# Polynomial

## Sample polynomial

In [1]:

```
def p(x):
    return 3*x**4 + 5*x**3 + 6*x**2 + x +9

for x in [-1,0,2,3.4]:
    print(x,p(x))
```

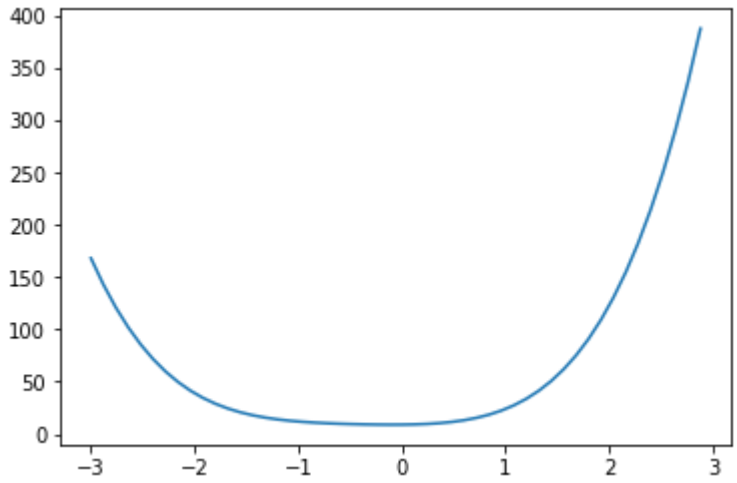
```
-1 12
0 9
2 123
3.4 679.1807999999999
```

## Implementation through a function

In [7]:

```
import numpy as np
import matplotlib.pyplot as plt

def p(x):
    return 3*x**4 + 5*x**3 + 6*x**2 + x +9

X=np.linspace(-3,3,50,endpoint=False)
print(X)
F=p(X)
print(F)
plt.plot(X,F)
plt.show()
```

```
[-3.   -2.88 -2.76 -2.64 -2.52 -2.4  -2.28 -2.16 -2.04 -1.92 -1.8  -1.68
 -1.56 -1.44 -1.32 -1.2  -1.08 -0.96 -0.84 -0.72 -0.6  -0.48 -0.36 -0.24
 -0.12  0.    0.12  0.24  0.36  0.48  0.6   0.72  0.84  0.96  1.08  1.2
  1.32  1.44  1.56  1.68  1.8   1.92  2.04  2.16  2.28  2.4   2.52  2.64
  2.76  2.88]
[168.         142.83825408 120.90620928 101.90485248  85.55010048
  71.5728      59.71872768  49.74859008  41.43802368  34.57759488
  28.9728      24.44406528  20.82674688  17.97113088  15.74243328
  14.0208      12.70130688  11.69395968  10.92369408  10.33037568
   9.8688       9.50869248   9.23470848   9.04643328   8.95838208
   9.           9.21566208   9.66467328  10.42126848  11.57461248
  13.2288      15.50285568  18.53073408  22.46131968  27.45842688
  33.7008      41.38211328  50.71097088  61.91090688  75.22038528
  90.8928     109.19647488 130.41466368 154.84555008 182.80224768
 214.6128     250.62018048 291.18229248 336.67196928 387.47697408]
```

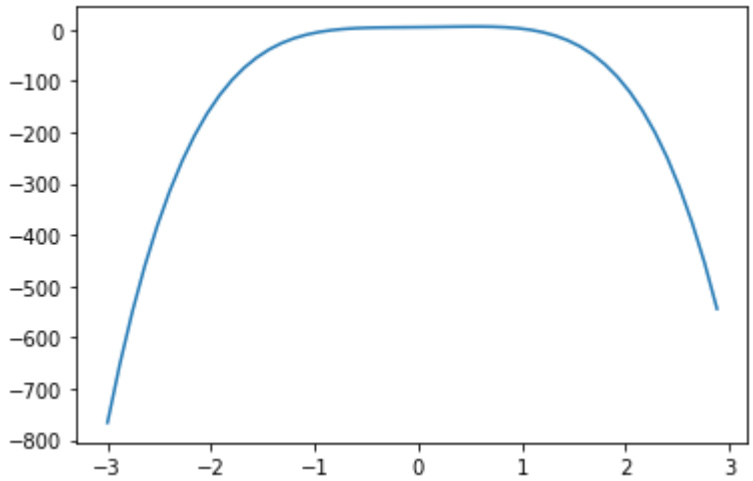



## Implemetation through a class

In [14]:

```
import numpy as np
import matplotlib.pyplot as plt

class Polynomial:
    def __init__(self,*coefficients):
        self.coefficients=coefficients[::-1] #tuple is also turned into list
#[::-1] flips the coefficients
    def __repr__(self):
        return "Polynomial" + str(self.coefficients[::-1])

    def __call__(self,x):
        res=0
        for index,coeff in enumerate(self.coefficients):
            res += coeff *x**index
        return res

p=Polynomial(-9,2,2,2,5)
for x in range(-3,3):
    print(x,"   ",p.__repr__(),"   ",p(x))

X=np.linspace(-3,3,50,endpoint=False)
print(X)
F=p(X)
print(F)
plt.plot(X,F)
plt.show()
```

```
-3     Polynomial(-9, 2, 2, 2, 5)     -766
-2     Polynomial(-9, 2, 2, 2, 5)     -151
-1     Polynomial(-9, 2, 2, 2, 5)     -6
0     Polynomial(-9, 2, 2, 2, 5)     5
1     Polynomial(-9, 2, 2, 2, 5)     2
2     Polynomial(-9, 2, 2, 2, 5)     -111
[-3.   -2.88 -2.76 -2.64 -2.52 -2.4  -2.28 -2.16 -2.04 -1.92 -1.8  -1.68
 -1.56 -1.44 -1.32 -1.2  -1.08 -0.96 -0.84 -0.72 -0.6  -0.48 -0.36 -0.24
 -0.12  0.    0.12  0.24  0.36  0.48  0.6   0.72  0.84  0.96  1.08  1.2
  1.32  1.44  1.56  1.68  1.8   1.92  2.04  2.16  2.28  2.4   2.52  2.64
  2.76  2.88]
[-7.66000000e+02 -6.51120586e+02 -5.49584420e+02 -4.60318205e+02
 -3.82293437e+02 -3.14526400e+02 -2.56078167e+02 -2.06054602e+02
 -1.63606359e+02 -1.27928881e+02 -9.82624000e+01 -7.38919398e+01
 -5.41473126e+01 -3.84031206e+01 -2.60787558e+01 -1.66384000e+01
 -9.59102464e+00 -4.49039104e+00 -9.35050240e-01  1.43165696e+00
  2.92160000e+00  3.80185856e+00  4.29472256e+00  4.57769216e+00
  4.78347776e+00  5.00000000e+00  5.27038976e+00  5.59298816e+00
  5.92134656e+00  6.16422656e+00  6.18560000e+00  5.80464896e+00
  4.79576576e+00  2.88855296e+00 -2.32176640e-01 -4.92640000e+00
 -1.15988838e+01 -2.06991846e+01 -3.27216486e+01 -4.82054118e+01
 -6.77344000e+01 -9.19373286e+01 -1.21487703e+02 -1.57103818e+02
 -1.99548759e+02 -2.49630400e+02 -3.08201405e+02 -3.76159229e+02
 -4.54446116e+02 -5.44049098e+02]
```



# Currying:

Breaking down the evaluation of a function that takes multiple argument into evaluating a sequence of single-argument functions.

In [1]:

```
def compose(b,c):
    def a(x,y):
        return b(c(x,y))
    return a

def BMI(weight, height):
    return weight / height**2

def evaluate_BMI(bmi):
    if bmi < 15:
        return "Very severely underweight"
    elif bmi < 16:
        return "Severely underweight"
    elif bmi < 18.5:
        return "Underweight"
    elif bmi < 25:
        return "Normal (healthy weight)"
    elif bmi < 30:
        return "Overweight"
    elif bmi < 35:
        return "Obese Class I (Moderately obese)"
    elif bmi < 40:
        return "Obese Class II (Severely obese)"
    else:
        return "Obese Class III (Very severely obese)"


f = compose(evaluate_BMI, BMI)

weight = 1
weight = float(input("weight (kg) "))
height = float(input("height (m) "))
while weight > 0:
    print(f(weight, height))
    weight = float(input("weight (kg) "))
    height = float(input("height (m) "))
```

```
weight (kg) 56
height (m) 1.7
Normal (healthy weight)
weight (kg) 0
height (m) 0
```