

National University of Singapore

Faculty of Law (2024-2025)

LC2017: Law and Technology

Tutorial (Week 3): Python – Strings, Booleans, Variables, IO

This tutorial is an in-class Python coding exercise. Your participation in the tutorial will make up your class performance grades comprising 15% of your overall grades for this course.

Write Python 3.12.x+ code for all the following, except for questions marked with an *, which are optional.

You are encouraged to run your code in script mode (rather than in interactive mode).

1. Store the following string into the variable `testimony`:

"He's owing me \$10 dollars each year for every \$100 dollars he borrowed for the past year," said witness PW-1.

- (a) Using the variable `testimony` and the Python string `replace` method, update the variable by replacing "10" with the word "ten" and "100" with the words "one hundred". Verify that the update is done correctly by printing the revised variable `testimony`.

- (b) Add the following string (in all upper case) as a new line *before* the contents of the revised variable `testimony` in (a) and store this as `full_testimony`.

the witness' testimony in court on 1 April 2022:

- (c) Verify that this is done correctly by printing `full_testimony`.

Hints:

- ① Note that the string in (a) starts (and uses) double quotes ". It also uses single quotes ' as the apostrophe.
 - ① Use the string `replace()` method which has the syntax `str.replace(old, new)` where `str` is the string or variable that you want to change, `old` is the substring you want to replace, and `new` is the replacement substring.
 - ① Use an escape code to represent a new line. See Lecture 4B for the table of escape codes.
2. You are given a string in the variable `info` that combines the casename with its case citation and court level. Every such string is in the format (the square brackets [*with italicized phrase*] are placeholders):

[*casename*], [*citation*] ([*court*])

The following is a sample of three such strings:

CBS Songs Ltd v Amstrad Consumer Electronics Plc, [1988] UKhl 15 (House of Lords)

Dow Jones v Gutnick, [2002] Hca 56 (High Court of Australia)

RecordTV Pte Ltd v MediaCorp TV Singapore Pte Ltd, [2010] SgCA 43 (Singapore Court of Appeal)

- (a) Using the variable `info` and the Python string `find` method, locate the ending index position for the casename substring and save it in the variable `casename_end`.

LC2017: Law and Technology

Tutorial (Week 3): Python – Strings, Booleans, Variables, IO

- (b) Using the ending index position for the casename, extract the casename substring in the variable `info`, save it as `casename` and print it.
- (c) As above, extract and save the citation (after converting it to uppercase) and court substrings in the variable `info` into the variables `citation` and `court`.
- *(d) Using (a), (b) and (c), generate a customized input prompt (using the Python input function) that reads:

Use this citation "[citation]" for [casename], decided by the [court]? Enter Y/N:

For instance, for *Gutnick*, the input prompt should be:

Use this citation "[2002] HCA 56" for Dow Jones v Gutnick, decided by the High Court of Australia? Enter Y/N:

- (e) If the `info` string is as follows, will your code be able to extract the citation, casename and court level? Explain why/why not.

Cartoon Network LP, LLLP v CSC Holdings, Inc., 536 F 3d 121 (2nd Cir, 2008)

- *(f) Based on your explanation in (e), write a script to extract the citation, casename and court level for such citations, in addition to the previously-listed citations.

Hints:

- ① Make no assumptions about a fixed index position for the start and end of a particular substring; instead, use code to locate the correct index positions of that substring.
- ① Use the Python `find` method to locate, from *left* to right, the correct index position for the particular substring you are looking for in the string `info`.
- ① Use the Python `rfind` method to locate, from *right* to left, the correct index position for the particular substring you are looking for in the string `info`.
- ① Remember that for the Python slicing operation, the end index is always one position after the last character of the sliced substring.
- ① Use the Python `len` method to get the length of the string. The index position of the last character of the string is always `len() - 1`. For instance:

```
s = "Hello"
print(len(s))
5
print(s[4])
o
```

- ① Sometimes, negative indices are easier to use if the substring is to the end/right of the string.
- ① Use the Python `fstring` function to build a complex string from different substrings.

3. Express the following legal rules in the Copyright Act 2021 as Boolean expressions, given the supplied variables (and their indicated types).

For example, under s 78, a person is a “qualified person” (`qualified_person`) (variable) if the person is a “qualified individual” (`qualified_individual`) (Boolean) or a body corporate incorporated in Singapore under any written law (`where_body_corporate_incorporated`) (string). The rule will be written as:

```
qualified_person = qualified_individual == True or
where_body_corporate_incorporated == "Singapore"
```

LC2017: Law and Technology

Tutorial (Week 3): Python – Strings, Booleans, Variables, IO

- (a) Under s 9, an “authorial work” (`is_authorial_work`) (variable) is a literary (`is_literary_work`) (Boolean), dramatic (`is_dramatic_work`) (Boolean), musical (`is_musical_work`) (Boolean) or artistic work (`is_artistic_work`) (Boolean).
- (b) Under s 10, an authorial work is a “work of joint authorship” (`is_joint_authorship_work`) (variable) if it is produced by the collaboration (`is_collaboration`) (Boolean) of 2 or more authors (`no_authors`) (integer) and the contributions of the authors are not separate (`separate_authors_contributions`) (Boolean).
- (c) Under s 122(2), for a sound recording first published on or before 31 December 2022 (`first_published_year`) (integer), its copyright (`has_copyright`) (variable) expires 70 years after the end of the year in which the recording is first published. (You will also need the `current_year` (integer).)

Hints:

- ① Use the comparison operators to evaluate numbers and strings; the result is a Boolean expression. For instance, to test if there are 2 or more authors, use the expression:

```
no_authors >= 2
```

- ① Use the Boolean operators `and` and `or` to combine Boolean variables and other Boolean expressions.
- ① To test if a variable or expression is True or False, use the comparison operator `==`.
- ① In Python, you can simplify the expression if it is evaluated to be True or False by skipping the comparison with True. In other words:

```
qualified_individual == True
```

can be simply written as:

```
qualified_individual
```

And the rule for `qualified_person` can be simply written as:

```
qualified_person = qualified_individual or  
where_body_corporate_incorporated == "Singapore"
```

4. The new provision in s 376E of the Penal Code is defined as follows:

Sexual grooming of minor below 16 years of age

376E.—(1) Any person of or above 18 years of age (A) shall be guilty of an offence if having met or communicated with another person (B) on at least one previous occasion —

(a) A intentionally meets B or travels with the intention of meeting B or B travels to attend a meeting with A which A has either initiated or agreed to whether expressly or by implication; and

(b) at the time of the acts referred to in paragraph (a) —

(i) A intends to do anything to or in respect of B, during or after the meeting, which if done will involve the commission by A of a relevant offence;

(ii) B is below 16 years of age; and

(iii) A does not reasonably believe that B is of or above 16 years of age.

- (a) Express the rule in s 376E(1) (without referencing the sub-subsections (a) and (b)) as a Boolean expression using the following pre-defined variables:

LC2017: Law and Technology

Tutorial (Week 3): Python – Strings, Booleans, Variables, IO

A_age (integer)

has_met (“having met” - Boolean)

has_communicated (“having communicated” - Boolean)

prev_occasion (integer)

- (b) Express the rule in s 376E(1)(a) as a Boolean expression using the following pre-defined variables:

A_intentionally_meets_B (Boolean)

A_travels_with_intent_to_meet_B (Boolean)

B_travels_to_meet_A (Boolean)

A_initiated_meeting (Boolean)

A_agreed_to_meet_B (Boolean)

- (c) Express the rule in s 376E(1)(b) as a Boolean expression using the following pre-defined variables:

s 376E(1)(b)(i):

A_intends_to_do_anything_to_B (Boolean)

involve_A_committing_relevant_offence (Boolean)

s 376E(1)(b)(ii):

B_age (integer)

s 376E(1)(b)(iii):

A_reasonable_belief_B_is_16_or_more (Boolean)

- (d) Combining (a), (b) and (c), express the rule in s 376E(1) as a Boolean expression.

Hints:

- ① This question may be continued in tutorials in Week 4.
- ① Note that Boolean operators have operator precedence. The Boolean operator precedence from high to low is not, and and or. See Lecture 4A for the table of operator precedence.
- ① Test the Boolean expressions for their correctness before combining them with other Boolean expressions. Evaluate Boolean expressions by setting the Boolean variables to True or False. For instance, set has_met to True and has_communicated to False to test if the Boolean expression that combines has_met and has_communicated is True:

has_met or has_communicated

You are strongly encouraged to work on these exercises ahead of tutorials. And you may discuss techniques and approaches, including exchanging hints, with each other. The temptation to get another student's answer or search for answers online or use AI-powered tools is there but do remember that you will learn best by thinking and working through the exercises yourself.

Have fun coding!

A/P Daniel Seng

July 28, 2024