# PRECÁLCULO

## FUNCIONES Y SUS LÍMITES

### Juliho Castillo Colmenares

*IMPORTANTE: Para correr el siguiente código, necesitarás* [*Sagemath.*](https://youtu.be/8KT9GMruOrU)

El cálculo es muy diferente de las matemáticas que has estudiado hasta hoy, porque tiene que ver con cambios. En esta sección estudiaremos el comportamiento en cambio de funciones y sus límites.

## Funciones y su representación.

Las funciones surgen cuando una cantidad depende de otra. Consideremos algunos ejemplos.

#### Ejemplo A

El área de un círculo depende del radio del mismo:

def area\_circulo(radio):  
 return 3.14159\*radio^2  
  
print(area\_circulo(2.5267490372))

20.0573578810604

#### Ejemplo B

La población humana (en todo el mundo) depende del tiempo . En la siguiente tabla, se muestra el año y la población en milliones correspondiente:

poblacion = [  
 [1900, 1650],  
 [1910, 1750],  
 [1920, 1860],  
 [1930, 2070],  
 [1940, 2300],  
 [1950, 2560],  
 [1960, 3040],  
 [1970, 3710],  
 [1980, 4450],  
 [1990, 5280],  
 [2000, 6080],  
 [2010, 6870]  
 ]

A partir de esta tabla, podemos crear una función punto a punto:

def P(t):  
 for dato in poblacion:  
 if t==dato[0]:  
 return dato[1]  
   
print(P(1950))

2560

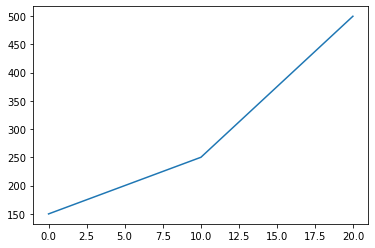
#### Ejemplo C

El costo de enviar un paquete depende del peso . Por ejemplo, supongamos que los precio están dados por las siguiente reglas:

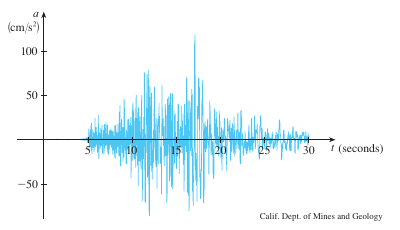
* Menor o igual a 10 kgs: Costo fijo de $150 más $10 por kilogramo
* Mayor a 10 kgs: $25 por kilogramo
* La capacidad máxima del envíoi es 20 kgs

"""  
Definimos la función de costo C en función del peso w  
"""  
def C(w):  
 if w<0:  
 return "¿Seguro que estás enviando algo?"  
 elif w<=10:  
 return 150 + 10\*w  
 elif w<=20:  
 return 25\*w  
 else:  
 return "¡Excediste la capacidad!"

"""  
Graficamos en el intervalo (0,20)  
"""  
import numpy as np  
import matplotlib.pyplot as plt  
  
x = np.arange(0,20,0.01)  
C = np.vectorize(C)  
y = C(x)  
  
plt.plot(x,y)  
plt.show()



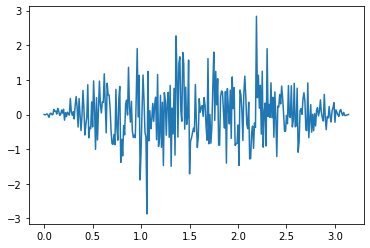
#### Ejemplo D

La aceleracion vertical del suelo en un terremoto  


"""  
Simularemos la acelaración a respecto al tiempo t  
"""  
def a(t):  
 return np.sin(t)\*np.random.normal()

"""  
Graficamos en el intervalo (0, pi)  
"""  
x = np.arange(0,np.pi,0.01)  
a = np.vectorize(a)  
y = a(x)  
  
plt.plot(x,y)

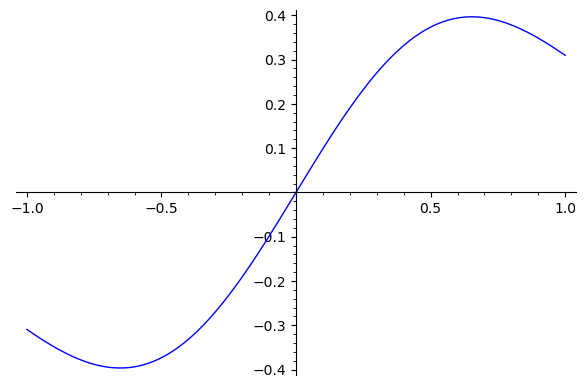
[<matplotlib.lines.Line2D object at 0x6ffe91c1f1d0>]



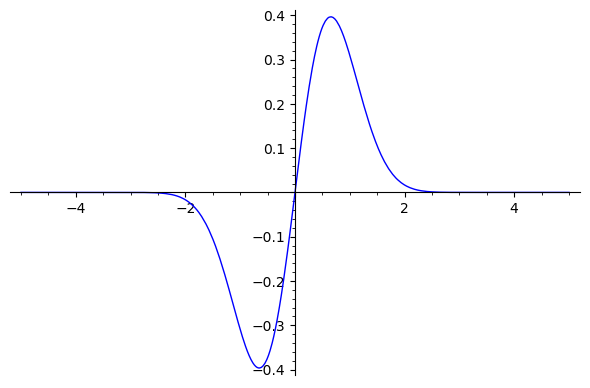
Una **función** es una regla que asigna a cada elemento de un conjunto un elemento de un conjunto .

Al conjunto se le llama **dominio**, mientras que a se le llama **contradominio**.

x = var("x")  
f(x) = exp(-x^2)\*sin(x)  
grafica = plot(f)  
show(grafica)



grafica = plot(f, (x,-5,5))  
show(grafica)



### Representación de funciones

#### Ejercicio

Encuentre el dominio de cada función:

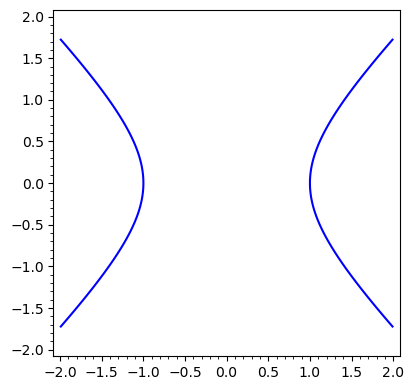
#### Prueba de la linea vertical

Una curva en el plano es la gráfica de una función si y solo si no existe línea vertical alguna que intersecte la curva más de dos veces.

#### Ejemplo

Determina si la ecuación define una función .

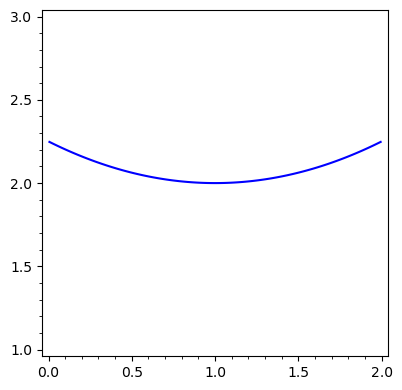
x,y = var("x,y")  
ecuacion = x^2-y^2==1  
grafica = implicit\_plot(ecuacion, (x,-2,2), (y,-2,2))  
show(grafica)



#### Ejemplo

Determina si la ecuación define una función .

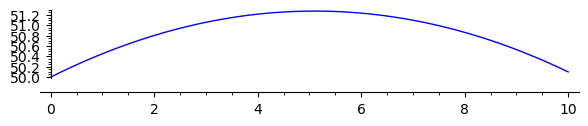
x,y = var("x,y")  
ecuacion = (x-1)^2==4\*(y-2)  
grafica = implicit\_plot(ecuacion, (x,0,2), (y,1,3))  
show(grafica)



#### Ejemplo

Determina si la gráfica de las ecuaciones paramétricas:  
  
define una función .

t = var("t")  
x(t) = 10\*t  
y(t) = 50+5\*t-(1/2)\*(9.8)\*t^2  
grafica = parametric\_plot((x(t),y(t)), (t,0,1))  
  
show(grafica)



#### Ejercicio de exploración

Las funciones anteriores representan la posición de un proyectil en función del tiempo. Determina gráficamente:

* la distancia a la que choca contra el suelo
* cuanto tiempo tarde en ocurrir esto
* la altura máxima y el tiempo en que se alcanza

¿Puedes determinar cuál es la función ?

#### Ejercicio de exploración

Muchas relaciones fundamentales en matemáticas y física no se pueden expresar como funciones. Aún así se puede definir el dominio de una relación. Averigua en internet como es que se hace esto y determinalo gráficamente para la relación

*¿Cuál es el intervalo minimal en el eje que necesitas para graficar completamente la relación?*

### Funciones representadas a trozos

#### Ejemplo

Una función esta definida por

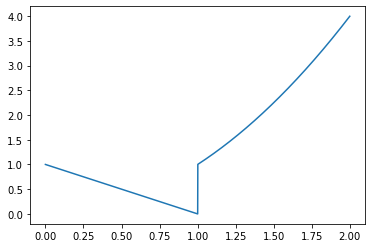
Evaluar y graficar la función.

def f(x):  
 if x<=1:  
 return 1-x  
 else:  
 return x\*\*2

import numpy as np  
f = np.vectorize(f)  
x = [0,1,2]  
y = f(x)  
print(y)

[1 0 4]

import matplotlib.pyplot as plt  
  
x = np.arange(0,2,0.001)  
y = f(x)  
  
plt.plot(x,y)  
plt.show()



#### Ejercicio (Valor absoluto)

* Investiga como se define la función *valor absoluto*
* Implementala en Python
* Evaluala en el arreglo
* Traza la gráfica

### Simetría

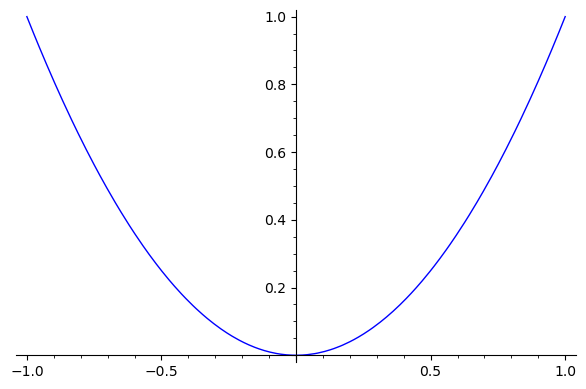
Si una función satisface la relación en su dominio, diremos que la función es **par**. Su gráfica es simétrica respecto al eje .

#### Ejemplo

x = var("x")  
f(x) = x^2  
show(f(-x))

\newcommand{\Bold}[1]{\mathbf{#1}}x^{2}

plot(f)

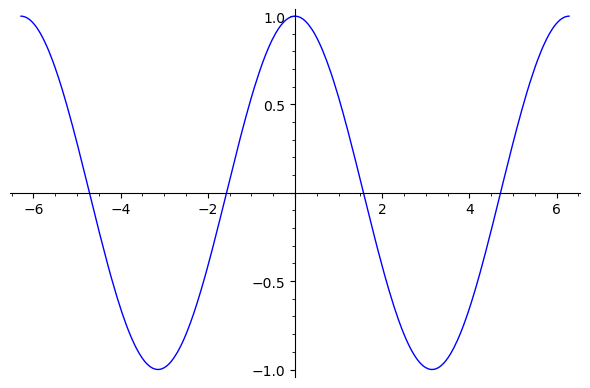


#### Ejemplo

x = var("x")  
g(x) = cos(x)  
show(g(-x).simplify())

\newcommand{\Bold}[1]{\mathbf{#1}}\cos\left(x\right)

plot(g, (x,-2\*pi, 2\*pi))



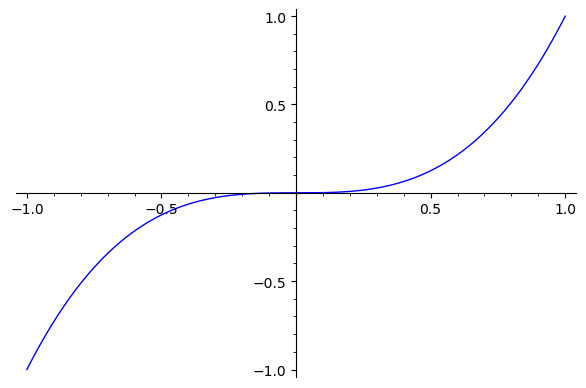
Si una función satisface la relación en su dominio, diremos que la función es **impar**. Su gráfica es simétrica respecto al origen.

#### Ejemplo 4

x = var("x")  
f(x) = x^3  
show(f(-x).simplify())

\newcommand{\Bold}[1]{\mathbf{#1}}-x^{3}

plot(f)

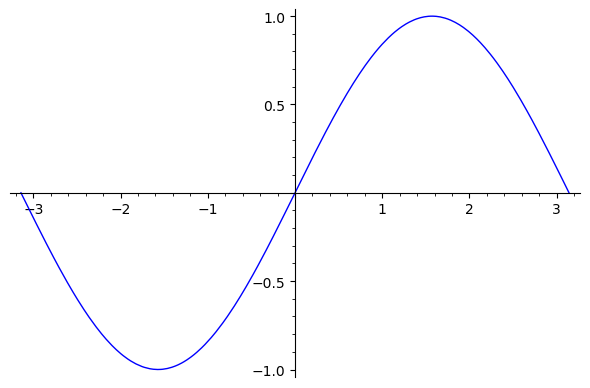


#### Ejemplo

x = var("x")  
g(x) = sin(x)  
show(g(-x).simplify())

\newcommand{\Bold}[1]{\mathbf{#1}}-\sin\left(x\right)

plot(g, (x,-pi,pi))



#### Ejercicio

Determina si las siguiente funciones son pares, impares o ninguna:

#### Funciones crecientes y decrecientes

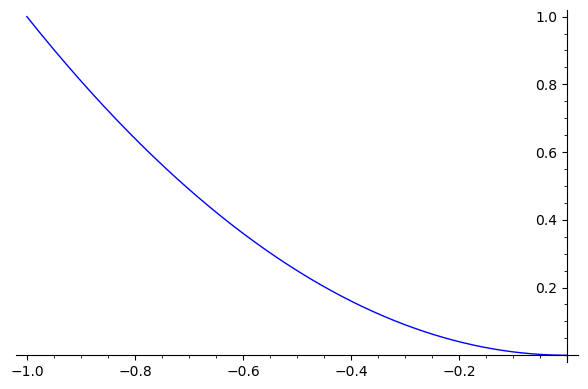
Una función es **creciente** en un intervalo si  
 siempre que .

Una función es **decreciente** en un intervalo si  
 siempre que .

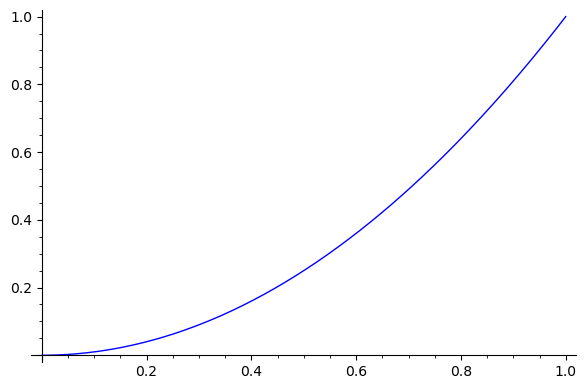
**Observación** El intervalo es muy importante para determinar si la función es creciente o no. Consideremos la función en los intervalos

x = var("x")  
f(x) = x^2

plot(f, (x,-1,0))



plot(f, (x,0,1))



plot(f, (x,-1,1))

