Data types

1. Introduction to data types

Programs work with many different types of data. Some of it might be strings of alphabetical characters, others might be numbers, others might be values of true or false.

You want your programs to be able to tell these data types apart - trying to perform mathematical operations on a piece of text or convert a list of numbers to upper case doesn't make any sense.

In this section, we will discuss how programs classify data into different types, and what each of those types are used for. The basic data types in computer programming are:-

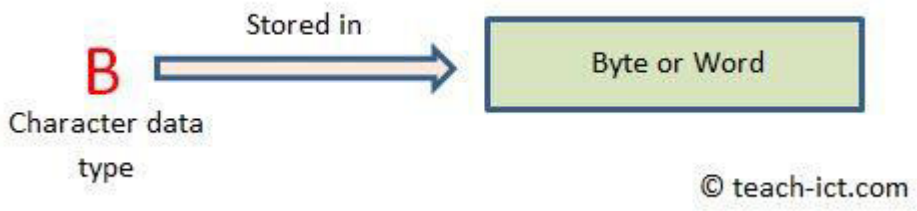
- Character
- String
- Boolean
- Integer
- Real

Data types

2. Character and String

The 'character' data type is able to store a single letter, number or symbol such as 'A', 'b', $, 4.

Using the character data type is very efficient in terms of memory as a character is stored either in a single byte (8 bits) or a single word (word length depends on the cpu type) depending on the computer system.



Every alphabet in the world can be represented by an international standard called 'Unicode'.

String data type

Characters are strung together to form words and sentences. It is convenient to treat these as a *single variable* the data type for this is called 'string'.
We have a whole section dedicated to **describing string handling**.

Examples that use string	
Item	**Description**
Name	Harry Potter
Address	29 Craft Avenue
Telephone number	022984848
Car Registration number	CX45 829
Colours	Blue

Note: Telephone numbers are stored as a string rather than an integer data type because an integer data type will remove any leading zeros from the number, thus 01926 becomes 1926. Also, no mathematics are likely to be carried out on such numbers.

Data types

3. Integer

The integer data type is used to store whole numbers - i.e. those without a decimal point.

For example

1234 is an integer

1234.0 is **not** an integer

An integer can deal with both positive and negative whole numbers.

If you try and store a decimal number in an integer data type, it will cut off everything after the decimal point, which is likely to cause a problem with the program later down the line. This 'type error' can be quite difficult to spot.

```
int MyInteger
MyInteger = 1.234
```

The number in MyInteger will actally be 1, with the decimal part discarded.

Number size matters

Many computer languages (such as 'C') break down integer into different types depending on the kind of numbers you wish to store in it - its size and sign matter a great deal.

unsigned int - can only store positive numbers up to 64,000. It cannot store negative whole numbers.
signed int - stores both positive and negative numbers but with only half the range of an unsigned int as one half of the range is dedicated to negative numbers. For example +/- 32000 rather than 0 to +64000
short int - it only uses a single word (16 bits) for storage so the number must not be larger than about 32,000, or +/-16,000 if it's signed.
long int - it uses two words (32 bits) for storage and can handle huge numbers, but it is wasteful of memory for small numbers.

Data types

4. Real number

The **real** data type also stores numbers. But, unlike integer, **real** is used to store numbers that use decimal points.
However it is important for the program to know how much data after the decimal point to keep track of. Otherwise, if it was asked to store a never-ending number like π, it would fill its entire memory with just one variable.

As such, there are two common formats for real data - "floating point" (or "float") and "real"', depending on how much data you want to keep after the decimal point.

float - uses 4 bytes (32 bits) for storage. It can store numbers up to seven decimal places
real - uses 8 bytes (64 bits) for storage. It can store up to fifteen decimal places

For example, a floating point real number can be declared in Python like this:

```
float MyReal
MyReal = 1.223
```

For most programs, a precision of seven decimal places is good enough, so float is used.

Data types

5. Boolean data type

Computers are based on logic and so to handle a logic variable the **Boolean** data type is used.

The Boolean data type can only contain two values - either TRUE or FALSE.

In terms of a number, logic FALSE is normally a 0, and logic TRUE is 1

In Python for example, if you wish to keep track of a Boolean variable, the True and False statements set its value.

Like this

```
logged_in = false
```

We will discuss how to use boolean data and boolean operators in a later section on computational logic.

Data types

6. Casting

We have covered the standard data types as used in programming. However there is another issue that needs to be covered as well. Namely how do we actually input a data type other than character or string?

If you enter data with the keyboard, the program normally assumes that you are entering characters even if you are entering a set of numbers.

For example, imagine the program is asking you to enter your telephone number, say 01345633

It makes sense to handle this as a string within the program - it makes no sense doing any maths on it. You can't add two telephone numbers!

However, what if the program is asking for your age in years? You still type in a number such as 15, but you need for this to be handled as a proper number. In this case an integer.

There needs to be a way of converting the string '15' into the integer 15. And there is.

It is called 'casting'.

Casting transforms one data type into another data type. Such as string to integer.

Most programming languages have a command to convert one data type into another.

These are the OCR Exam reference language casting commands

Symbol	Example	Purpose
str()	str(123)	number into string
int()	int("4")	string into integer
float()	float("1.234567")	string into 7 digits of precision number
real()	real("1.123456789123456")	string into 15 digits of precision number
bool()	bool("true")	string into boolean true or false

A floating point number can use 4 bytes or 8 bytes of storage. Sometimes called single precision and double precision floating point numbers.

The 4 byte version is called float and the high precision 8 byte version is called real. Obviously using the real data type would be a waste of memory if the program did not need that level of precision.

A typical pseudocode for the use of casting is:

```
// The user enter their age and it is cast into an integer
// then stored in the variable 'age'
age = int(input("Please enter your age in years"))
```

```
// The stored number is output as a string for storage
// in a text file
x = 1.3454
myFile.writeLine(str(x))
```

Data types

6. Summary

- There are five basic data types: Character, String, Boolean, Integer, Real
- Character stores a single character or symbol
- String stores text
- Boolean stores TRUE or FALSE
- Integer stores whole numbers
- Integers can handle both positive and negative numbers
- Real stores numbers with a decimal point and high precision
- Float stores numbers with a decimal point and normal precision
- There are two common real formats - floating point and double