

# Loops

## Python for Ecologists

Chance Pascale, Tom Purucker, Tao Hong

Ecological Society of America Workshop  
Portland, OR

`purucker.tom@gmail.com`

July 31, 2013

# Conditionals

- whitespace is very important, determines the close

```
if grade >= 90:  
    print "A"  
elif grade >= 80:  
    print "B"  
elif grade >= 70:  
    print "C"  
elif grade >= 60:  
    print "D"  
else:  
    print "F"
```

# Booleans

```
cloudy = True  
rainy = False
```

```
# Testing booleans
```

```
print( bool( "alpha"<"beta" ) )
```

```
True
```

- Strings are compared alphabetically when sorted

# Comparison operators

## ■ Python comparison operators

# <code>x == y</code>	<i>x equals y</i>
# <code>x &lt; y</code>	<i>x is less than y</i>
# <code>x &gt; y</code>	<i>x is greater than y</i>
# <code>x &gt;= y</code>	<i>x is greater than or equal to y</i>
# <code>x &lt;= y</code>	<i>x is less than or equal to y</i>
# <code>x != y</code>	<i>x is not equal to y</i>
# <code>x is y</code>	<i>x and y are the same object</i>
# <code>x is not y</code>	<i>x and y are different objects</i>
# <code>x in y</code>	<i>x is a member of the container y</i>
# <code>x not in y</code>	<i>x is not a member of the container</i>

# Loop iteration

*# through a list of numbers*

```
for value in [1,2,3,4,5,6,7]:  
    print value
```

*# using range*

```
for value in range(1,8):  
    print value
```

# enumeration

*# through a list of strings with indices*

```
watersheds = ["Suwanne", "Ocone", "Tennessee", "Flint"]
```

```
for index, value in enumerate(watersheds):  
    print index, value
```

*# can also loop through dictionaries*

## break, continue, and pass

*# using break*

```
for number in range(1,8):  
    if number < 5:  
        print number  
    else:  
        break
```

*# using continue*

```
for number in range(1,8):  
    if number < 5:  
        print number  
        continue
```

*# pass*

```
for number in range(1,8):  
    if number < 5:  
        print number  
    else:  
        pass
```