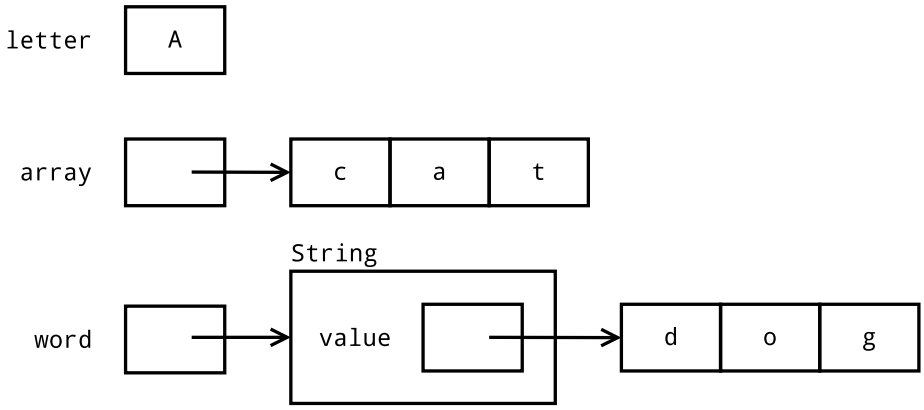
Model 1 Character Arrays

The primitive type `char` is used to store a single character, which can be a letter, a number, or a symbol. In contrast, the reference type `String` ***encapsulates*** an array of characters.

```
char letter;
letter = 'A';

char[] array;
array = new char[]
        {'c', 'a', 't'};

String word;
word = "dog";
```



Questions (15 min) Start time:

1. How is the syntax of character literals (like `'A'`) and string literals (like `"dog"`) different?

2. What is the index of `'d'` in the string above? What is the index of `'g'`?

3. What is the *value* of a `char` variable (i.e., stored in the variable's memory)? What is the *value* of an array variable? What is the *value* of a `String` variable?

4. Based on the diagram, what does it mean for an object to encapsulate data? How do you access the `char[]` inside of a `String` object?

5. Draw a memory diagram for the given code. (List the name of each variable next to a box containing its value.)

```
String str;
str = "Hi!";

char let;
let = 'X';

int num;
num = -1;

double foo;
foo = num;

String hmm;
hmm = str;
```

6. Recall that the == operator compares the *value* of two variables. What does it mean for two `char` variables to be ==? What does it mean for two `String` variables to be ==?

7. How could you determine whether two character arrays have the same contents? In other words, how does the `String.equals` method work internally?