulation by means of lowercase and uppercase operators. The second activity teaches students to replace substrings and individual characters in a string. In the third activity, students learn how to determine the length of a string.

Guiding Questions

What can the computer tell us about a string and the characters and words inside it?
How do we change strings?

Elements

Representations

Types of data

Sequencing

User / Environmental Input

Output

Visual programming

Purpose/Hook - Strings



Discussion:

What can we do with Strings inside a computer? Can we ask the computer to tell us about the length of the String? Why would this be important?

What could be good reasons to change a string in a computer? Think of a telephone book.

Structure of the lesson :

Types of component:



Discussion



Worksheet



Plugged Activity



Group Activity



Unplugged Activity



Video



Animation



Reflection



Game



App

Lesson Components:

Step 1: Strings: Uppercase and Lowercase



Plugged Activity

This activity introduces introduces the concept of string manipulation by means of lowercase and uppercase operators.

Start at Module 3, slide 0

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/strings/0/>

Step 2: String Length

**Plugged Activity**

In this activity, students learn how to determine the length of a string.

Start at Module 3, slide 5

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/strings/5/>

Step 3: Parts of a string**Plugged Activity**

In this activity, students learn how to replace substrings and individual characters in a string.

Start at Module 3, slide 8

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/strings/8/>

Review:

**Group Activity**

Students share some examples where they think the computer could be used to help them with string manipulation. Possible Examples: Spellchecker, School directory (people move and change addresses and phone numbers).

Resources and Links

Printables

None

Digital Resources

None

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 4: Project 1 - Making Simple Games

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/project1/>

Previous: Strings: Working with Words

Next: Making Decisions

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as string
Specification, algorithms, implementation	Simple Algorithms
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps
ACTDIP020	Implement digital solutions as simple visual programs involving user input

What are we learning? (Abstract)

In this module, students consolidate their learnings from the previous modules through practical activities that combine input, output and variables.

Module outline

The module consists of three activities of increasing difficulty: In the first activity, students combine two input strings with some static text in order to produce a simple sentence. In the second activity, students extend this concept to program simple nonsensical stories with user inputs for names, relations, adjectives, verbs, etc.

In the third activities, students apply the same concept to program the Swedish Chef from the Muppet Show.

Elements

Representations

Types of data

Sequencing
User / Environmental Input
Output
Visual programming
Designing (Algorithms)

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 5: Making decisions

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/decisions/>

Previous: Project 1

Next: Investigating Strings

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as integer and string
Specification, algorithms, implementation	Simple Algorithms, decisions, branching, reading flowcharts, comparison operators
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps, and branching
ACTDIP020	Implement digital solutions as simple visual programs involving branching and user input.

What are we learning? (Abstract)

Computer programs are able to make decisions through the use of if statements. If statements comparing things, such as two variables, or a variable and a constant value. If statements result in yes/no decisions, which lead to branching.

Module outline

The module consists of four parts: The first part introduces the concept of simple decisions via if-statements. In the second part, decisions with two options are introduced, in which the program runs different code depending on the result of a decision being true or false. In the third part of the module, further comparison operators are introduced. In the fourth and final part of the module, students learn about refining decision making through nested decisions.

Guiding Questions

How do we get the computer to make simple decisions?
How does decision making lead to branching?
How can more complex decisions be made by a computer?

Elements

Representations
Types of data
Whole numbers represent data
Sequencing
User / Environmental Input
Output
Branching
Visual programming
Designing (Algorithms)
Flowcharts

Purpose/Hook - Decisions



Discussion:

Did you make a decision today? What was the decision. How did you come to a result. Was it hard?

Structure of the lesson :

Types of component:



Discussion



Worksheet



Plugged Activity



Group Activity



Unplugged Activity



Video



Animation



Reflection



Game



App

Lesson Components:

Step 1: Making Decisions



Video

Hour of Code - Bill Gates explains If statements

<https://www.youtube.com/watch?v=m2Ux2PnJe6E>



Plugged Activity

This activity introduces the concept of simple decisions via if-statements. If statements consist of a comparison block at the top, which is followed by one or more blocks in the 'do' section, which are executed when the if-condition is true.

Start at Module 5, slide 0

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/decisions/0/>

Step 2: Decisions with two Options



Plugged Activity

This activity introduces decisions with two options, in which the program runs different code depending on the result of a decision being true or false.

Start at Module 5, slide 5

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/decisions/5/>

Step 3: Decisions about Numbers



Plugged Activity

In this activity, further comparison operators are introduced: not equal (!=), less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=)

Start at Module 5, slide 10

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/decisions/10/>

Step 4: Making Complex Decisions



Plugged Activity

In this activity, students learn about refining decision making by means of nested decisions.

Start at Module 5, slide 14

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/decisions/14/>

Review:



Group Activity

Students share some examples how they would code decisions in the school environment or at home.

Example: What decisions are made at school when a parent reports that a student is sick? Who needs to be notified? What entries need to be made in the school's administration system?

Resources and Links

Printables

None

Digital Resources

Hour of Code - Bill Gates explains If statements:

<https://www.youtube.com/watch?v=m2Ux2PnJe6E>

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 6: Investigating Strings

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/string-decisions/>

Previous: Making Decisions

Next: Project 2

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as integer and string
Specification, algorithms, implementation	Simple Algorithms, decisions, branching, comparison operators
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps and branching
ACTDIP020	Implement digital solutions as simple visual programs involving branching, and user input.

What are we learning? (Abstract)

Text is fundamental to our civilisation. Therefore, software engineers have developed algorithms to work with strings as if they were numbers. This enables computers to extract strings, change their case as needed, and to make decisions based on the content and shape of a string.

Module outline

In this module students learn how to manipulate strings so that they can write more interesting programs that are a bit smarter.

The module consists of five activities: In the first activity, students learn how to find substrings in a string and to make simple decisions if or if not a substring has been found. The second activity introduces students to the concept of lists (specifically arrays) which allow students to access individual characters inside a string. In the third activity, students extend their knowledge of uppercase and lowercase towards decisions that are connected with the case of an entire string.

The fourth activity of the module extends from the first activity and introduces students to techniques concerning the extraction of substrings from a string in relation to the beginning, the end or somewhere inside the string.

Guiding Questions

How can we make decisions on the content of a String?

How can we extract specific information from a String?

Elements

Representations

Types of data

Whole numbers represent data

Sequencing

User / Environmental Input

Output

Branching

Visual programming

Designing (Algorithms)

Purpose/Hook - String Manipulation



Discussion:

What can we do with Strings inside a computer? Can we ask the computer to tell us about the content of a String? How can we find a name inside a long text?

Structure of the lesson :

Types of component:



Discussion



Worksheet



Plugged Activity



Group Activity



Unplugged Activity



Video



Animation



Reflection



Game



App

Lesson Components:

Step 1: Investigating Strings



Plugged Activity

In this activity, students learn how to find substrings in a string and to make simple decisions if or if not a substring has been found.

Start at Module 6, slide 0

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/string-decisions/0/>

Step 2: Checking String Case



Plugged Activity

In this activity, students extend their knowledge of uppercase and lowercase towards decisions that are connected with the case of an entire string.

Start at Module 6, slide 4

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/string-decisions/4/>

Step 3: Characters in a String



Plugged Activity

This activity introduces students to the concept of lists (specifically arrays) which allow students to access individual characters inside a string.

Start at Module 6, slide 9

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/string-decisions/9/>

Step 4: Slices of Strings



Plugged Activity

This activity of the module extends from the first activity about substrings and introduces students to techniques concerning the extraction of substrings from a string in relation to the beginning, the end or somewhere inside the string.

Start at Module 6, slide 16

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/string-decisions/16/>

Review:



Group Activity

Students share some examples where they think the computer could be used to help them with string extraction. Possible Examples: Internet Search Engine, Searching for text inside a PDF document.

Resources and Links

Printables

None

Digital Resources

None

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 7: Project 2 – More Word Games

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/project2/>

Previous: Investigating Strings

Next: Repeating Things

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as string
Specification, algorithms, implementation	Simple Algorithms, decisions, branching, comparison operators
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps, and branching
ACTDIP020	Implement digital solutions as simple visual programs involving branching and user input.

What are we learning? (Abstract)

In this module, students consolidate their learnings from the previous modules through practical activities that combine input, output and variables.

Module outline

The module consists of two activities: In the first activity, students write a helper program for the game Taboo. Taboo is a word game where one person describes a word on a card so that their partner can guess it. However, there are certain words that the person can't say when trying to describe the thing! A player might have to describe 'cereal' without using the word 'breakfast'. The program analyses a string to determine if it contains a forbidden word and notifies the user accordingly.

The second activity is about the game word chain. This is a word game where players take turns saying words that start with the last letter of the previous word. Students are tasked to write a program to help them and their friends play word chain. The program should read in two words and print out whether they are valid to follow each other.

Elements

Representations

Types of data

Whole numbers represent data

Sequencing

User / Environmental Input

Output

Branching

Visual programming

Designing (Algorithms)

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 8: Repeating Things

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/loops/>

Previous: Project 2

Next: Project 3

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as integer and string
Specification, algorithms, implementation	Simple Algorithms, loops
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition)
ACTDIP020	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input.

What are we learning? (Abstract)

Looping/iteration is the act of repeating the same or similar steps over and over again until a result has been produced. Looping and automation are strongly connected. In this module, students learn about the foundations of repetition through simple repeat loops.

Module outline

The module consists of five activities: The first activity introduces the repeat loop. The second activity explores some typical conditions that can be used in the repeat loop with strings. In the third part, we apply if statements to make decisions inside loops. In the fourth activity, we apply the knowledge about integer variables in order to make a counter that keeps track of the number of iterations of a while loop. The fifth activity, students apply the learnings from this module to write a program that counts the lines that contain a given term within a text.

Guiding Questions

How can we get a computer program to repeat things?

Elements

Representations

Types of data

Whole numbers represent data

Sequencing

User / Environmental Input

Output

Branching

Iteration

Functions

Visual programming

Designing (Algorithms)

Flowcharts

Purpose/Hook - Repetition and Automation



Discussion:

Name things in our world that repeat on a regular basis.

Structure of the lesson :

Types of component:



Discussion



Worksheet



Plugged Activity



Group Activity



Unplugged Activity



Video



Animation



Reflection



Game



App

Lesson Components:

Step 1: Repeating Things



Video

Hour of Code - Mark Zuckerberg teaches Repeat Loops. Sufficient to watch first 45 seconds.

<https://www.youtube.com/watch?v=mgoogyWMTxk>



Plugged Activity

This activity introduces the concept of looping with the repeat-loop.

Start on Module 8, slide 0

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/loops/0/>

Step 2: Reading while True



Plugged Activity

In this activity, we explore some typical conditions that can be used in the repeat loop with strings.

Start on Module 8, slide 5

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/loops/5/>

Step 3: Making Decisions Inside a Loop



Plugged Activity

In this activity, students learn to combine decision making and loops apply if-statements to make decisions inside a loop.

Start on Module 8, slide 9

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/loops/9/>

Step 4: Counters



Plugged Activity

Keeping track of the number of iterations of a loop is a very common case in programming. In this activity, students apply the knowledge about integer variables from module 2 in order to make a counter that keeps track of the number of iterations of a loop.

Start on Module 8, slide 12

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/loops/12/>

Step 5: Counting with Decisions



Plugged Activity

In this activity, we apply the learnings from this module to write a program that counts the lines that contain a given term within a text.

Start on Module 8, slide 17

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/loops/17/>

Review:



Reflection

Loops are useful to automate tasks. Can you think of a task at home or at school that you would want to automate? How would you approach this automation?

Resources and Links

Printables

None

Digital Resources

Hour of Code - Mark Zuckerberg teaches Repeat Loops. Sufficient to watch first 45 seconds.

<https://www.youtube.com/watch?v=mgooqyWMTxk>

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 9: More complex word games

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/project3/>

Previous: Repeating Things

Next: Project 4: Putting it all together

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as string
Specification, algorithms, implementation	Simple Algorithms, decisions, branching, comparison operators
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition)
ACTDIP020	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input.

What are we learning? (Abstract)

In this module, students consolidate their learnings from the previous modules through practical activities that combine input, output and variables.

Module outline

The module consists of three activities: The first activity is called Questions. This is a game played by maintaining a dialogue of only questions for as long as possible. Students have to write a program that reads in each line of dialogue and checks that it is a question, printing Statement! and ending the game if a line is not a question. In this game, we will assume that everything ending in a question mark ('?') is a question.

The second activity is an extension of the Taboo game from project 2. In this extended version, students have to deal with uppercase/lowercase and substrings to find the taboo word in a given

string. To increase the level of automation, the decision code is embedded in a loop that runs until a taboo word is identified.

The third activity is an extension of the Word Chain game from project 2. Like in the second activity above, students have to convert all input to lowercase strings to ensure that strings are rejected even if their case differs. To increase the level of automation, a loop is executed until the user enters an empty string to end the game.

Elements

Representations

Types of data

Whole numbers represent data

Sequencing

User / Environmental Input

Output

Branching

Iteration

Visual programming

Designing (Algorithms)

Feedback Link: ★★★★★

DT Challenge Year 5/6 Blockly – Chatbot

Module 10: Project 4 - Putting it all together

<https://groklearning.com/learn/aca-dt-56-bk-chatbot/project4/>

Previous: Project 3

Next: None

Key Concepts

Key Concept	Coverage
Abstraction	
Data: collection, representation, interpretation	Data representation as string
Specification, algorithms, implementation	Simple Algorithms, decisions, branching, comparison operators, iteration
Digital Systems	
Interaction	Interaction (command line input/output)
Impact	

Objectives (Content Descriptions)

ACTDIK015	Examine how whole numbers are used to represent all data in digital systems
ACTDIP019	Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition)
ACTDIP020	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input.

What are we learning? (Abstract)

In this module, students consolidate their learnings from the previous modules through practical activities that combine input, output and variables.

Module outline

The overarching topic of this module is the development of a chatbot. That is a program that can interact with a user and respond through the command line. Through four activities, students are introduced to Captain Featherbot and extend its functionality step by step. In the final activity, students are given a blank slate to develop their own chatbot, if they choose to do so. This is an open activity.

In order to successfully master these activities, students have to apply all concepts from this course.

Elements

Representations

Types of data

Whole numbers represent data

Sequencing

User / Environmental Input

Output

Branching

Iteration

Visual programming

Designing (Algorithms)

Feedback Link: ★★★★★