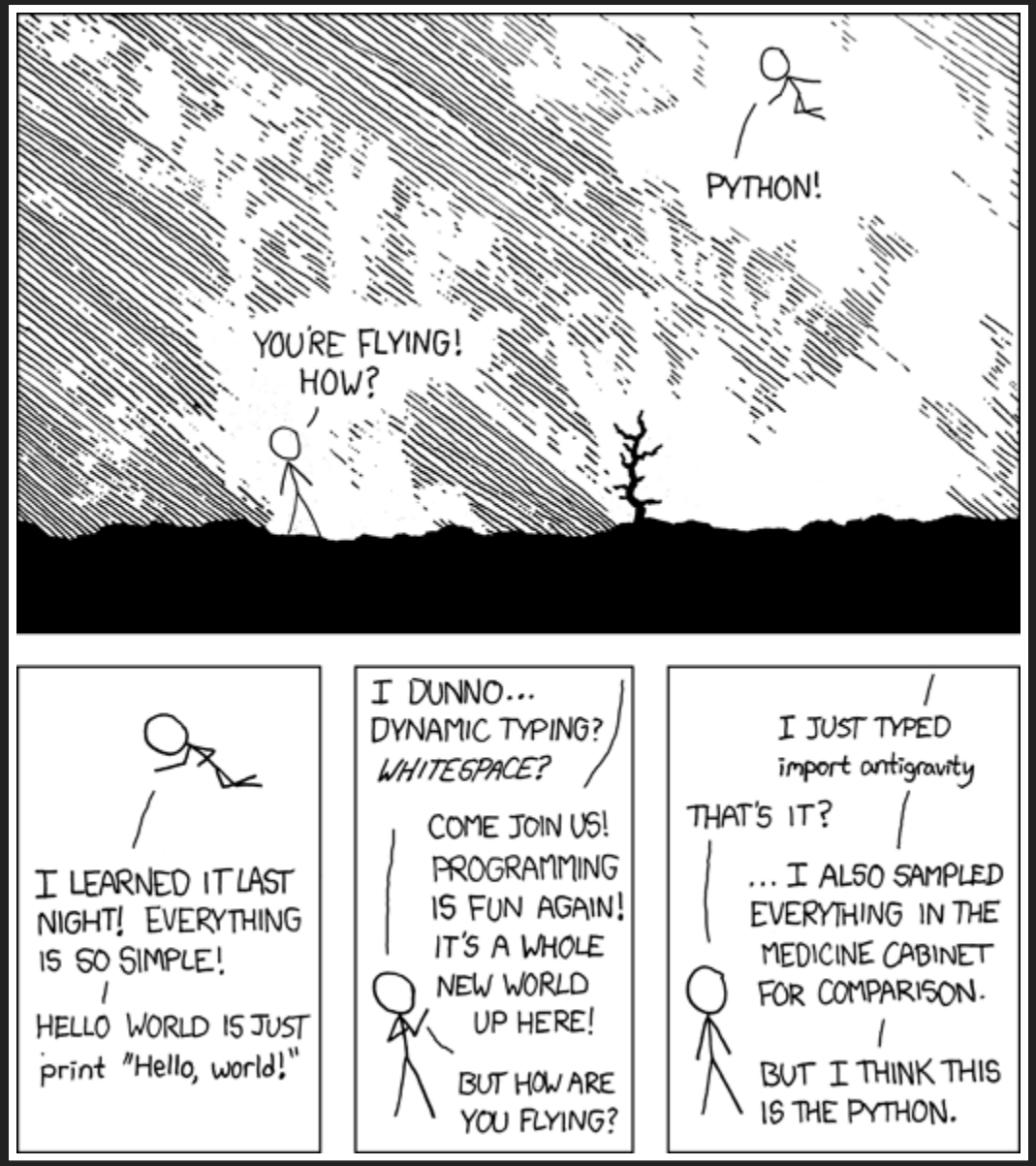
PYTHON

MAKING PROGRAMMING FUN AGAIN

Created by Chris Foster / @chrisfosterelli

PYTHON?

- Dynamic
- Interpreted
- Untyped
- White-space based
- High-level
- Extensible


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

WHAT CAN YOU DO WITH PYTHON?

- Command line scripts
- Desktop applications
- Webservers
- Computer Vision
- Artificial Intelligence
- Scientific processing
- ...whatever else you can think of!

PYTHON AS AN INTERPRETER

```
$ python
Python 3.4.0 (default, Apr 11 2014, 13:05:11)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more info
>>> a = 6
>>> a + 2
8
```

PYTHON AS A PROGRAM

```
$ python myscript.py
8
$
```

SIMPLE VARIABLES

```
>>> a = 10
>>> b = a + 10
>>> print(b)
20
```

```
>>> print(c)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'c' is not defined
```

```
>>> b = 'hello'
>>> print(b)
hello
```

LISTS

```
>>> mylist = []
>>> mylist.append(1)
>>> mylist.append(2)
>>> mylist.append(3)
>>> print(mylist[0])
1
>>> print(mylist[1])
2
>>> print(mylist[2])
3
```

```
>>> mylist = [1,2,3]
>>> print(mylist[10])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
```

DICTIONARIES

```
>>> phonebook = {}
>>> phonebook["John"] = 938477566
>>> phonebook["Jack"] = 938377264
>>> phonebook["Jill"] = 947662781
```

```
>>> phonebook = {
...     "John" : 938477566,
...     "Jack" : 938377264,
...     "Jill" : 947662781
... }
```

CASTING

```
>>> m = 'Your number is: '
>>> n = 10
>>> m + n
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Can't convert 'int' object to str implicitly
```

```
>>> m + str(10)
'Your number is: 10'
```

CONTROL FLOW - IF

```
>>> x = int(input("Please enter an integer: "))
Please enter an integer: 42
>>> if x < 0:
...     x = 0
...     print('Negative changed to zero')
... elif x == 0:
...     print('Zero')
... elif x == 1:
...     print('Single')
... else:
...     print('More')
...
More
```

CONTROL FLOW - WHILE LOOP

```
>>> a, b = 0, 1
>>> while b < 10:
...     print(b)
...     a, b = b, a+b
...
1
1
2
3
5
8
```

CONTROL FLOW - FOR LOOP

```
>>> words = ['cat', 'window', 'dog']
>>> for w in words:
...     print(w, len(w))
...
cat 3
window 6
dog 3
```

FUNCTIONS

```
>>> def fib(n):
...     a, b = 0, 1
...     while a < n:
...         print(a, end=' ')
...         a, b = b, a+b
...     print()
...
>>> fib(2000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

COMMENTS

```
>>> z = 10 # This is a single line comment
>>> def b():
...     """ This is a function's explanation comment """
...     return 10
...
```

WORKSHOP

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

MODULES

```
>>> import sys
>>> sys.ps1
'>>> '
```

```
>>> import sys
>>> sys.ps1
'>>> '
```

CUSTOM MODULES

```
def fib2(n): # return Fibonacci series up to n (fibo.py)
    result = []
    a, b = 0, 1
    while b < n:
        result.append(b)
        a, b = b, a+b
    return result
```

```
>>> import fibo
>>> fibo.fib2(100)
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

FILE INPUT/OUTPUT

```
>>> f = open('workfile', 'r')
>>> f.read()
'This is the entire file.\n'
>>> f.close()
```

```
>>> f = open('workfile', 'w')
>>> f.write('This is a test\n')
15
>>> f.close()
```

EXCEPTIONS/ERRORS

```
>>> while True:
...     try:
...         x = int(input('Please enter a number: '))
...         break
...     except ValueError:
...         print('Oops! Try again...')
...
```

```
>>> raise NameError('HiThere')
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
NameError: HiThere
```

CLASSES

```
>>> class Dog:
...     def __init__(self, name):
...         self.name = name
...         self.tricks = []
...     def add_trick(self, trick):
...         self.tricks.append(trick)
...
>>> d = Dog('Fido')
>>> d.add_trick('roll over')
>>> d.add_trick('play dead')
>>> d.tricks
['roll over', 'play dead']
```

WORKSHOP

Write a Ceasar Cipher decryptor in Python, to decrypt the following message:

`ESPBFTNVMCZHYQZIUFXAPOZGPCESPWLKJOZRD`

Note: The message has a shift of 11

LIKE PYTHON?

THE END

- TRUSU Computer Science Club (http://trucsclub.ca)
- Chris Foster (http://fosterelli.co)