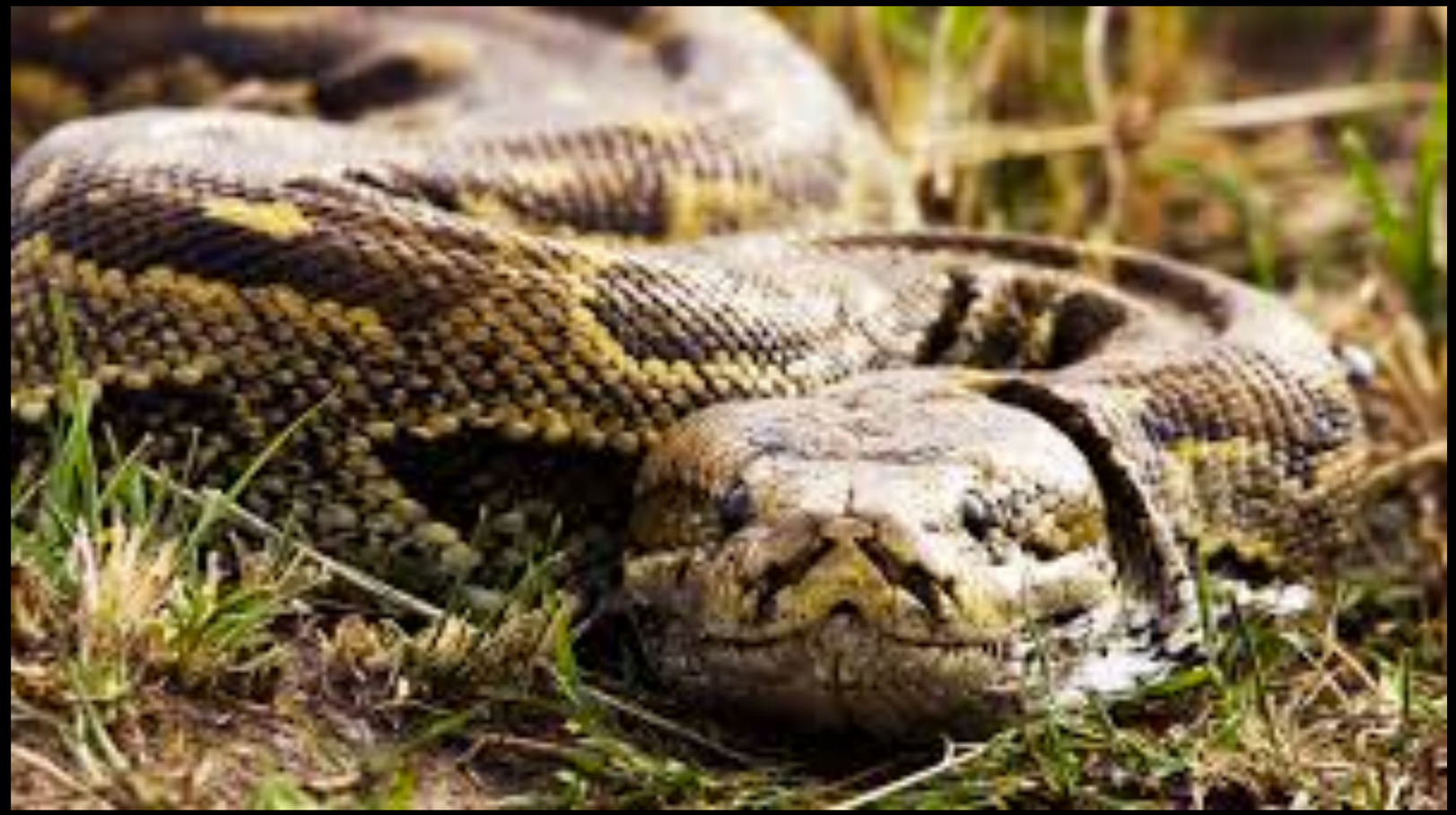
Monty Python’s Flying Circus

What’s Python?



Introduction

- Created by Guido van Rossum in 1991
- First released in 1991
- Named after the BBC show *Monty Python's Flying Circus*

Python Family

- CPython - Reference implementation, write in C/Python
- Cython - C-like compiled language, compiles Python to C/C++
- PyPy - Just-in-time compiler, usually faster than CPython
- Jython/RPython/Pyjs/IronPython, etc.

Python(CPython)

- High-level
- “Interpreted”
- Dynamically typed
- Garbage-collected
- Multiple programming paradigms

Python Interactive Shell

```
Python 3.7.4 (default, Aug 14 2019, 15:07:38)
[Clang 10.0.1 (clang-1001.0.46.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World!')
Hello World!
>>> 2 + 2
4
>>> 50 - 5*6
20
>>> 2 ** 3
8
>>> 5 // 2
2
>>> [1, 2, 3]
[1, 2, 3]
>>>
```

help

```
>>> help(help)

Help on _Helper in module _sitebuiltins object:

class _Helper(builtins.object)
 |  Define the builtin 'help'.
 |
 |  This is a wrapper around pydoc.help that provides a helpful message
 |  when 'help' is typed at the Python interactive prompt.
 |
 |  Calling help() at the Python prompt starts an interactive help session.
 |  Calling help(thing) prints help for the python object 'thing'.
 |
 |  Methods defined here:
 |
 |  __call__(self, *args, **kwds)
 |      Call self as a function.
 |
```

dir

```
>>> greeting = 'Hello World'

>>> dir(greeting)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__',
'__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count',
'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit',
'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans',
'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',
'translate', 'upper', 'zfill']
>>>
```

type

```
>>> greeting = 'Hello World'
>>> type(greeting)
<class 'str'>
>>>
```

Data Types

- Numbers
- Strings
- Lists
- Dicts

```
# Numbers
2 + 2


# Strings
'Hello World'

# Lists
[1, 2, 'a'] + [3, 'b', 5]
a_list = [1, 2, 3, 'a', 'b', 4, 5]

# List Comprehensions
[x * 2 for x in a_list if isinstance(x, int)]

# Dicts
tel = {'jack': 4098, 'sape': 4139}
tel = dict(jack=4098, sape=4139)

{x: x * 2 for x in [1, 2, 3]}

for key, value in tel.items():
    print(key, value)
```

Functions

```
Python 3.7.4 (default, Aug 14 2019, 15:07:38)
[Clang 10.0.1 (clang-1001.0.46.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> def foo(name):
...     print('hello', name)
...
>>> foo('Jimmy')
hello Jimmy
>>> foo
<function foo at 0x1012bf0e0>
>>>
```

Python Built-in Functions

```
>>> import builtins
>>> dir(builtins)
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'BlockingIOError', 'BrokenPipeError', 'BufferError',
'BytesWarning', 'ChildProcessError', 'ConnectionAbortedError', 'ConnectionError', 'ConnectionRefusedError',
'ConnectionResetError', 'DeprecationWarning', 'EOFError', 'Ellipsis', 'EnvironmentError', 'Exception', 'False', 'FileExistsError',
'FileNotFoundError', 'FloatingPointError', 'FutureWarning', 'GeneratorExit', 'IOError', 'ImportError', 'ImportWarning',
'IndentationError', 'IndexError', 'InterruptedError', 'IsADirectoryError', 'KeyError', 'KeyboardInterrupt', 'LookupError',
'MemoryError', 'ModuleNotFoundError', 'NameError', 'None', 'NotADirectoryError', 'NotImplemented', 'NotImplementedError',
'OSError', 'OverflowError', 'PendingDeprecationWarning', 'PermissionError', 'ProcessLookupError', 'RecursionError',
'ReferenceError', 'ResourceWarning', 'RuntimeError', 'RuntimeWarning', 'StopAsyncIteration', 'StopIteration', 'SyntaxError',
'SyntaxWarning', 'SystemError', 'SystemExit', 'TabError', 'TimeoutError', 'True', 'TypeError', 'UnboundLocalError',
'UnicodeDecodeError', 'UnicodeEncodeError', 'UnicodeError', 'UnicodeTranslateError', 'UnicodeWarning', 'UserWarning',
'ValueError', 'Warning', 'ZeroDivisionError', '__build_class__', '__debug__', '__doc__', '__import__', '__loader__', '__name__',
'__package__', '__spec__', 'abs', 'all', 'any', 'ascii', 'bin', 'bool', 'breakpoint', 'bytearray', 'bytes', 'callable', 'chr', 'classmethod',
'compile', 'complex', 'copyright', 'credits', 'delattr', 'dict', 'dir', 'divmod', 'enumerate', 'eval', 'exec', 'exit', 'filter', 'float', 'format',
'frozenset', 'getattr', 'globals', 'hasattr', 'hash', 'help', 'hex', 'id', 'input', 'int', 'isinstance', 'issubclass', 'iter', 'len', 'license', 'list',
'locals', 'map', 'max', 'memoryview', 'min', 'next', 'object', 'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit', 'range', 'repr', 'reversed',
'round', 'set', 'setattr', 'slice', 'sorted', 'staticmethod', 'str', 'sum', 'super', 'tuple', 'type', 'vars', 'zip']
>>>
```

Built-in Functions

Iterators

```
>>> foo = 'abcdefg'
>>> foo_iter = iter(foo)
>>> foo_iter
<str_iterator object at 0x106f17820>
>>> next(foo_iter)
'a'
>>> next(foo_iter)
'b'
...
>>> next(foo_iter)
'g'
>>> next(foo_iter)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
StopIteration
>>>
```

Iterators

```
class Reverse():
    """Iterator for looping over a sequence backwards."""
    def __init__(self, data):
        self.data = data
        self.index = len(data)

    def __iter__(self):
        return self

    def __next__(self):
        if self.index == 0:
            raise StopIteration
        self.index = self.index - 1
        return self.data[self.index]
```

Generators

```
def reverse(data):
    for index in range(len(data)-1, -1, -1):
        yield data[index]


for char in reverse('golf'):
    print(char)
```

Decorator

```
@staticmethod
def foo():
    pass

# vs

def foo():
    pass


foo = staticmethod(foo)
```

Decorator

```
def calculate_time(func):
    def wrapper(*args, **kwargs):
        begin = time.time()
        func(*args, **kwargs)
        end = time.time()
        total_cost = end - begin
        print(f'Takes: {total_cost}')

    return wrapper


@calculate_time
def slow_task():
    time.sleep(2)
    print('A slow task')


slow_task()
```

Context Manager

```
with open('/tmp/tmp.txt', 'r') as f:
    lines = f.readlines()
    print(lines)
```

Assignment Expressions

```
# Avoid calling len() twice
a = [1, 2, 3]
if (n := len(a)) > 10:
    print(f"List is too long ({n} elements, expected <= 10)")


# Loop over fixed length blocks
f = open('/tmp/tmp.txt')
while (block := f.read(256)) != '':
    process(block)
```

Classes

```
>>> class Foo:
...     pass
...
>>>
>>> foo = Foo()
>>> foo
<__main__.Foo object at 0x10783d9d0>
>>> type(foo)
<class '__main__.Foo'>
>>> foo.__class__
<class '__main__.Foo'>
>>>
```

Dynamic Classes

```
>>> Foo = type('Foo', (), {})
>>> Foo
<class '__main__.Foo'>
>>> foo = Foo()
>>> foo
<__main__.Foo object at 0x10e9da820>
>>> foo.__class__
<class '__main__.Foo'>
>>>
```

Methods

```
class Foo:

    def foo(self):
        print(self)

    @staticmethod
    def unbound_method():
        print('unbound method')


f = Foo()
f.foo()
f.unbound_method()

Foo.foo(f)
```

FastAPI

```python
from fastapi import FastAPI


app = FastAPI()


@app.get("/")
def read_root():
    return {"Hello": "World"}


@app.get("/items/{item_id}")
def read_item(item_id: int, q: str = None):
    return {"item_id": item_id, "q": q}

# import requests
# result = requests.get('http://127.0.0.1:8000')
# result.json()
```

Python Resources

Major Python Website: https://www.python.org
Python Documentation: https://docs.python.org
Python Standard Library: https://docs.python.org/3/library/index.html#library-index
Python Language Reference: https://docs.python.org/3/reference/index.html#reference-index
Python Cookbook: https://code.activestate.com/recipes/langs/python/

Talk Python to me: https://talkpython.fm/

Python mailing list: python-list@python.org

Github: https://github.com/pallets/flask

Questions?

Thanks!