The standard if conditional statement follows this syntax:

```python
if expression:
    if_suite
```

If the expression is non-zero or True, the if_suite is executed. Then, execution continues on the first statement after the if block.

Suite is the term used in Python to refer to a sub-block of code and can consist of single or multiple statements.

Parentheses are not required in if statements as they are in other languages.
if expression:
    if_suite
else:
    else_suite
# Determine whether the user entered an even or odd number.
# Uses the modulus operator to determine whether a number is even or odd.

user_input = raw_input('Please enter a number: ')
number = int(user_input)

if (number % 2 == 0):
    print number, 'is an even number.'
else:
    print number, 'is an odd number.'
# Determine whether the user entered an even or odd number.
# Looks at the last character of the string to determine
# if it is even or odd.

# This program also uses \ to show how to continue a line. There is
# no space after the continuation character \.

number = raw_input('Please enter a number: ')
last_digit = number[-1]

if last_digit == '0' or last_digit == '2' or last_digit == '4' or last_digit == '6' or last_digit == '8':
    print number, 'is an even number.'
else:
    print number, 'is an odd number.'
Listing 3: odd-even-version3.py

```python
# Determine whether the user entered an even or odd number.
# Looks at the last character of the string to determine if it is even or odd.

number = raw_input('Please enter a number: ')
last_digit = number[-1]

if last_digit in ['0','2','4','6','8']:
    # uses a list for odd or even
    print number, 'is an even number.'
else:
    print number, 'is an odd number.'
```
# Determine whether the user entered an even or odd number.
# Looks at the last character of the string to determine if it is even or odd.

even_digits = ['0', '2', '4', '6', '8']  # create list of even digits
number = raw_input('Please enter a number: ')
last_digit = number[-1]

if last_digit in even_digits:
    print number, 'is an even number.'
else:
    print number, 'is an odd number.'
For the if/elif/else statement, you can have an unlimited number of elif expressions.

```python
if expression:
    if_suite
elif expression:
    elif_suite
else:
    else_suite
```
Listing 5: if-elif-else.py

```
# An example of how to use the if-elif-else construct.

score = int(raw_input('Please enter your score: '))

if score >= 90:
    letter = 'A'
elif score >= 80:
    letter = 'B'
elif score >= 70:
    letter = 'C'
elif score >= 60:
    letter = 'D'
else:
    letter = 'F'

print "Here's your letter grade: ", letter
```
Determining letter grade from a score (bad example with use of if-else)

Listing 6: if-else-bad-example.py

```python
# Using if-else constructs to display a letter grade.
# In this case, if-elif-else construct would be better.

score = int(raw_input('Please enter your score: '))

if score >= 90:
    letter = 'A'
else:
    if score >= 80:
        letter = 'B'
    else:
        if score >= 70:
            letter = 'C'
        else:
            if score >= 60:
                letter = 'D'
            else:
                letter = 'F'

print "Here’s your letter grade: ", letter
```
while expression:
    while_suite
Counting with while loops

Listing 7: counting-with-while-loop.py

```python
1  count = 5
2  while count == 5:
3      print 'the value of count is', count
4      count += 1
5  print 'Done with while loop!!'
```
# A game where the user guesses a secret number between 1 and 100.

```python
import random

solution = random.randint(1, 100)
guess = 0
attempts = 0

while guess != solution:
    attempts = attempts + 1
    user_input = raw_input('Enter a number between 1 and 100: ')
    guess = int(user_input)
    if guess == solution:
        print 'Congratulations! It took you %d guesses.' % (attempts)
    elif guess < solution:
        print 'HIGHER'
    else:
        print 'LOWER'
```

Tiffani L. Williams (Texas A&M)  CSCE 110: Basics of Python (Part 2)  Spring 2012  13 / 14
# The computer attempts to guess a number selected by the user.

```python
import random
response = 'whatever'
attempts = 0
min = 1
max = 100

print 'My name is Alice.'
print 'I will try to guess a secret number between 1 and 100.'
while response != 'c':
    attempts = attempts + 1
    guess = ((max - min) / 2) + min
    print 'Alice’s guess: ', guess
    response = raw_input('(c)orrect (h)igher (l)ower: ')
    if response == 'h':
        print 'HIGHER'
        min = guess + 1
    else:
        print 'LOWER'
        max = guess - 1
print 'CORRECT! It took %d attempts.' %(attempts)
```