



## Tutorial 1 : Variables

When you run DarkBASIC Professional for the first time, you may realize just how empty a blank code document is. Where do you start? This tutorial will help you get going with a couple simple commands and introduce one of the most important concepts to all programming – variables.

Most of the things computer programs do is behind the scenes. In fact, only a small percent of the code for most of our games actually has to do with how things are displayed to the user. However, we can't see any of the things going on inside our computer unless we have a way to put it on the screen so the first command I'll talk about is the `print` command. This will let you print out numbers and letters to the monitor. To see how it works, make the following simple program .

```
print "Hello World"  
wait 3000
```

One thing you might notice about this code is the `wait` command which I didn't talk about before. The purpose for that is to make the program wait a bit before exiting. The way programs work is the computer executes each line of your code from top to bottom. When it gets past the last line of code, it simply closes. So, I put in the `wait` command so we could see what it printed out. The number after it tells it how many milliseconds to wait. So since 3000 milliseconds is three seconds, it will wait 3 seconds before exiting.

To run this, hit F6 and it should run and display the words, Hello World, in the top left corner of the screen. Note that I put the text in quotes. I'll explain why later but for now, just remember that if you want it to print out words, you always have to put those words in quotes.

If you want to print out numbers, you can just replace "Hello World" with whatever number you want to print out. For example, try making a program like this:

```
print 3
wait 1000
print 2
wait 1000
print 1
wait 1000
print "LIFT OFF!"
wait 3000
```

DarkBASIC is such an easy language that it's easy to see, even before running it, what it will do.

This is a good point to talk about variables. A variable is basically a word that represents a number. This is best explained by an example:

```
countdown = 3
print countdown
```

Since the = sign also means "is" we can interpret this as `countdown` "is" 3. So when we print out the variable `countdown`, it will print out 3.

We can also alter variables after they are made. There are a number of basic mathematic operators DBPro recognizes but the four basic ones are : + (add), - (subtract), / (divide), and \* (multiply). Using this information, we can rewrite the above program as:

```
countdown = 3
print countdown
wait 1000
countdown = countdown - 1
print countdown
wait 1000
countdown = countdown - 1
print countdown
wait 1000
print "LIFT OFF!"
wait 3000
```

So it starts off by setting countdown to 3 and printing it out. Then, after waiting for a second, it subtracts one from countdown ( $3-1=2$ ) and thus prints out 2 when it prints out countdown this time. Then it subtracts one from it again and thus prints out 1.

We can also use multiple variables to create mathematic equations. For example:

```
x=5
y=19
z=2
print 9*x + 5*y - z/x + x*y*z
wait 3000
```

Now, whenever we want to change the values for  $x$ ,  $y$ , and  $z$ , we can just change them once up at the top rather than going through the entire equation and changing them every place they show up. This is particularly useful in larger equations.

Though variables may seem like a bit of a waste at this point, in future tutorials, I'll show how useful and important they are to programmers.

In the meantime, I encourage you to experiment around with them and get very comfortable using them as well as the `print` command, the `wait` command, and the mathematical operators since you will use those in the upcoming tutorials and in future game development if you continue in it, and the best way to become really good at programming, I've found, is to do a lot of it. Also, when I was learning DarkBASIC, I found that if I went for awhile without programming, I quickly lost what I knew so it's something you should not only do a lot, but on a regular basis if you want to be really good at it.