Data types and casting

2.2.2 · Algorithms and programming (Paper 2) · revision notes

Every value in a program has a data type: integer, real, Boolean, character or string. You need to choose the right type for a piece of data and use casting to convert between types. Expect 'choose the most suitable data type' tables and questions on why casting is needed.

Key facts

- Integer: whole number (42, −7)
- Boolean: True or False
- String: a sequence of characters ("CB4 1XY")
- input() always returns a string
- Real / float: number with a decimal part (3.75)
- Character: a single symbol ("A", "7", "?")
- Casting: str(), int(), float() or real(), bool()
- Phone numbers and postcodes: string (leading zeros, spaces, no arithmetic)

Notes

The data types

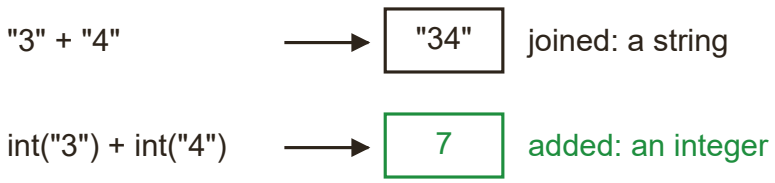
- **Integer**: a whole number, positive or negative, such as 42 or −7.
- **Real** (also called float): a number with a decimal (fractional) part, such as 3.75 or −0.5.
- **Boolean**: one of two values, True or False.
- **Character**: a single letter, digit or symbol, such as "A", "7" or "?".
- **String**: a sequence of characters, such as "CB4 1XY" or "Hello".

Choosing the right data type

- Count of things (pupils, goals, items): integer.
- Money, measurements, averages: real.
- Yes/no, on/off, paid/not paid: Boolean.
- Phone numbers, postcodes, ID codes: string. A phone number can start with 0, which an integer would lose; it may contain spaces or a +; and you never do arithmetic on it.
- Choosing the correct type means the right operations can be done on the data and the program can check that input is valid.

Casting

- **Casting** converts a value from one data type to another.
- `int("12")` gives 12; `str(12)` gives "12"; `float("3.5")` (or `real("3.5")`) gives 3.5; `bool("True")` gives True.
- `input()` always returns a string. Adding two inputs without casting joins them: "3" + "4" gives "34", but int("3") + int("4") gives 7.



Without casting, + joins strings; after casting to integers, + adds

- To join a number to a message, cast it to a string: `print("Score: " + str(score))`.

How to answer each type of question

Choose the most suitable data type 1 mark each **Grade 4**

1. Ask: is it a number that you calculate with? Whole number: integer; decimal: real.
2. Only two possible values: Boolean.
3. One symbol: character. Text, or digits you do not calculate with: string.

Example. State the most suitable data type for each item.
(a) The number of pupils in a class (b) The price of a ticket (c) Whether a ticket has been paid for (d) A customer's postcode (e) The first letter of a surname [5]

Model answer.

(a) Integer (1)
(b) Real (1)
(c) Boolean (1)
(d) String (1)
(e) Character (1)

Explain why casting is needed

2 marks | Grade 6

1. Say that input() returns a string.
2. Say what goes wrong: + joins (concatenates) strings instead of adding.
3. Say how to fix it: cast the input to an integer or real.

Example. The user enters 5 and then 3. The program outputs 53 instead of 8.

```
a = input("Enter first number")
b = input("Enter second number")
print(a + b)
```

Explain why, and how the program should be corrected. [2]

Model answer. The inputs are strings, so + joins (concatenates) them instead of adding them (1).

Cast each input to an integer, for example `a = int(input("Enter first number"))` (1).

Justify a data type

2 marks | Grade 6

1. State the data type.
2. Give a reason based on the data: leading zeros, spaces, decimal places, or whether calculations are done on it.

Example. Explain why a customer's phone number should be stored as a string rather than an integer. [2]

Model answer. A phone number can start with 0, which would be lost if it was stored as an integer (1).

No calculations are done on it / it may contain spaces or a + sign (1).

Work out the result of casting

1 mark each | Grade 7

1. Cast first, then apply the operator.
2. String + string joins; number + number adds.
3. Give the data type of the result if asked.

Example. State the value and data type of each result.
(a) str(15) + "0" (b) int("15") + 5 (c) float("2.5") * 2 [3]

Model answer.

(a) "150", string (1)

(b) 20, integer (1)

(c) 5.0, real (1)

Don’t lose marks

- Storing phone numbers or postcodes as integers.
- Choosing integer for money: prices need real.
- Writing 'number' or 'text' instead of the proper names integer, real and string.
- Forgetting that input() returns a string, so arithmetic joins the digits instead.
- Mixing up character (one symbol) and string.

More tips

Memory tricks

- Ask 'would I ever do maths with it?' If not, it is probably a string, even if it is made of digits.
- Character = one symbol; string = any number of symbols.
- When you see + with input(), check whether it should add or join.

Exam technique

- Use the exact data type names: integer, real (or float), Boolean, character, string.
- In 'justify' questions, give a reason about the data itself, not just the name of the type.
- When writing code, cast numeric input straight away: int(input(...)).

What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 3 — **Name the five data types**
Integer, real, Boolean, character and string.

Grade 4 — **Choose a suitable data type for data**
For example a price is real, a count is integer and a yes/no value is Boolean.

Grade 5 — **Explain what casting is**
Converting a value from one data type to another, such as the string "12" to the integer 12.

Grade 6 — **Explain why casting is needed in code**
Input is a string, so it must be cast before arithmetic, or + will join the strings.

Grade 6 — **Justify storing phone numbers as strings**
They can start with 0 or contain spaces, and no arithmetic is done on them.

Grade 7 — **Work out the result and type of casts**
For example str(15) + "0" gives the string "150".

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.