Conditionals

Darin Brezeale

The University of Texas at Arlington

Basic Concepts – Conditionals

Conditionals are used for making decisions.

The conditionals available in Python are `if`, `elif`, and `else` statements.

Operators – Relational

The following relational operators are available in Python:

==	equality
!=	not equal
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

Conditionals – `if`

The basic format of the `if` statement is

```
if condition_is_true :
    do_something
```

Examples

```
if x > 13:
    print "x is greater than 13"

if y == 20:
    a = 66
```

Conditionals – `if` cont.

Question: What if we want to do more than one thing in response to a condition being true?

Answer: Create a block of statements (the textbook calls this a suite).

Example

```
if x >= 2:
    y = 10
    print "y is now 10"
```

The two statements that follow the `if` statement are only executed if `x` is greater than or equal to 2.

Conditionals – `if` cont.

In Python, indentation of code has meaning.

Example

```
if x != 44:
    y = 72
    z = x * y

print "z is", z
```

The statement

```
print "z is", z
```

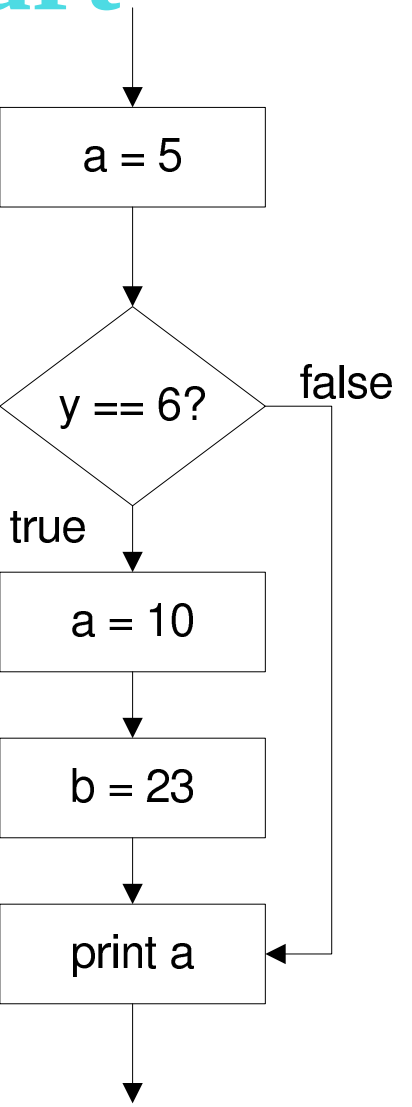
will be executed regardless of the value of x since it is at the same level of indention as the `if` statement.

`if` statement flow chart

```
a = 5

if y == 6:
    a = 10
    b = 23

print a
```



Conditionals – `if-else`

Sometimes we wish to do one thing if a condition is true but another if the condition is false. For this we can use a set of `if-else` statements:

```
if condition_is_true :
    do_something
else :
    do_something_else
```

Example

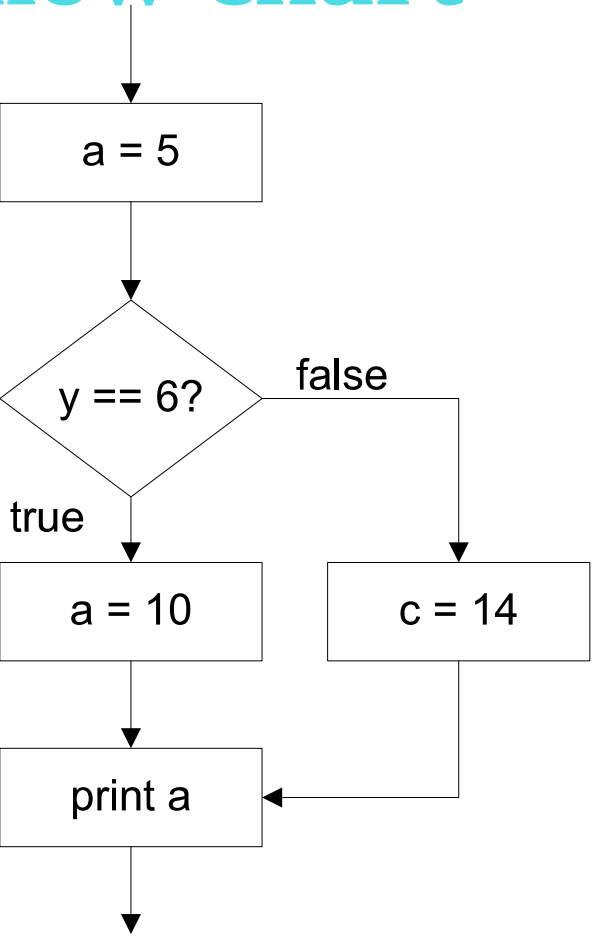
```
if x >= 2:
    print "x is greater than or equal to 2"
else:
    print "x is less than 2"
```

`if-else` statement flow chart

```
a = 5

if y == 6:
    a = 10
else:
    c = 14

print a
```



Conditionals – `elif`

Sometimes we need to ask multiple questions, but each additional question is only asked if the previous condition is false.

```
if condition1_is_true :
        do_something
elif condition2_is_true :
        do_something_else
elif condition3_is_true :
        do_a_third_thing
    .
    .
    .
```

For this we can use the `elif` statement (short for “else if”).

Conditionals – `elif` cont.

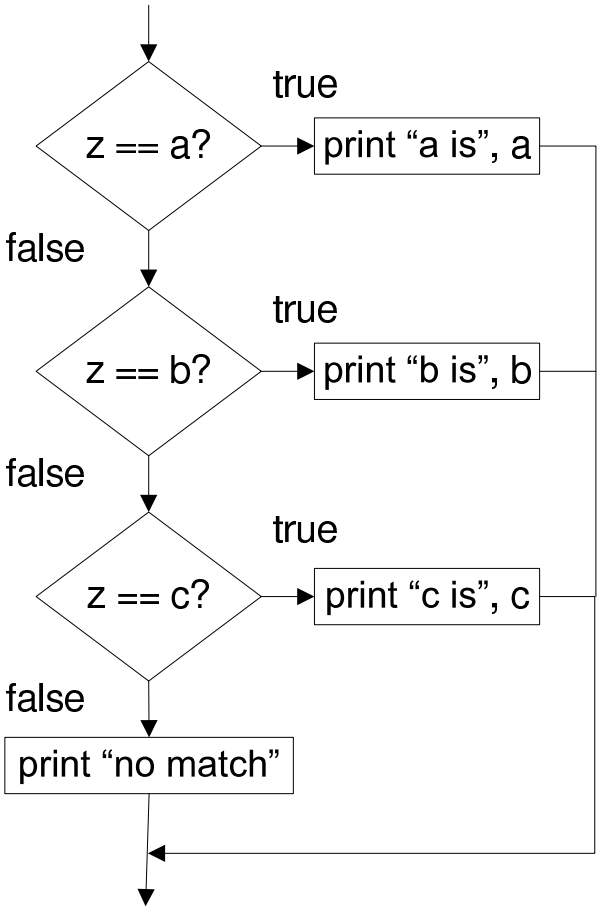
Example:

```
a = 10
b = 20
c = 30
z = 23

if z == a:
    print "a is", a
elif z == b:
    print "b is", b
elif z == c:
    print "c is", c
else:
    print "no match"
```

`if/elif` statement flow chart

```
if z == a:
    print "a is", a
elif z == b:
    print "b is", b
elif z == c:
    print "c is", c
else:
    print "no match"
```



Conditionals Notes

Keep in mind that any condition that evaluates to a nonzero value is considered true.

```
if 8:
    print "non-zero values are true"
else:
    print "this never prints"

if -3.4:
    print "non-zero values are true"
else:
    print "this never prints"

if 0:
    print "zero is false"
else:
    print "this is always false"
```

Conditionals Notes cont.

WARNING: Don't use `=` when you really mean `==`

`=` is used for assigning values

Example:

```
a = 5
```

`==` is used for determining if two values are equal

Example:

```
if a == 5:
```

The first gives `a` a value of 5; the second asks if `a` already has a value of 5.

Operators – Logical

Python has the following logical operators:

not

and

or