- **Loops** are structures that we can use to repeat instructions until a certain condition is met.  
- When we see repetition in our code, it’s a good sign that a loop may prove useful.   
- By not writing out instructions multiple times, we reduce our chances of errors and save ourselves some time.

A black background with white text

Description automatically generated

**While Loops:**

- Used to repeat a set of instructions based on a set of conditions  
- Similar to an if statement, it executes the code inside of it if the condition of the statement is true  
- Useful when you know at what spot a program should stop

A blue screen with white text

Description automatically generated A black background with white text

Description automatically generated

- Will not stop executing code inside of it while the condition is true (can lead to an infinite loop)

**Do…While Loop:**

- Will continue to run until a stopping condition is met but will always run once  
- Will execute and then check condition in code block. Will continue running while condition is true  
- Good for when a program should execute at least once and then, depending on the circumstances, continue executing or stop

A screen shot of a computer program

Description automatically generated

**For Loop:**

- Used when we want our code to execute a specific number of times  
- The for loop tells the computer how many times to repeat the instructions using the for keyword and three expressions inside of parentheses. Each of these expressions uses what’s known as an iterator variable, which is a variable that keeps track of how many times the program goes through the loop.

A screen shot of a computer code

Description automatically generated A computer screen shot of a number of text

Description automatically generated with medium confidence

- Initializer: defines where the loop begins via the iterator variable  
- Conditional expression: the condition that the iterator variable is evaluated against — the loop will run until this condition evaluates to false  
- Iteration expression: used to update the iterator variable on each loop

**For Each Loop;**

- Used to iterate over collections, such as in an array or other collections

A screen shot of a computer

Description automatically generated A math equation with numbers

Description automatically generated with medium confidence  
A computer screen shot of a computer code

Description automatically generated

**Comparing Loops:**

- while loops are good when you know your stopping condition, but not how many times the loop will need to run.  
- do...while loops are only necessary if you definitely want something to run once, but then stop if a condition is met.  
- for loops are best if you want something to run a specific number of times, rather than given a certain condition.  
- foreach loops are the best way to loop over an array, or any other collection.

**Jump Statements:**

- There are keywords we can add to for loops to control flow, typically within the context of while loops to prevent them from creating infinite loops

**Break:**

- Ends a loop immediately

A computer screen shot of a computer code

Description automatically generated

**Continue:**

- Used to bypass portions of a loop.  
- It will ignore any additional code in the loop body, go back to the top, and move on to the next iteration.

A computer screen shot of a code

Description automatically generated A screenshot of a computer program

Description automatically generated