

**Arrays** — An array is a *variable* that can hold a **group** of data.

So, for example, you could have a bunch of **integers** packed together as follows:

```
int array[25];
```

This produces a grouping of 25 integers that are all packed together right next to one another. And have **25 locations** in which to put data. The **locations** are labelled 0-24.

```
array[0] // the first item in the array
array[1] // the second item in the array
array[i] // the "ith" item in the array (valid for 0 <= i <= 24)
array[24] // the last item in the array
array[25] // NOT a location in your array(Only 0 to 24)!
```

*The position of data in the array is call **index** in this case the index is labelled *i**

How is this useful? Well, let's say we wanted to store 3 light sensor values, each taken 1 second apart. We could define 3 variables, as in this example:

```
int value1 = SensorValue[light];
wait1Msec(1000);
int value2 = SensorValue[light];
wait1Msec(1000);
int value3 = SensorValue[light];
```

This is all well and good, until you decide that 3 value isn't enough. Let's say you now want to store 100 light sensor values! You probably don't want to have to define 100 variables to define and fill with data So, you can use an array:

```
int MyArray[100];
int i
i=0;

while( i < 100)

{
  MyArray[i] = SensorValue(lightsensor);
  i=i+1;
  wait1Msec(2000);
}
```

Can you see what is happening?...Light values from the light sensor are being stored in the Array called "MyArray" every 2 seconds. The **index** value "*i*" is being used to shift the "to the next space" in MyArray after we fill the previous space.

In summery, **Arrays** are kind of like a variable that have **several storage locations**. These locations all have labels called **an index**. Arrays are very helpful for keeping track of large amounts of data and keeping this data organized and easily located.

In the following example the code:

1. *“fills”* an **Array** with capital letters
2. then displays them on the screen in order.

```
task main ()
{
int i;
char alphabet [26];
i=0;

while (i<26)
{
alphabet [i] = 'A' + i;
i=i+1;
}

i=0;
while (i<26)
{
nxtDisplayCenteredBigTextLine(3,"%c",alphabet[i]);
wait1Msec(900);
eraseDisplay();
i=i+1;
}
}
```

Put this code into RobotC and test it out.

Then print out the code with comments, explaining what the program does and how it does it. **Also include comments for each line explaining what it does.**

## **“Name” Assignment**

**Using your “scroll” code from your last assignment *and* the array example on this handout, do the following:**

- *Get a user to select the letters of their name in order*
- *Place the letters into in an array called “name”.*
- *Have the user press one of the buttons on the NXT to signal they are done.*
- *Then display the name on the screen (with a fun sound to go along with the name,... of course)!*

*Exemplary:*

- *Then prompt the user to press a second button to see their name spelled backwards*

**Hand in your code, an algorithm, and an assessment marking sheet for this one.**

Have fun... or should I say *Nuf evah*