

Python notes



Old python logo, [wikipedia]



Current python logo, [wikipedia]

Python was started in 1989 by Guido van Rossum.

Functions

- Use `global` keyword to define a global variable (in the global scope)
 - we need this when we assign a value to a global variable
- `def fun(arg1, arg2="default", *args, **kwargs)`
- Recursion happens when a function calls itself
- `lambda x: int(x)`, a lambda function is used to write a small inline function definition

Handling exceptions

- `finally` block will always work no matter what happens in `try` or `except` blocks.
- `except` accept a tuple of exceptions or a single exception
- We can use `as` keyword to access the exception as a variable
- The keyword `raise` is use to raise exceptions

```
import random
def main():
    try:
        1 // random.randint(0, 1)
        print("ok")
        return
    except (ValueError, ZeroDivisionError):
```

```
print("custom error message")
except Exception as e:
    print(e)
finally:
    print("this will always be printed")
main()
```

Working with files

- `__file__` is the path of the current python file, so a file that references this variable will return its path
- Paths can be relative (to a directory/folder) or absolute (to the containing hard drive partition)
- Built-in library `os.path` is used to deal with paths and manage files
- Built-in function `open` is used to open a file
 - We provide the file path as the first argument
 - The second argument is the opening mode (read `r`, write `w`, read/write `r+`, ...)
 - Useful file methods: `read, write, readlines, writelines, close, seek, truncate, tell,`
 - Looping over a file is looping over its lines but with the `'\n'` at the end
 - `for line in f`
 -
- We can use the keyword `with` to auto-close the file when the block of code exists.
- If you tried to open many files an exception of type `OSError` will be raised

Modules and importing

```
import module as m
from module import * # import all, not recommended
```

```
from module import function as f, variable
```

Miscellaneous

- `all, any` the first check if all are truthy and the later checks if at least one is truthy
- `bin` print the binary representation
- `range, enumerate` are generators, which is more efficient
- `slice` is a function used to create slices, this is helpful when the slice can be passed as an argument or a variable
- `map, filter, reduce` are used to handle lists and iterables
- `dir, help` are used to get more information about the code (e.g. `help(str)`)
- `assert` is a function to assert some thing is True and it raises `AssertionError` with optional custom message when the value is False

```
assert something_is_true, "optional message"
```