4 Iteration

### General Teacher Notes

#### Iteration

Iteration is used to loop certain lines of code to prevent the coder having to type them again and again.

There are several different ways of doing this. We are using a **conditional** iteration called a while loop. This type of loop runs **while** a condition (or conditions) is/are true.

A while loop starts with the command  **while** followed by the condition(s) for the loop to run.

The loop will only run while the condition is true, otherwise the program will skip the loop and carry on with the rest of the program

Instructions to be repeated go **inside** the loop, in python we do this by indenting the code.

Students often have trouble identifying which instructions should be inside the loop. Question them to get them to think about which steps of their algorithm the computer should repeat.

The code for a while loop looks like this:

while (*condition goes here):*

*Instructions to be repeated are indented to put them inside the loop*

*This can be as many instructions as necessary.*

*They will be repeated as long as the condition is true.*

*This instruction is not in the loop because it isn't indented. The loop has now ended.*

### 

### 

### 

### 

### 1 - Validation

#### TEACHER NOTES

Previously we have used selection to create error messages for incorrect input, using iteration allows us to get the program to repeat until we get the input that we want.

The condition should be set to be true when the input is **NOT** what is acceptable.

For text input use != not equal to.

For number range use the ‘outside a range’ conditions from the selection lessons.

Students often miss the ‘get input again inside the loop’ step. If they do this, they will create an infinite loop that never ends because the variable storing the input will never be changed.

# Checks a password and loops until the user enters the correct one.

correctPW = “pa55w0rD”

#Get the input

password = input(“Enter your password”)

# Set the loop with condition to catch incorrect input

while correctPW ! = password:

#Get input again inside the loop - this will be repeated while the condition is true

password = input(“Incorrect password, try again.”)

#Outputs once the loop ends - when the condition is false.

print(“Password accepted)”

# Loops while number input is not between 1 and 10

#Get the input

num1 = input(“Enter a number between 1 and 10”)

# Set the loop with condition to catch input outside the range

while num1 < 1 or num1 > 10:

#Get input again inside the loop - this will be repeated while the condition is true

num1 = input(“Incorrect input, try again.”)

#Outputs once the loop ends - when the condition is false.

print(“Number accepted)”

#### Validation - Tasks

##### Predict And Run

Task and instructions - <https://repl.it/@MrAColley/241-Validation-Predict-Run>

Example solution - <https://repl.it/@MrAColley/241-Validation-Predict-Run-Solution>

##### Investigate And Modify

Task and instructions - <https://repl.it/@MrAColley/242-Validation-Investigate-Modify>

Example solution - <https://repl.it/@MrAColley/242-Validation-Investigate-Modify-Solution>

##### Make

1. Task and instructions - <https://repl.it/@MrAColley/243-Validation-Make>

1. Example solution - <https://repl.it/@MrAColley/243-Validation-Make-1-Solution>

A more complex version of make 1

2. Task and instructions - <https://repl.it/@MrAColley/243-Validation-Make-2>

2. Example solution 1 - <https://repl.it/@MrAColley/243-Validation-Make-2-Solution-1> An inefficient solution with lots of potentially confusing while loops and nesting

2. Example solution 2 - <https://repl.it/@MrAColley/243-Validation-Make-2-Solution-2>