Funciones de Pozo Infinito

$\psi(x) = \sqrt{\frac{2}{L}}\sin(\frac{n\pi}{L}x)$

Darío Mitnik

In [1]:

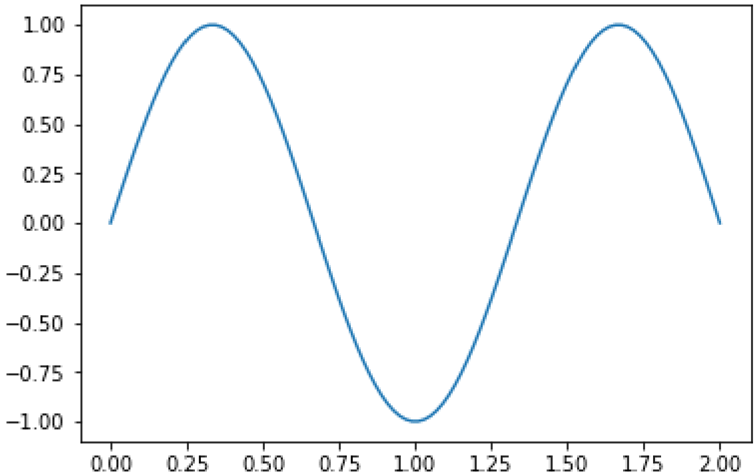
```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
# Definición de funciones de onda

def psi(n,L,x):
    y = np.sqrt(2/L) * np.sin(n * np.pi * x / L)
    return y

# Energías
def ener(n,L):
    y = 1./2. * (n * np.pi / L)**2
```

In [3]:

```
L = 2
npts = 100
x = np.linspace(0,L,npts)
y = psi(3,L,x)

plt.plot(x,y)
plt.show()
```



In [4]:

```
# Chequeo Ortonormalidad

yy = psi(3,L,x)*psi(2,L,x)
sum = np.trapz(yy,x)
print('sum=',sum)
```

```
('sum=', -1.2302984152279262e-16)
```

Cálculo Analítico

In [5]:

```
# Importamos sympy
import sympy as sy
from sympy import init_printing,pi,symbols,sqrt,sin,Function,integrate,oo,diff,I

init_printing()

# Definimos las funciones phi(n,L,x)
x = symbols('x', real=True)
L = symbols('L', real=True,positive=True)
n = symbols('n', integer=True,positive=True)

def phi(n,L,x):
    arg =   n * pi / L * x
    psi =  sqrt(2/L) *  sin(arg)
    return psi
```

In [6]:

```
# Chequeo Ortonormalidad

f1, f2 =  symbols('f1 f2', cls=Function)
n1 = symbols('n1', integer=True,positive=True)
n2 = symbols('n2', integer=True,positive=True)



f1 = phi(n1,L,x)
f2 = phi(n2,L,x)

Soverlap =  integrate(f1.conjugate()*f2,(x,0,L))
Soverlap
```

Out[6]:

$$\begin{cases} 0 & \text{for}\: n_{1} \neq n_{2} \\ 1 & \text{otherwise} \end{cases}$$

In [7]:

```
integrate(phi(n1,L,x)*phi(n2,L,x),(x,0,L))
```

Out[7]:

$$\begin{cases} 0 & \text{for}\: n_{1} \neq n_{2} \\ 1 & \text{otherwise} \end{cases}$$

In [8]:

```
sy.simplify(integrate(phi(n1,L,x)*x*phi(n2,L,x),(x,0,L)) )
```

Out[8]:

$$\begin{cases} \frac{4Ln_1n_2\left((-1)^{n_1+n_2}-1\right)}{\pi^2\left(n_1^4-2n_1^2n_2^2+n_2^4\right)} & \text{for}\ n_1 \neq n_2 \\ \frac{L}{2} & \text{otherwise} \end{cases}$$

In [9]:

```
def xmed(n1,n2,L):
    y = integrate(phi(n1,L,x)*x*phi(n2,L,x),(x,0,L))
    return sy.simplify(y)
```

In [10]:

```
xmed(2,2,L)
```

Out[10]:

$$\frac{L}{2}$$