python™

Leif Andersen

University of Utah

What is Python?

An interpreted, interactive,
object-oriented
programming language.

Reasons to use Python

Reasons to use Python

- Develop faster.

Reasons to use Python

- Develop faster.
- Write cleaner code.

Reasons to use Python

- Develop faster.
- Write cleaner code.
- Scientific Computing.

Reasons to use Python

Reasons NOT to use Python

Reasons NOT to use Python

- Can be slower.

Reasons NOT to use Python

- Can be slower.
- Scalability issues.

Reasons NOT to use Python

- Can be slower.
- Scalability issues.
- Library incompatibilities.

Compiled > Interpreted

Compiled $<$ Interpreted

Compiled = Interpreted

Hello World

Hello World!

```
public class Hello {
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

Hello World!

```
"Hello World!"
```

Hello World!

```
print("Hello World!")
```

REPL

Read

Eval

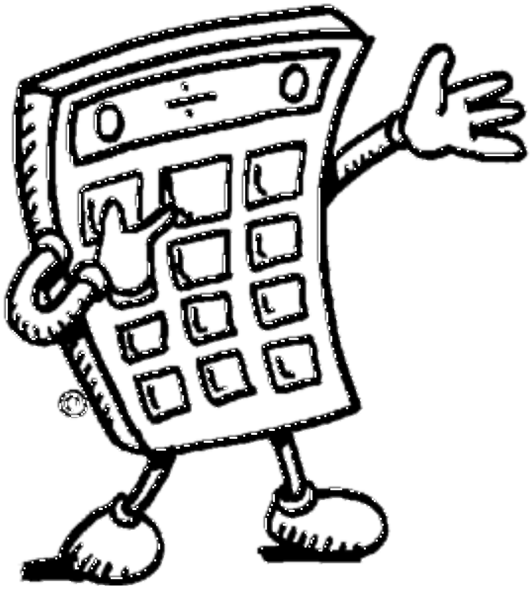
Print

Loop

Turning this:

Turning this:

Into This:



Arithmetic

```
<exp> + <exp>
<exp> - <exp>
<exp> * <exp>
<exp> / <exp>
<exp> // <exp>
<exp> ** <exp>
```

Arithmetic

```
5 + 3
3 - 1
4 * 5
2 / 9
1 // 2
2 ** 10
```

Comparators

```
<exp> == <exp>
<exp> != <exp>
<exp> > <exp>
<exp> < <exp>
<exp> >= <exp
<exp> <= <exp>
<exp> is <exp>
<exp> in <collection>
```

Comparators

```
1 == 3
5 != 3
18 > 3
5 < 5
134 >= 2
5 <= 14
8 != 3
5 is 5
5 in {1, 5, 10}
```

Boolean logic

```
True
False
```

Boolean logic

```
and
or
not
```

Boolean logic

```
not True
```

Boolean logic

```
not True and False or True and (True or False)
```

Boolean logic

```
((not True) and False) or (True and (True or False))
```

Variables

Variables

```
<var> = <val>
```

Variables

```
x = 2
```

Variables

```
x = 1 + 5
```

Variables

```
x = 5
y = str(x)
z = int(z)
a = float(z)
```

Lists

Lists

```
[<item0>, <item1>, ..., <itemn>]
```

Lists

```
x = [0, 1, 2, 3]
```

Lists

Length of a list

Lists

```
x = [1, 2, 3]
```

Lists

```
x = [1, 2, 3]
   len(x)
```

Lists

append to a list

Lists

```
x = [1, 2, 5
x.append(3)
```

Lists

```
x = [1, 2, 5
x.append(3)
print(x) # [1, 2, 5, 3]
```

Lists

List Indexing

Lists

```
x = ["red", "green", "blue"]
print(x[2]) # blue
```

Lists

```
x = ["red", "green", "blue"]
print(x[2]) # blue
x[1] = "yellow"
print(x) # ["red", "yellow", "blue"]
```

while loops

While Loops

```
while <condition>:
    <exp>
```

While Loops

```
while True:
    print("y")
```

While Loops

```
while 18 > 3:
    print("y")
```

While Loops

Note the Whitespace

While Loops

```
while 1 == 1:
    print("I'm in the loop.)"
print("I'm not in the loop.")
```

While Loops

A word on variable scope

While Loops

A word on variable scope
Loops do not create new scope

While Loops

While Loops

```
i=1
while i < 10:
    i = i + 1
    tmp = i
print(tmp) # 10
```

for loops

for loops

Like Java's for/each loop

for loops

```
for <var> in <collection>:
    <exp>
```

for loops

```
for i in [1, 2, 3, 4, 5]:
    print(i)
```

for loops

Creating Java style for loops

for loops

```
x = [1, 2, 3, 4, 5]
for i in len(range(x)):
    print(x[i])
```

for loops

```
x = [1, 2, 3, 4, 5]
for i in len(range(x)):
    print(x[i])
```

Don't do this!

(It's ugly)

functions

functions

```
def <name>(<args>):
    <body>
```

functions

```
def sum(a, b):
    return a + b
```

functions

A word on scope:

functions

A word on scope:

Functions DO make new scope.

functions

A bigger function

functions

```
def sum_range(a, b):
    sum = 0
    for i in range(a, b):
        sum += i
    return sum
```

sets

dicts

sets, dicts

```
x = {1, 2, 3}
```

sets, dicts

```
x = {1, 2, 3, 1}
```

sets, dicts

```
y = {"red": 5,
  "blue": 10,
  "orange": 42}
```

sets, dicts

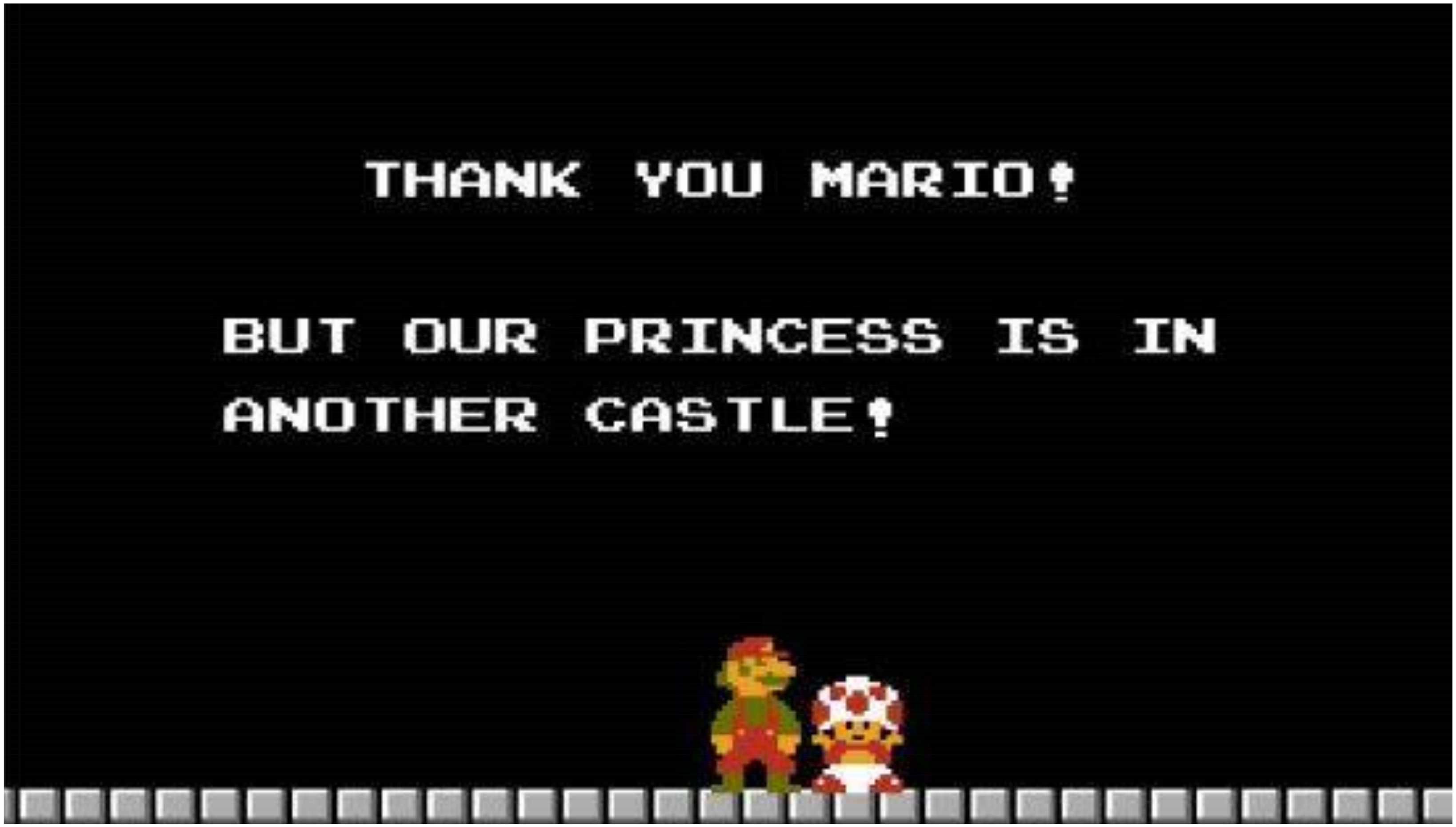
```
y = {"red": 5,
  "blue": 10,
  "orange": 42}
print(y["orange"]) # 42
y["yellow"] = 10
```

sets, dicts

{ }

sets, dicts

classes



THANK YOU MARIO!

BUT OUR PRINCESS IS IN
ANOTHER CASTLE!

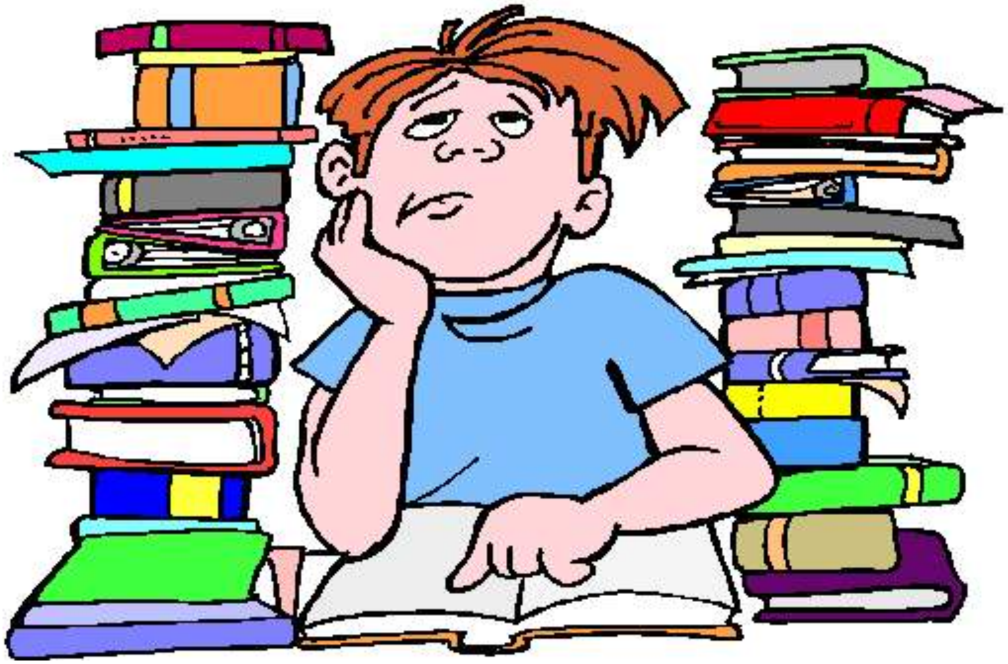
Running a Python file

```
python3 <filename>
```

```
python3 foo.py
```

Why `python3`

Homework



Homework

- Download Python

Homework

- Download Python
- Make factorial function

Homework

- Download Python
- Make factorial function
- Use python to solve some math problems.

Homework

- Download Python
- Make factorial function
- Use python to solve some math problems.
- Do something cool with python (pygame, numpy, etc.).

Homework

- Download Python
- Make factorial function
- Use python to solve some math problems.
- Do something cool with python (pygame, numpy, etc.).
- Fill out the survey

Python 2 vs. 3

```
print "text"
```

VS

```
print("text")
```

CADE

CADE

Use python3

Recommended Tools

Recommended Tools

- bpython3

Recommended Tools

- bpython3
- Pydev/Aptana Studio (Eclipse Environment

Recommended Tools

- bpython3
- Pydev/Aptana Studio (Eclipse Environment
- PDB (good with emacs)

Recommended Resources

Recommended Resources

- Python API

Recommended Resources

- Python API
- Stack Overflow

Recommended Resources

- Python API
- Stack Overflow
- Google

Questions?

Questions?

leif@leifandersen.net