

# While Loops: Searches

## Introduction

A `while` loop is another way to iterate code, as you discovered while using the Scratch™ programming language. Unlike `for` loops, which are used to iterate across a collection of a known length, a `while` loop is controlled by a conditional expression that might be less predictable. The conditional expression is evaluated once before an iteration through the `while` block and again before each additional iteration. If the conditional expression is `False` when evaluated (always at the beginning/end of an iteration), execution jumps to the code after the `while` block.



## Procedure

1. Sorting and searching are two important algorithmic problems. You can compare different sorting algorithms (insertion sort, selection sort, bubble sort, etc.) at <http://www.sorting-algorithms.com/>. Which algorithm is fastest when a list is nearly sorted? Compare the algorithms empirically as shown below.
2. The most important solutions to the problem of searching are linear search (checking one at a time, potentially going through a whole list) and binary search (checking against the middle term in a list to narrow the search down to half the items with each comparison.)

Pair up with someone whose birthday you do not know. Find out their birthday using binary search, asking whether their birthday is earlier or later in the year than each of your guesses. How many guesses might be required to guarantee you find out their birthday?

Sorting a list helps you search the list. Explain why an unsorted list cannot be searched with binary search.

Sorting consumes a lot of processor cycles. It's not worth it if you're only going to search a list once, but definitely worth the time to sort if you're going to search the list a lot. Explain how you know whether Google keeps its inventory of webpage content sorted.