In the previous lecture, we developed the notion of the anti-derivative and computed a variety of examples. In this lecture, we are going to build on this theme.

Let us consider a few more examples of anti-derivatives before moving on to other topics. Let $g(x) = 4\sin(x) + \frac{2x^5 - \sqrt{x}}{x}$. Find the most general form of the function $G(x)$ such that $G'(x) = g(x)$.

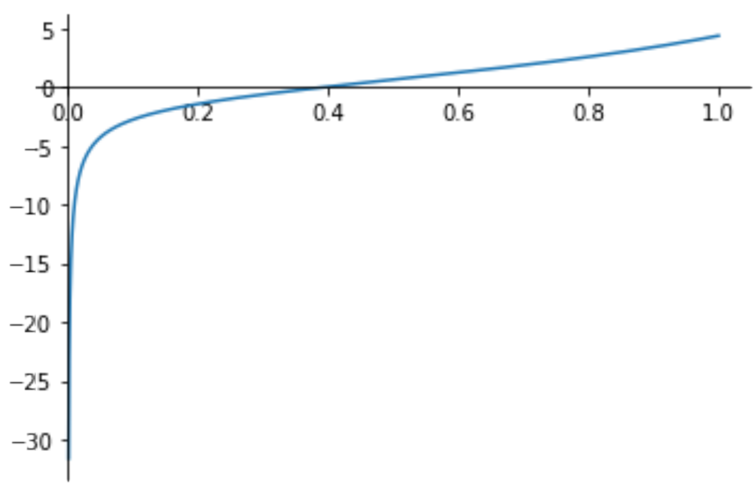
Here is a graph of $g(x)$.

In [19]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(1/1000, 1, 10000)
y = 4*np.sin(x)+2*x**4-1/x**(1/2)

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
```

Out[19]:
```
[<matplotlib.lines.Line2D at 0x122d6d8d0>]
```



To find the anti-derivative of $g(x)$, we first rewrite it in the form

$$4\sin(x) + 2x^4 - x^{-\frac{1}{2}}.$$

We already know how to find the anti-derivative of a constant multiplied by x raised to a power, as long as this power is not -1. We just get that same constant multiplied by x raised to the same power plus one, divided by the power plus one. So the anti-derivative of $2x^4$ is $\frac{2}{5}x^5 + C$, where C is an arbitrrary constant, and the anti-derivative of $-x^{-\frac{1}{2}}$ is $-2x^{\frac{1}{2}} + C$, where C is an arbitrary constant.

To compute the anti-derivative of $4sin(x)$, recally that the derivative of $sin(x)$ is $\cos(x)$ and the derivative of $\cos(x)$ is $-\sin(x)$. It follows that the anti-derivative of $4\sin(x)$ is $-4\cos(x) + C$, where C is, once again, an arbitrary constant.

We conclude that the anti-derivative of $g(x)$ is

$$G(x) = -4\cos(x) + \frac{2}{5}x^5 - 2\sqrt{x} + C,$$

where C is an arbitrary constant.

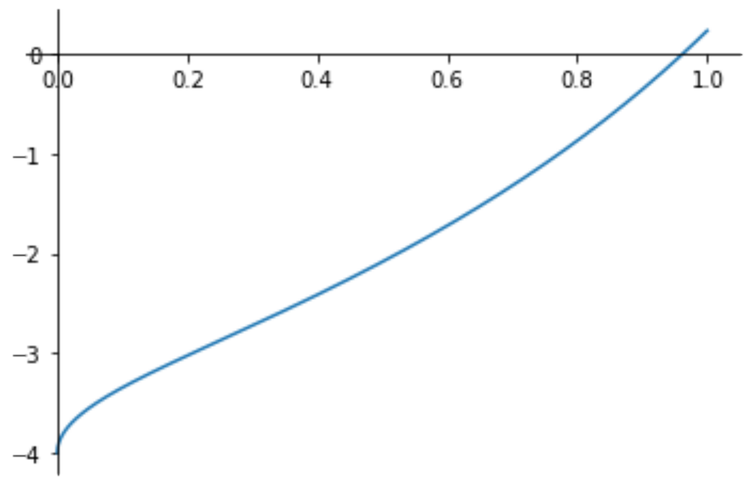
Can you imagine what the graph of $G(x)$ looks like?

In [23]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 1, 10000)
y = -4*np.cos(x)+(2/5)*x**5+2*x**(1/2)

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
```

Out[23]:
```
[<matplotlib.lines.Line2D at 0x123368f10>]
```



Let us do one more example involving the exponential function. Let $f(x) = e^x + \frac{1}{1+x^2}$. The derivative of e^x is e^x , and we know from Calculus 1 that the derivative of $tan^{-1}(x)$ is $\frac{1}{1+x^2}$. It follows that if we choose

$$F(x) = e^x + tan^{-1}(x) + C,$$

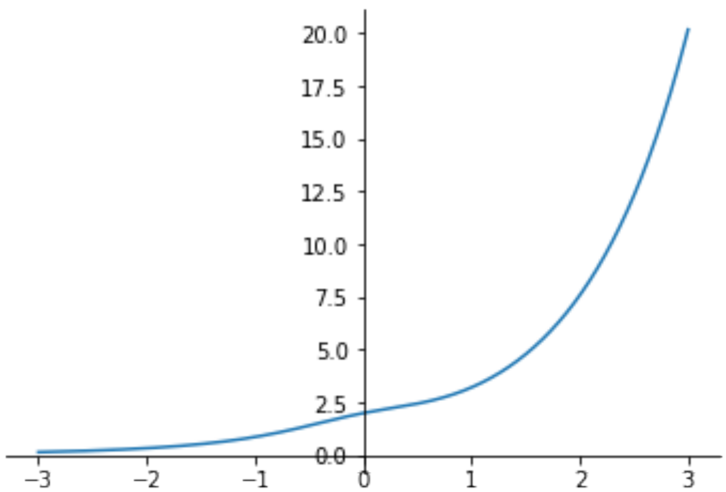
where C is an arbitrary constant, $F'(x) = f(x)$.

In [28]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 10000)
y = np.exp(x)+1/(1+x**2)

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
print("Graph of f(x)")
```

```
graph of f(x)
```

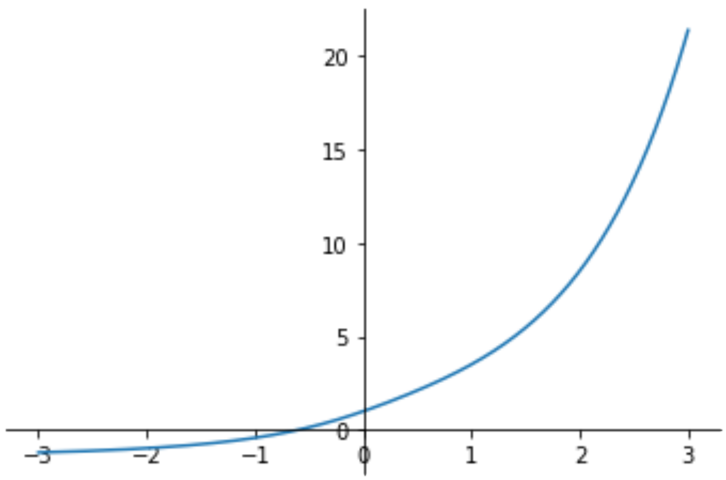


In [30]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 10000)
y = np.exp(x)+np.arctan(x)

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
print("Graph of F(x)")
```

```
Graph of F(x)
```



A particle moves in a straight line and has acceleration given by $a(t) = 6t + 4$. Its initial velocity is $v(0) = -6$ centimeters per second, and its initial displacement is $s(0) = 9$ centimeters. Find its position function $s(t)$.

Since accelleration is the derivative of velocity, velocity must be the anti-derivative of accelleration. We have

$$v(t) = 3t^2 + 4t + C,$$

where C is a constant. By C is not arbitrary since we know that $v(0) = -6$, which forces

$$v(t) = 3t^2 + 4t - 6.$$

Since velocity is the derivative of position, the position is the anti-derivative of velocity. Therefore,

$$s(t) = t^3 + 2t^2 - 6t + C,$$

where C is a constant. Once again C is not arbitrary since $s(0) = 9$, which means that

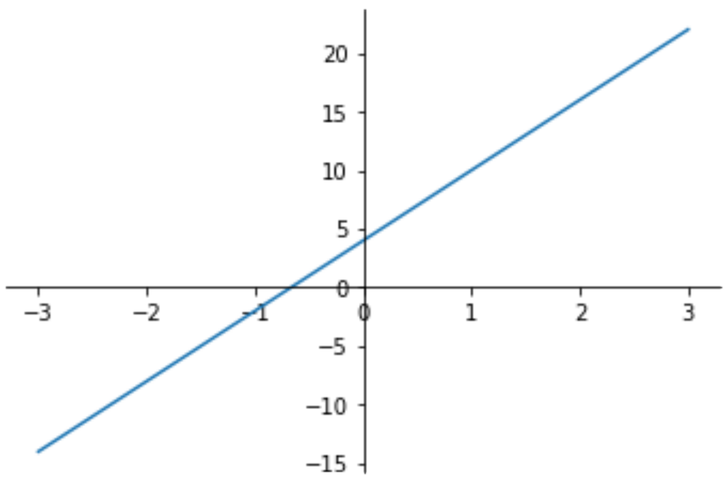
$$s(t) = t^3 + 2t^2 - 6t + 9.$$

In [31]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-3, 3, 10000)
y = 6*x+4

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
print("Graph of a(t)")
```

Graph of a(t)

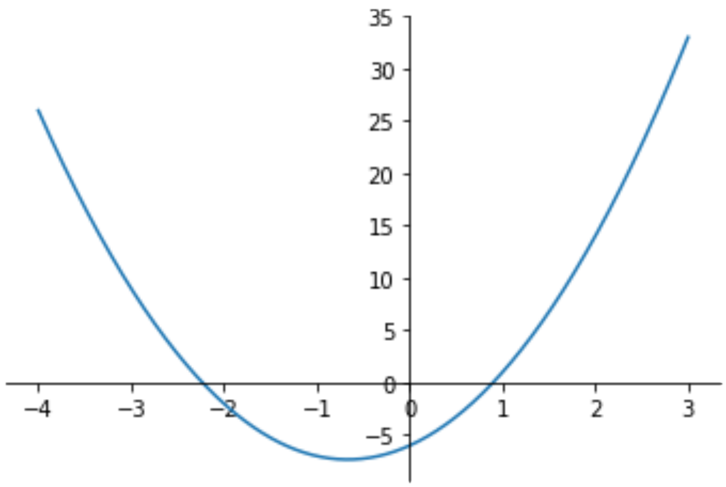


In [35]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-4, 3, 10000)
y = 3*x**2+4*x-6

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
print("Graph of v(t)")
```

```
Graph of v(t)
```

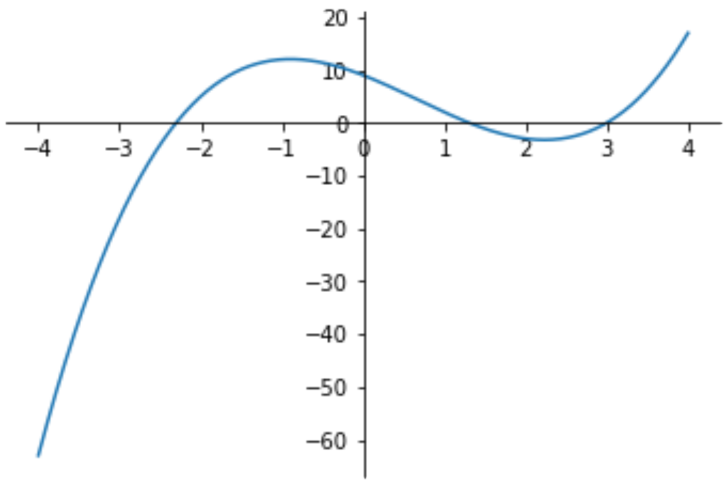


In [37]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-4, 4, 10000)
y = x**3-2*x**2-6*x+9

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
plt.plot(x,y)
print("Graph of s(t)")
```

```
Graph of s(t)
```



Let us now look at a similar problem from a different point of view. Suppose that a ball is dropped from the mountain 500 meters tall. How long does it take for the ball to hit the ground?

The accelleration due to gravity is a constant. It is, approximately, 9.8 meters per second squared. for the purposes of this discussion, we will just call it 10 meters per second squared, not exactly right, but close enough. This means that if $a(t)$ denotes accelleration,

$$a(t) = -10.$$

Since velocity is the anti-derivative of accelleration,

$$v(t) = -10t + C,$$

where C is a constant. But C is not arbitrary since $v(0) = 0$ (the object starts at rest). It follows that

$$v(t) = -10t.$$

The position $s(t)$ is the anti-derivative of velocity. It follows that

$$s(t) = -5t^2 + C,$$

where C is a constant. Once again C is not arbitrary since $s(0) = 100$. It follows that

$$s(t) = -5t^2 + 500.$$

So when does the ball hit the ground? It happens when $s(t) = 0$. Setting $s(t)$ equal to 0 above, we see that

$$t = 10.$$

We shall now graph $v(t)$ and $s(t)$ on the same graph. There is no point graphing $a(t)$ since it is a constant. Which one is which?

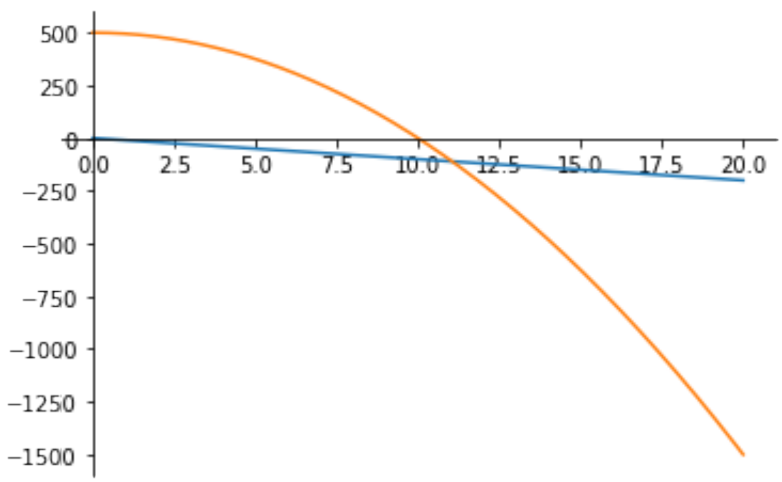
In [46]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 20, 10000)
#y = -10
y1 = -10*x
y2 = -5*x**2+500

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')

#plt.plot(x,y)
plt.plot(x,y1)
plt.plot(x,y2)
print("Graph of v(t) and s(t)")
```

```
Graph of v(t) and s(t)
```



What is the velocity of the ball when it hits the ground? Well, when $t = 10$, $v(t) = v(10) = -10(10) = -100$, so the speed of the ball when it hits the ground is 100 meters per second- pretty fast!

Notice that in the diagram above, the graph of $s(t)$ intersects the horizontal axis precisely when $t = 10$, just as our calculations showed.

Let us now modify the problem at bit. The ball is now thrown upwards from a building of height s_0 meters with the initial velocity of v_0 meters per second. How high is the ball going to get and how long is it going to take for it to hit the ground?

To solve this problem, we set things up as before, with a few small differences. We still have $a(t) = -10$. This is a law of nature. It follows that

$$v(t) = -10t + C,$$

where C is computed by plugging in $t = 0$. Since $v(0) = v_0$, by assumption,

$$v(t) = -10t + v_0.$$

Moving right along,

$$s(t) = -5t^2 + v_0 t + C,$$

where C is computed by plugging in $t = 0$. Since $s(0) = s_0$.

$$s(t) = -5t^2 + v_0 t + s_0.$$

Computing how high the ball is going to get is equivalent to maximizing the function $s(t)$. Its derivative is $v(t)$, and setting it equal to 0 yields

$$t = \frac{v_0}{10}.$$

Never mind calculus for a minute. The point in time where the velocity is 0 is, of course, the point in time when the ball reaches its highest point. We do not need calculus to tell us that!

Plugging this value of t into the formula for $s(t)$ yields

$$s\left(\frac{v_0}{10}\right) = -\frac{v_0^2}{20} + \frac{v_0^2}{10} + s_0 = \frac{v_0^2}{20} + s_0.$$

To fins when the ball is going to hit the ground, we must solve the equation

$$-5t^2 + v_0 t + s_0 = 0.$$

Just to review the concepts, let's do that. Dividing both sides by -5 we obtain the equation

$$t^2 - \frac{v_0}{5}t - \frac{s_0}{5} = 0.$$

Completing the square, we obtain

$$\left(t - \frac{v_0}{10}\right)^2 - \frac{v_0^2}{100} - \frac{s_0}{5} = 0.$$

By the way, this tells us that maximum of $s(t)$ is at $t = \frac{v_0}{10}$, independently of the dertivative calculation above. As I mentioned many times, completing the square is a powerful technique.

It follows if t_{ground} denotes the time when the ball hits the ground,

$$t_{ground} = \frac{v_0}{10} + \sqrt{\frac{v_0^2}{100} + \frac{s_0}{5}}.$$

Let's throw some numbers in. Suppose that $s_0 = 500$ meters and $v_0 = 10$ meters per second. Then

$$t_{ground} = 1 + \sqrt{101},$$

which is pretty close to 11 seconds.

Let's graph this situation to confirm our theoretical calculations

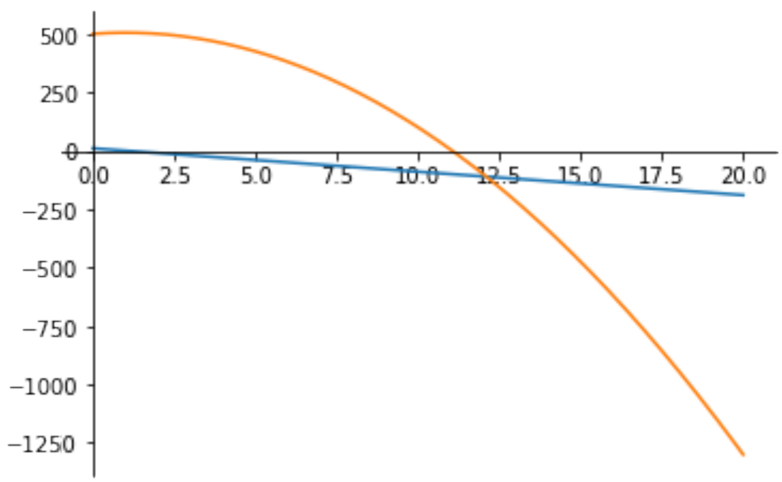
In [48]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 20, 10000)
#y = -10
y1 = -10*x+10
y2 = -5*x**2+10*x+500

ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')

#plt.plot(x,y)
plt.plot(x,y1)
plt.plot(x,y2)
print("Graph of the new v(t) and s(t)")
```

```
Graph of the new v(t) and s(t)
```



In [ ]: