

### Tutorial 3

---

#### Aims

- to become familiar with storing data;
- to be able to define the essential Python types; and
- to get experience using the Date and Time module.

#### 1. Assigning values

- Create a variable and assign it an integer value. Create a statement that prints the <sup>data type</sup> string of your named variable, and a statement that prints the value of your variable.

- Write a program that will output a Boolean value. An example could be to see if one number is greater than another number. Review this link if you need revision on which symbol to use:  
<https://www.mathsisfun.com/equal-less-greater.html>

- Work through the assigning variables example in Lecture 3. Create varA and assign it the value '5'. Create varB and assign it the value '7'. Display the name and value of varA. Display the name and value of varB.

- Write a program that will assign the same integer value to three different variables. Print the value of each variable.

- Work through the example in Lecture 3 that demonstrated simultaneous assignment of variables. pp. 36-37

- Look through the follow table. Which variable names are allowed in Python?

Revisit: PP. 19 - week 3 lecture

Variable Name	Allowed?
units_per_day	
dayOfWeek	
3dGraph	
June1997	
Mixture#3	

- Create a string variable called first\_name, and a string variable called last\_name. Assign the variables with your first name and last name, respectively. Write a statement that will display the first\_name and last\_name variable on the same line.

- Create an int, a float, a Boolean, and a string variable. Determine the type of each type by using the:

```
type(variable)
```

## 2. Write an algorithm that uses arithmetic operators

- Write a program that will assign the same integer value to three different variables. Add the first two variables together and multiply that result by the third variable.

- Write a program that displays the area of a rectangle with the width of 4.5 and height of 7.9 using the following formula:

*Area = width x height*

- Write a program that will consider a number and display the number of 10's in that number (using the '/' operator) and the number of 1's in that number (using the '%' operator).

### 3. Creating multiline strings

To create a string that runs over multiple lines we can use the `'''` at the start and end of the string. An example is:

```
print ('''One  
Two  
Three ''')
```

- Write a statement that displays your name, address and email over multiple lines.

### 4. Importing the datetime command

Remember from the lecture we can import a datetime command to get the current date and time from the computer's internal clock. To do this we first must import the datetime module.

```
import datetime
```

To get the current time we then type:

```
print(datetime.datetime.now())
```

We can also import a calendar module:

```
import calendar
```

Let's see if we can print out the current month's calendar. We can set a calendar value with the following statement:

```
calendar.month(year, month)
```

where year is expressed as the numerical year value (2015) and the month is expressed as the numerical month value (1 to 12).

- What happens when you execute this statement?

We can make it easier to read by doing the following:

```
cal=calendar.month(year, month)
print(cal)
```

## 5. Additional work if you have time

While using pen and paper and not the computer...

Recall from the lectures that the following arithmetic operations exist:

- + (addition)
- − (subtraction)
- \* (multiplication)
- / (division — discarding the remainder if both operands are ints) and
- % (remainder or modulus — discarding the quotient).

Write English/pseudo-code for an algorithm that reverses the digits of a 3 digit number  $x$ . Assume that  $x$  is already declared as a whole number and already given a value within the range 100–999, e.g. 123. Your algorithm should ensure that at the end of the algorithm,  $x$  contains the reverse of its original digits. Hint: you should use all of /, %, and \*, and at least one additional variable.

➤ Implement the program