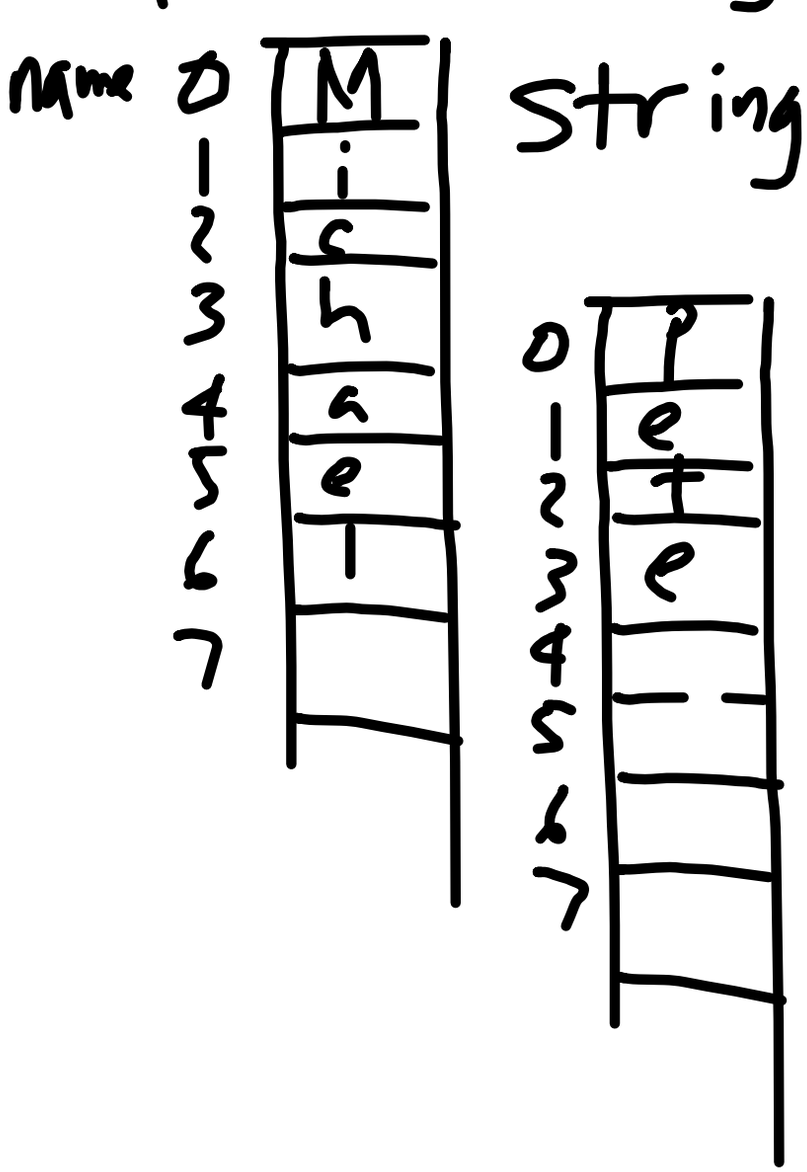
Strings

```
string name      = "Michael";
string name      = "Pete";
string name      ;
```



declaring (initializing)

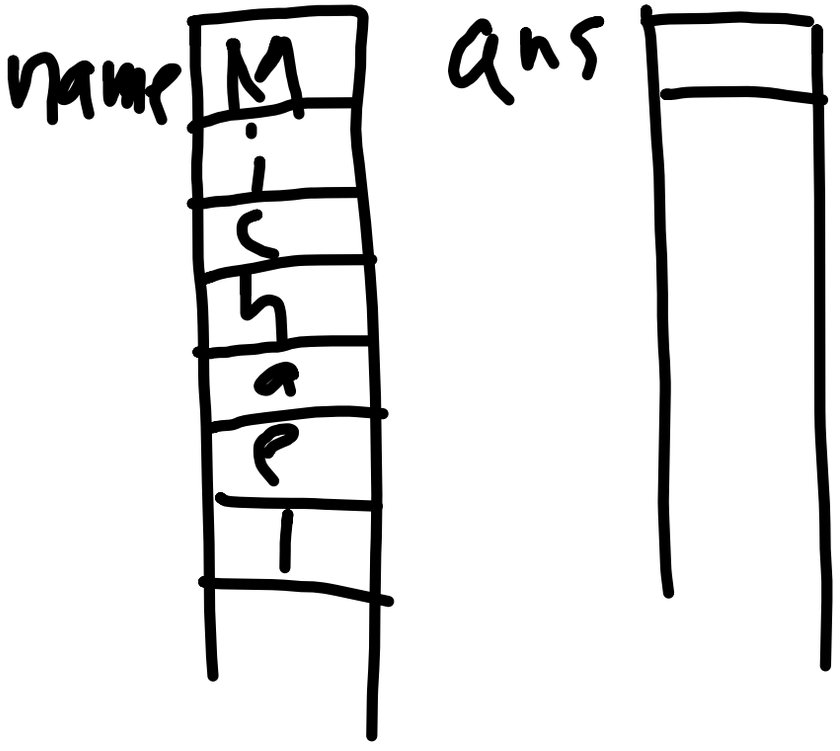
Think of Strings as primitives even though they are not

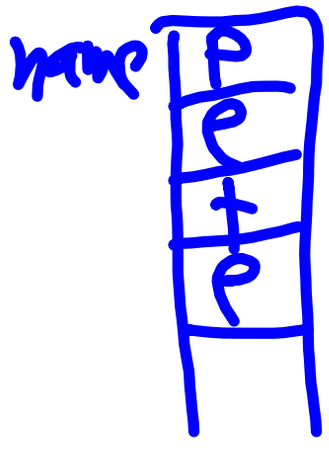
technically primitives.

(Primitives: int, long, float, double, bool, char)

Also, Strings are immutable. That means that once you create them, they cannot be changed.

name[3] = 'k'; // will not work to change a single char in a String.





name = "Pete"; //works

even though already had a value

Think of Strings as primitives even though they are not technically primitives.

You can use the + and += operators

```
name += " is my name";
name = "my name is: " + name;
```

→ "Michael is my name"

→ "my name is Michael"

You can use equality comparators ==, !=, etc.

```
if (name == "stewart") Console.Write("Goodbye.");
if ("Gwen".ToLower() != name) Console.Write("Hello.");
```

String name = "Michael";

Comparing Strings

String s1, s2;

Compare strings with

int result = String.Compare(s1, s2);

or

int result = s1.String.CompareTo(s2);

result

0 means equal

negative means first preceded second alphabetically

positive means first follows second alphabetically

You can use something called a StringBuilder if you would like your "strings" to not be immutable. (But you will need to look it up yourself.)

binary number conversion

bin → dec

(8 bits)

string bin = "";

```
  0 1 2 3 4 5 6 7   ← index
bin 1 0 1 0 0 0 1 1
    ↑
    2^0
```

suggestion → tell user to input leading zeros

00000001

binary number conversion

bin → dec

can address
single part of array

```
if( bin[0] == '1')
```

bin[i]