**Python** 5-8 Jan 2021

*(Learning from Radix Systems’ Python course, accompanying notes to course manual)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Topic** |  |  | | |
| Python2, Python3 | * Python3 has less features than Python2 (e.g. methods/functions) * Python3 is not backward compatible |  | | |
| Alternative Python implementations | * IronPython * Python running on .NET * Able to use .NET classes |  | | |
| Pypi | * Repository of software for the Python programming language * *snmp library*: network management |  | | |
| IDEs | * Spyder (looks like RStudio) |  | | |
| Python interactive shell/mode | Useful for   * Checking out features of a language without writing a program * Documentation (e.g. help(math))   Can check   * What’s inside namespace (e.g. dir()) |  | | |
| Namespace |  |  | | |
| Strings | * Sequences of unicode characters * Not bytes (like in C) * 1 chinese character = 3 bytes |  | | |
| Generator functions | * Those with keyword yield in them * Allows declaration of a function that behaves like an iterator class   Following from <https://stackoverflow.com/questions/102535/what-can-you-use-python-generator-functions-for>:   * Think of generators as returning multiple items, as if returning a list * But instead of returning items all at once, they return them one-by-one * Generator function is paused until next item is requested * Good for:   + Calculating large sets of results (esp. when calculation involves loops)   + When you don’t necessary need all the results   + When you don’t want to allocate memory for all results at the same time |  | | |
| Operator overloading | * Assigning a different meaning to an operator * Operators used in Python: <https://docs.python.org/3/library/operator.html#mapping-operators-to-functions> * Override function/method, use operator to test changes * Note: you cannot define your own operators 🙁 | **Operation** | **Syntax** (operator) | **Function** |
| Addition | a **+** b | add(a, b) |
| Concatenation | seq1 + seq2 | concat(seq1, seq2) |
| Division | a \ b | truediv(a, b) |
| Floor division | a [\\](file:///\\) b | floordiv(a, b) |
| Exponentiation | a \*\* b | pow(a, b) |
| Left shift | a << b | lshift(a, b) |
| Matrix multiplication | a @ b | matmul(a, b) |
| Right shift | a >> b | rshift(a, b) |
| Modulo | a % b | mod(a, b) |
| Ordering | a < b | lt(a, b) |
| Ordering | a <= b | le(a, b) |
| Difference | a != b | ne(a, b) |
| Equality | a == b | eq(a, b) |
| Bitwise Exclusive Or | a ^ b | xor(a, b) |
| String formatting | Helpful links   * <https://www.python.org/dev/peps/pep-0498/> * [**https://mkaz.blog/code/python-string-format-cookbook/**](https://mkaz.blog/code/python-string-format-cookbook/) | **Format** | **Description** | |
| **{:.2f}** | Float (2 d.p.) | |
| **{:+.2f}** | Float (2 d.p. with sign) | |
| **{:.2f}** | Float (no d.p.) | |
| **{:0>2d}** | Pad number with zeroes on left (s.t. integer width 2) | |
| **{:x<4d}** | Pad number with x’s on right (s.t. integer with 4) | |
| **{:.2%}** | Format percentage | |
| **{:.2e}** | Exponent notation (e.g. 1.00e+09) | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |
| Unicode | * A character set * A list of characters with unique numbers * (Character -> number) | **Example**: in Unicode character set, number for  A  is  41 | | |
| UTF-8 | * Unicode Transformation Format * An encoding * An algorithm that translates a list of numbers to binary so that it can be stored on disk * (Number -> binary) | **Example**: UTF-8 translates  1 2 3 4  to  00000001 00000010 00000011 00000100 | | |
| Unicode and UTF-8 | * <https://web.archive.org/web/20190710063815/http://www.polylab.dk/utf8-vs-unicode.html>   Flow   1. Application reads binary from disk 2. Application knows data represents Unicode string encoded with UTF-8 3. Convert binary data to numbers (use UTF-8 to decode) 4. Translate each number to a corresponding character (use Unicode character set) |  | | |
| Pylint | * Follows Python style guide * Standard recommendation | pylint hello.py | | |
| vi | * Tab-key means 4 spaces * :set expandtab * Tab-key means \t character * :setnoexpandtab |  | | |
| krita | Profession free and open-source painting program |  | | |

|  |  |  |
| --- | --- | --- |
| **Useful commands** |  | **Example** |
| help(...) |  | help(math) # modules  A = 2; help(A) # obj/var |
| **dir() \*\*\*** | List what objects are currently in my namespace   * Run in interactive shell * E.g. variables declared, imported modules |  |
| dir(...) | List attributes of an object   * Introspection | # attributes of a module (..., \_\_name\_\_, ..., log, log10, ..., sin, ...)  dir(math) |
| print(..., ...) | Concat and print multiple strings/objects   * Default joining delimeter: single whitespace | # output “x is 16”  print(“x is”, x) |
| print(..., end= “...”) | Specify default character(s) to print at the end of string   * Default: newline (\n) |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| #!/usr/bin/python3 | Tell what interpreter to use to run script   * **To place as the first line in script** * In Linux (Windows uses file extension to run file) * Default interpreter (if no specified): bash shell | # Command to run program  ./hello.py # no need python3 hello.py |
| chmod +x hello.py | Make program executable   * In Linux |  |
| hexdump file.txt | Display content of binary file in dexadecimal |  |