



Python Variables

An Introduction to Variables

A variable allows you to store data throughout your program. It is always best to name a variable to something appropriate so that it is clear what the variable's role is. There are a few rules with regards to the naming of variables in Python:

- Variable names must start with a letter, it cannot start with a number.
- Variable names must only contain alpha-numeric characters (a-z, 0-9 and _)
- Variable names are case sensitive

There are five different variable types that can be used to store data. The data types are: Numbers, Strings, List, Tuple, Dictionary. A variable may be assigned a value of one type and then later re-assigned a value of a different type.

Whatever the variable type, you do not need to declare the variable in advance within your program.

String Variables

A string is another name a collection of characters or words. String variables are either defines in single or double quotes. Using double quotes will allow you to include apostrophes in your strings.

```
name = "Jez"  
print(name)
```

Strings can be joined together as shown below:

```
hello = "hello"  
world = "world"  
helloworld = hello + " " + world  
print(helloworld)
```

```
weather = "raining"  
print("Today the weather is " + weather)
```

Wrapping the string in three single quotes will allow you do display a sentence across multiple lines:

```
MultipleLines = ''' This is over  
two lines'''  
print(MultipleLines)
```

Strings can also be printed out multiple times:

```
print ("How many times?" * 5)
```

Number Variables

Number variables are used to store numerical values against them.

```
Number = 15  
print(number)
```

The operators that are available to us are: +, -, *, /, %, **, // with ** being and exponential calculation and // being floor division. Some example are below:

```
print("5 + 2 =", 5+2)  
print("5 - 2 =", 5-2)  
print("5 * 2 =", 5*2)  
print("5 / 2 =", 5/2)  
print("5 % 2 =", 5%2)  
print("5 ** 2 =", 5**2)  
print("5 // 2 =", 5//2)
```

It is important to remember that when making calculation in your program to remember the order of operation. For example: `print("1 + 2 - 3 * 2 =", 1 + 2 - 3 * 2)` will produce a different result than `print("(1 + 2 - 3) * 2 =", (1 + 2 - 3) * 2)`

List Variables

A list allows you to create variables with multiple values that can be manipulated and changed. Lists contains individual items that are separated by a comma and contained in square brackets. Each list is indexed, with the first value having a value of 0, the second entry having an index value of 1 and so on. This allows us to find our way around the list if we ever need to retrieve or change particular values in a list.

```
shopping_list = ["Milk", "Bread", "Bananas", "Cereal"]  
print('The first item is', grocery_list[0])
```

Don't forget the first item in the list has an index value of 0.

You can also get a range of entries between two index values.

```
print(shopping_list[1:3])
```

You can also have lists contained within lists.

```
todo_list = ["Go shopping", "Call office", "Walk dog", "feed cat"]  
list_list = [todo_list, shopping_list]
```

```
print(list_list)
```

If we wanted to get the second item from the second list:

```
print(list_list[1][1])
```

The first index is the index value of the list, the second value is the index value of the item within that particular list.

To insert into a list:

```
shopping_list.insert(1, "Cat food")
```

To remove an item from the list:

```
shopping_list.remove("Pickle")
```

We can also sort the item within a list:

```
shopping_list.sort()
```

Let's sort them the other way:

```
shopping_list.reverse()
```

To delete an item from a list:

```
del shopping_list[4]  
print(shopping_list)
```

We can also combine lists together:

```
list_list = todo_list + shopping_list  
print(list_list)
```

To get a length of a list:

```
print(len(list_list))
```

To get the maximum item in a list:

```
print(max(list_list))
```

To get the minimum item in a list:

```
print(min(list_list))
```

Tuple Variables

Tuples are similar to lists but their values cannot be changed. Whilst their individual values are separated by commas like lists, they are contained within parentheses ().

```
tuple = (3, 1, 4, 1, 5, 9)
```

Just like with lists you can get the length, min and max values of a tuple using:

```
print(len(tuple))
print(min(tuple))
print(max(tuple))
```

Dictionary Variables

These are values that have a unique key for each entry. A dictionary variable can contain almost any type, but are usually numbers or strings.

```
super_villains = {'Fiddler' : 'Isaac Bowin',
                 'Captain Cold' : 'Leonard Snart',
                 'Weather Wizard' : 'Mark Mardon',
                 'Mirror Master' : 'Sam Scudder',
                 'Pied Piper' : 'Thomas Peterson'}
```

So if we wanted to know the real name of one of the villains:

```
print(super_villains['Captain Cold'])
```

To delete an entry:

```
del super_villains['Fiddler']
print(super_villains)
```

To replace a value:

```
super_villains['Pied Piper'] = 'Hartley Rathaway'
```

To print the number of entries in the dictionary:

```
print(len(super_villains))
```

Changing the Data Type of a Variable

```
i = 22                # The data type is an integer
i = "Twenty Two"     # The data type is now a string
i = str(22)          # Converts the integer to a string
```

Assigning a User Input to a Variable

```
print('What is your name?')
name = sys.stdin.readline()

print('Hello', name)
```