4 Objects

Key terms: expression `print bool int float str`

Reading: Severance 2

Exercise: Write a program that indicates, true or false, whether the current year is an even number.

We here review Python's basic object types, similar to those in other languages. `str`s (strings) contain text, broadly similar to its analogue in other languages. Literal strings can be delimited by either single quotes, `'`, double quotes, `"`, or triple quotes, `'''` or `"""`. Strings are not byte encodings; they are encoded in Unicode and can contain arbitrary linguistic content of all languages. The default string encoding for input and output is UTF-8, in which characters are represented with varying numbers of bytes.

`bool`s (Booleans) each have either the value `True` or `False`; `int`s (integers) are of unbounded size; these objects grow as large as needed on demand, and cannot be overflowed; `float`s (floating point numbers) have variable precision and come with certain complications.[9]

All these types are immutable objects and should not be called variables. Objects may be literals (e.g. `4`) or named references (e.g. `seconds_elapsed`). A new object (an instance of these classes) can be instantiated at any time by assigning something to a new name: `z_meters = 45`. They can also be created on the fly using their class name (e.g., `int(2.3)`). This construct does not "cast" or "typecast" as other languages call it, but just creates new anonymous objects as needed.

It is important to choose good names for objects. The best practice is to use names which are descriptive and accurate; this can be surprisingly tricky. Avoid repeating class names like `list` and `str`, or module names like `sys` and `math`, because doing so clobbers the existing references to those resources. Keywords such as `return` and `not` are also reserved and unavailable for use in naming.

With more than one object at hand, we might seek to combine them in a compound expression with its own value. For example, we can add and subtract numbers (`cash_balance-expenses`) or compare them (`hours_out!=hours_back`, `speed_mph>35`). These expressions call functions behind the scenes that return some desired value. Comparison operators raise `TypeError` when they can't handle the types passed (inquire whether `4>'4'`). Nonetheless, strings can be multiplied by integers: `5*'z'` is exactly what you'd think it is.

```
# Determine the percentage of U.S. counties that are in
    California:
counties_state = 58
counties_country = 3144
county_share = counties_state / counties_country
county_share_percent = 100 * county_share

print("This percentage of U.S. counties are in California:",
    county_share_percent)
```

Numbers support the combined assignment operators `+=`, `*=`, etc.; the famous `++` was pointedly left out because `+=1` is more consistent. The immutability of these kinds of objects means that if we change their value, we are really throwing away the old object and then using the same name for a new one.

Next, we will address mathematical resources available to your programs.

[9] `https://docs.python.org/3.6/tutorial/floatingpoint.html`