COMP110: Principles of Computing

.

Research journal .

ASCII
American Standard Code for Information Interchange
Defines a standard set of 128 characters (7 bits per character)
Originally developed in the 1960s for teletype machines, but survives in computing to this day
95 printable characters: upper and lower case English alphabet, digits, punctuation
33 non-printable characters
background canvasbg=white plain] [remember picture, overlay] [at=(current page.center)] [width=]ascii$_c hart_2$;
ASCII
ASCII works OK for English
Standards exist to add another 128 characters (taking us to 8 bits per character)
E.g. accented characters for European languages, other Western alphabets e.g. Greek, Cyrillic, mathematical symbols
However 256 characters isn't enough...
Unicode
Standard character set developed from 1987 to present day
Currently defines 137994 characters (Unicode 12.1)
First 128 characters are the same as ASCII
Covers most of the world's writing systems
Also covers mathematical symbols and emoji
Encoding Unicode
UTF-32 encodes characters as 32-bit integers
UTF-8 encodes characters as 8, 16, 24 or 32-bit integers
8-bit characters correspond to the first 128 ASCII characters backwards compatible
More common Unicode characters are smaller more efficient than UTF-32
String representation
"Hello world!" in ASCII or UTF-8 encoding:
72101108108111321191111114108100330
UTF-8 representation
For characters in ASCII, UTF-8 is the same:
a → [97]
Other characters are encoded as multi-byte sequences:
ü → [195, 188]
[height=1.5ex]chinese → [228, 184, 178]
[height=1.5ex]emoji → [240, 159, 152, 130]
`"Haha [height=1.5ex]emoji"` encoded in UTF-8:
H a h a space[height=1.5ex]emojinull
729710497 32 240159152 130 0
Strings in Python
Python 2 had separate types for ASCII and Unicode strings: str and unicode
Python 3 has just the str type, which uses Unicode
String literals are wrapped in 'single quotes' or "double quotes" (there is no difference)
fragile]Escape sequences
Backslash \ has a special meaning in string literals — it denotes the start of an **escape sequence**
Typically used to write **non-printable characters**
Most useful: "" is a new line
How to type a backslash character? Use "
"
String literal tricks in Python
Use triple quotes "' or """ for a multi-line string
Use r" " or r' ' to turn off escape characters (useful for strings with lots of backslashes, e.g. Windows file paths, regular
fragile]Text files
Stored on disk as essentially one long string
Line endings are denoted by non-printable characters
Unix format: line feed character (ASCII/UTF-8 character 10, "")
Windows format: carriage return character (ASCII/UTF-8 character 13) followed by line feed, " ◦ "
Most text editors can handle and convert both formats
Most languages allow files to be opened in "text mode" which automatically converts
Other types .
Booleans
A **boolean** can have one of two values: **true** or **false**
Python type: bool
In Python, we have the keywords True and False
Could be represented by a single bit in memory...
... but since memory is addressed in bytes (or words of multiple bytes), usually represented as an int with 0 meaning Fa
fragile]Boolean values
The if statement takes a boolean value as its condition:
if x ¿ 10: print(x)
Variables can also store boolean values:
result = (x ¿ 10) result now stores True or False if result: print(x)
The "None" value
Python has a special value None which can be used to denote the "absence" of any other value
Python type: NoneType
fragile]Checking types in Python
Call type() to check the type of a variable or value
Note that type() returns a value of type type
You can use these type values like any other value, e.g.
if type(x) == int: print("x has type int") elif type(x) == type(y): print("x and y have the same type")