

Get Parts of Strings

We can also use built-in .NET methods to grab parts of strings or specific characters in a string.

Substring

`.Substring()` grabs part of a string using the specified character position and continues until the end of the string and returns a new string. `.IndexOf()` is usually used first to get the specific character position.

```
string plantName = "Cactaceae, Cactus";  
int charPosition = plantName.IndexOf("Cactus"); // returns 11  
string commonName = plantName.Substring(charPosition); // returns Cactus
```

`.Substring()` is useful if we only want to use part of a string, but keep the original data intact. So in this instance, we want to keep the string `plantName`, but just grab the `"Cactus"` portion of the string. We use `.IndexOf()` to find where `"Cactus"` starts, then use `.Substring()` with the position information to save `"Cactus"` to the new variable `commonName`.

Bracket Notation

Bracket notation is a style of syntax that uses brackets `[]` and an integer value to identify a particular value in a collection. In this case, we can use it to find a specific character in a string.

```
string plantName = "Cactaceae, Cactus";  
int charPosition = plantName.IndexOf("u"); // returns 15  
char u = plantName[charPosition]; // returns u
```

Similar to the example above, we first use `.IndexOf()` to grab the character position, which in this case is 15. We then take the string value and append it with a set of brackets `[]` and place the `charPosition` value inside the brackets.

1.

You're creating a form letter and it addresses recipients by their first initial and last name, however you only have their full names.

First, use `.IndexOf()` and bracket notation to grab the first letter of the first name and save it to the variable `firstLetter`.

Hint



Remember that you'll need to use several data types as part of this process:

```
string plantName = "Cactaceae, Cactus";  
int charPosition = plantName.IndexOf("u"); // returns 15  
char u = plantName[charPosition]; // returns u
```

2.

Next, use the `.IndexOf()` and `.Substring()` methods to get the entire last name. Save this value to a variable called `lastName`. Run the code and see your results printed to the console.

Hint



Remember to type the *exact* string that you want to grab:

```
string plantName = "Cactaceae, Cactus";  
int charPosition = plantName.IndexOf("Cactus"); // returns 11  
string commonName = plantName.Substring(charPosition); // returns Cactus
```

