CS1001.py

Recitation 1

28.2-4.3.2013

Python general comments

1. Course site at http://tau-cs1001-py.wikidot.com
2. Programming language -> Interpreter -> Machine language
3. IDLE (editor + interpreter), see site for installation instructions
4. Interactive mode vs. Script mode
5. Python version 3.2

Variables, types

- int - integers: . . . , -3, -2, -1, 0, 1, 2, 3, . . .
- float - floating point numbers, decimal point fractions: -3.2, 1.5, 1e-8, 3.2e5
- str - character strings, text: “intro2CS”, ‘python’

Operators

Addition:

Subtraction:

Multiplication:

Division - a bit special:

Power:

String concatenation using +:

String duplication using *:

Strings vs. numbers:

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-37-f945f8c7e111> in <module>()
----> 1 "4" + 5

TypeError: cannot concatenate 'str' and 'int' objects
```

Conversions

Use the functions `int()`, `float()`, and `str()` to convert between types (we will talk about *functions* next time):

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-41-91097a4105a2> in <module>()
----> 1 int("a")

ValueError: invalid literal for int() with base 10: 'a'
```

Flow control

Conditional statements

The `if` condition formula - replace conditions and statements with meaningful code:

```
if *condition*:
    *statement*
    *statement*
    ...
elif *condition*: # 0 or more elif clauses
    *statement*
    *statement*
    ...
else:             # optional
    *statement*
    *statement*
```

Example:

```
Shvizut Yom Alef
Lecture in intro to CS!
```

Loops

While

```
while *condition*:
    *statement*
    *statement*
```

Example - count how many times 0 appears in an integer number:

```
126765060022822940149670320537 6
6
```

For

```
for *variable* in *iterable*:
    *statement*
    *statement*
```

Example - solve the same problem with a `str` type instead of `int`:

```
6
```

Builtin solution

```
6
```

Efficiency We can measure which solution is faster:

```
100000 loops, best of 3: 11.6 us per loop
```

```
100000 loops, best of 3: 4.08 us per loop
```

```
1000000 loops, best of 3: 1.33 us per loop
```

The builtin solution is 4 times faster than the `for` solution which is 3 times faster than the `while` solution.

Other notes

- The `while` solution will not work for `num <= 0`
- The `while` solution will not work for non-numerals (e.g, `num = "Cola 0 is awesome!"`)
- The builtin solution is implemented with C and that is why it is faster

Fin

The notebook was written using Python 3.2 and IPython 0.13.1.

The code is available at https://raw.github.com/yoavram/CS1001.py/master/recitation1.ipynb.

The notebook can be viewed online at http://nbviewer.ipython.org/urls/raw.github.com/yoavram/CS1001.py/master/recitation1.ipynb.

The notebooks is also available as a PDF at https://github.com/yoavram/CS1001.py/blob/master/recitation1.pdf?raw=true.

This notebook is part of the **Extended introduction to computer science** course at Tel-Aviv University.