

لنشغل البرنامج:

>> python pizza.py

وسنحصل على المخرج التالي:

Output

Pizza created: artichoke ($15)

Made 1 artichoke pizza(s)

Ate 1 pizza(s)

Pizza created: margherita ($12)

Made 2 margherita pizza(s)

Ate 1 pizza(s)

تسمح لنا تعليمات print برؤية أن البرنامج يعمل، ولكننا نستطيع أن نستخدم وحدة التسجيل لذات الغرض بدلا من ذلك.  
لنقم بإزالة تعليمات print من الشيفرة البرمجية، ونستورد الوحدة باستخدام الأمر import logging:

import logging

class Pizza():

def \_\_init\_\_(self, name, value):

self.name = name

self.value = value

...

المستوى التلقائي للتسجيل في وحدة التسجيل هو مستوى التحذير (WARNING)، وهو مستوى فوق مستوى التنقيح (DEBUG). بما أننا سنستخدم الوحدة بغرض التنقيح في هذا المثال، سنحتاج الى تعديل إعدادات التسجيل لتصبح بمستوى التنقيح logging.DEBUG بحيث تعود معلومات التنقيح لنا من خلال لوحة التحكم. ونقوم بإعداد ذلك بإضافة ما يلي بعد تعليماتة الاستيراد:

import logging

logging.basicConfig(level=logging.DEBUG)

class Pizza():

...

هذا المستوى المتمثل بlogging.DEBUG يشير لقيد رقمي قيمته 10.  
سنستبدل الآن جميع تعليمات print  بتعليمات logging.debug()، (logging.DEBUG ثابت بينما logging.debug() دالة). نستطيع أن نمرر لهذه الدالة نفس المدخلات النصية لتعليماتة print كما هو موجود بالأسفل:

import logging

logging.basicConfig(level=logging.DEBUG)

class Pizza():

def \_\_init\_\_(self, name, price):

self.name = name

self.price = price

logging.debug("Pizza created: {} (${})".format(self.name, self.price))

def make(self, quantity=1):

logging.debug("Made {} {} pizza(s)".format(quantity, self.name))

def eat(self, quantity=1):

logging.debug("Ate {} pizza(s)".format(quantity, self.name))

pizza\_01 = Pizza("artichoke", 15)

pizza\_01.make()

pizza\_01.eat()

pizza\_02 = Pizza("margherita", 12)

pizza\_02.make(2)

pizza\_02.eat()