Strings in C++

- Recursive Functions
  - More exploration on a theme
- Strings in C++
  - Representing and manipulating text
- Recursion on Strings
  - Featuring cute animals

Thinking Recursively

```
if (The problem is very simple) {
   Directly solve the problem.
   Return the solution.
 } else {
   Split the problem into one or more
   smaller problems with the same
   structure as the original.
   Solve each of those smaller problems.
   Combine the results to get the overall
   solution.
   Return the overall solution.
}
```

These simple cases are called *base cases*.

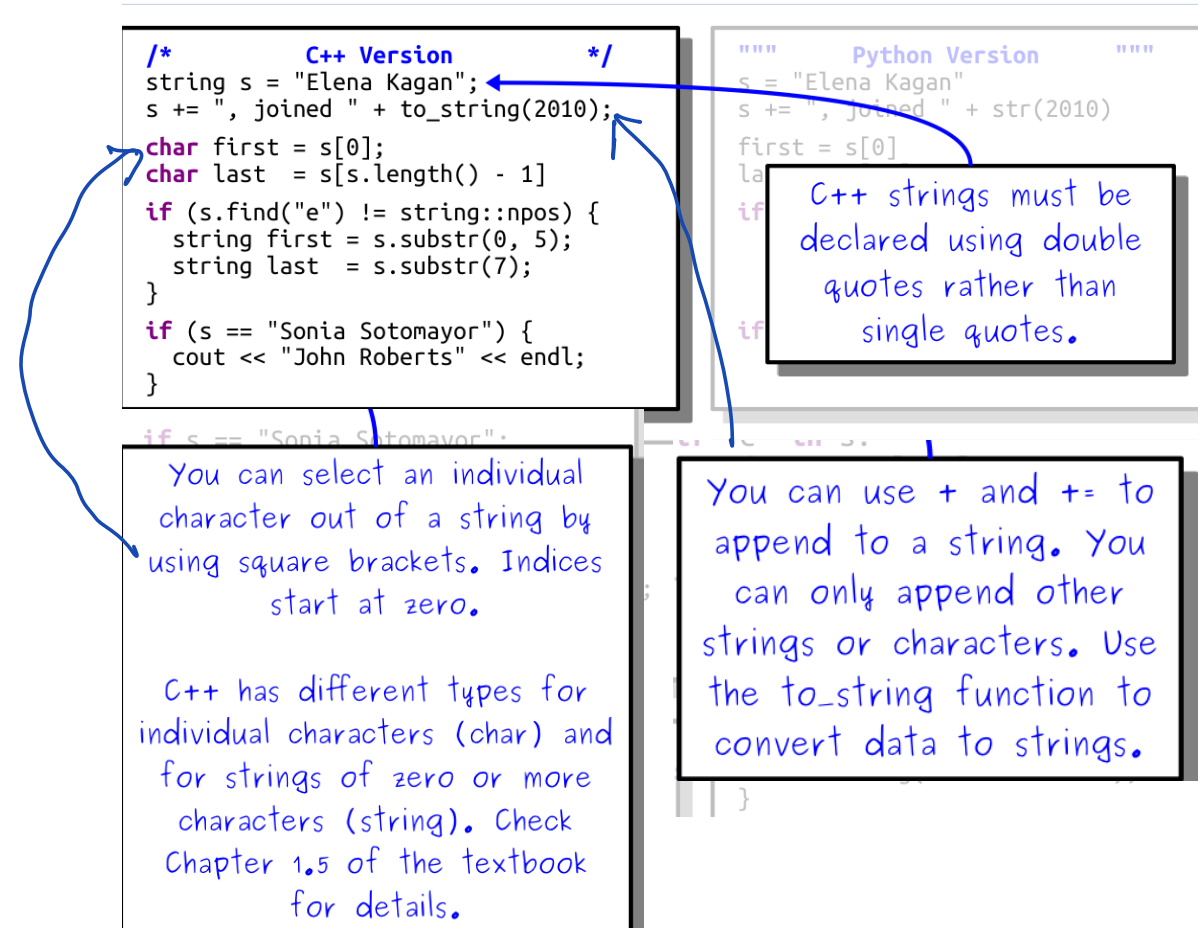
These are the *recursive cases*.

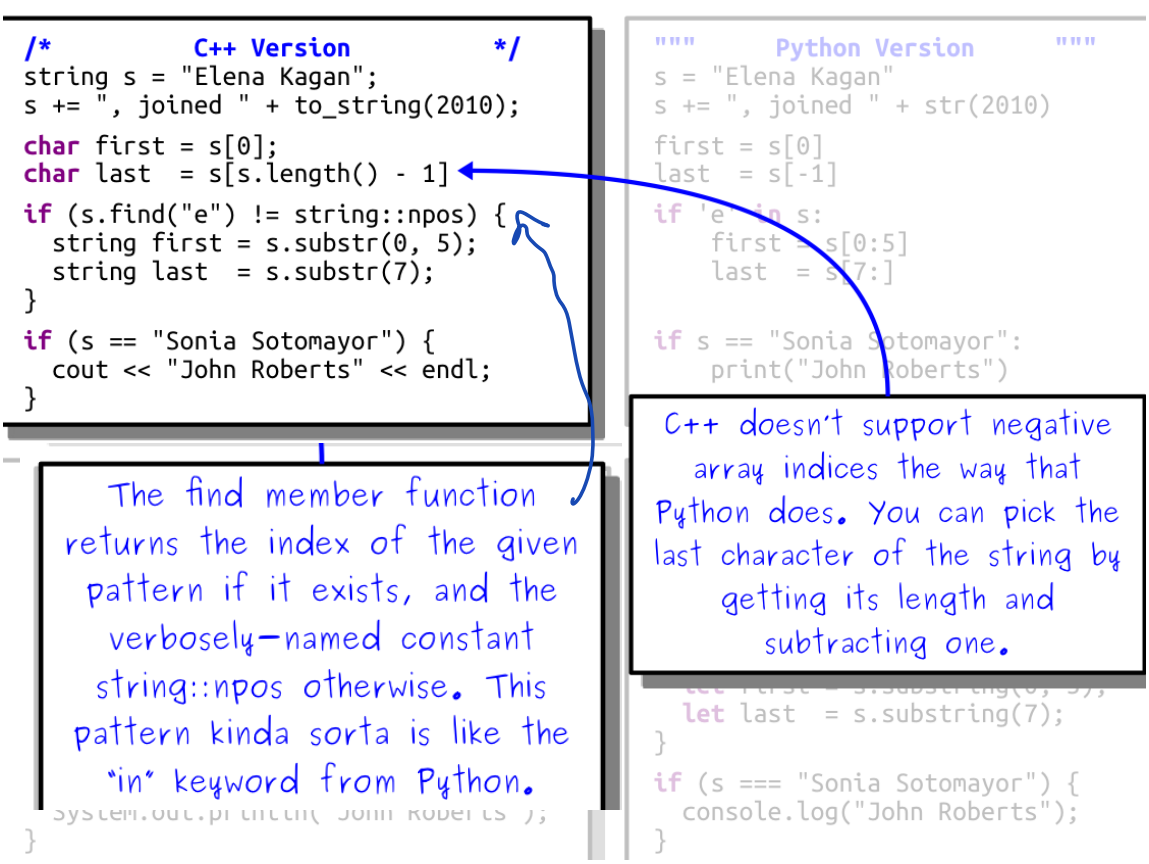
Digit Roots

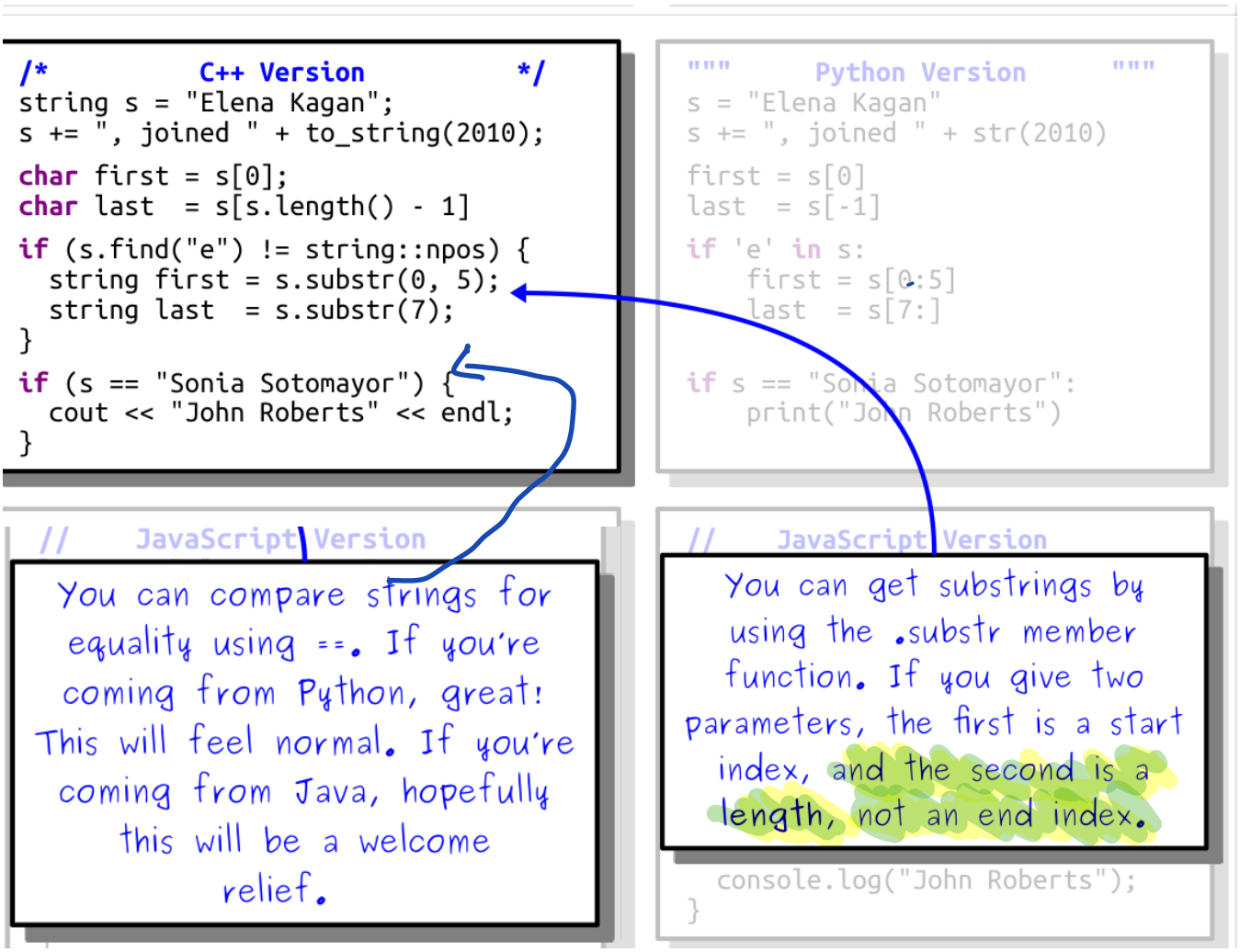
The digital root is the number you get by repeatedly summing the digits of a number until you're down to a single digit.

C++ Strings

To use strings, you need to add the line `#include <string>` to the top of your program to import the strings library.


/* C++ Version */
string s = "Elena Kagan";
s += ", joined " + to_string(2010);
char first = s[0];
char last = s[s.length() - 1]
if (s.find("e") != string::npos) {
string first = s.substr(0, 5);
string last = s.substr(7);
}
if (s == "Sonia Sotomayor") {
cout << "John Roberts" << endl;
}
""" Python Version """
C++ strings must be declared using double quotes rather than single quotes.
You can select an individual character out of a string by using square brackets. Indices start at zero.
C++ has different types for individual characters (char) and for strings of zero or more characters (string). Check Chapter 1.5 of the textbook for details.
You can use + and += to append to a string. You can only append other strings or characters. Use the to_string function to convert data to strings.


/* C++ Version */
string s = "Elena Kagan";
s += ", joined " + to_string(2010);
char first = s[0];
char last = s[s.length() - 1]
if (s.find("e") != string::npos) {
string first = s.substr(0, 5);
string last = s.substr(7);
}
if (s == "Sonia Sotomayor") {
cout << "John Roberts" << endl;
}
""" Python Version """
first = s[0]
last = s[-1]
if 'e' in s:
if s == "Sonia Sotomayor":
print("John Roberts")
let last = s.substring(7);
if (s === "Sonia Sotomayor") {
console.log("John Roberts");
The find member function returns the index of the given pattern if it exists, and the verbosely-named constant string::npos otherwise. This pattern kinda sorta is like the "in" keyword from Python.
C++ doesn't support negative array indices the way that Python does. You can pick the last character of the string by getting its length and subtracting one.



Recap

- Recursion works by identifying
  - One or more base cases, simple cases that can be solved directly, and
  - One or more recursive cases, where a larger problem is problem is turned into a smaller one.
- C++ strings have some endearing quirks compared to other