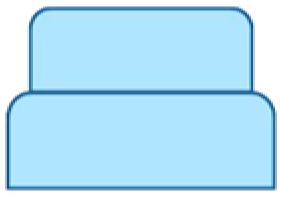
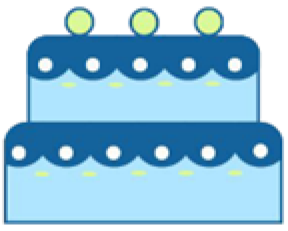
Funciones compuestas

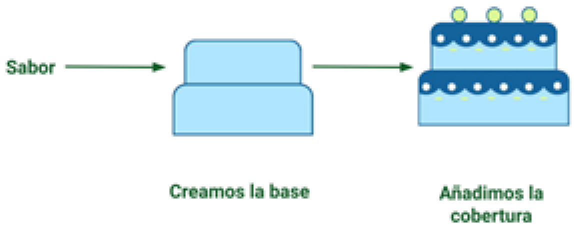
Expliquemoslo mediante el proceso de hacer un pastel, primero tenemos que pensar en hacer la base:



Después añadimos la cobertura

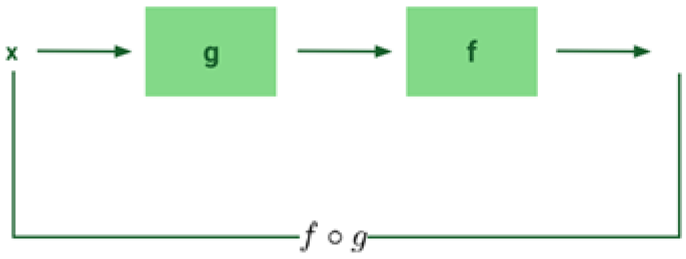


Entonces así es como resulta el pastel.
En resumen el algoritmo se podría resumir como;



Retomando el caso de las funciones. El proceso de composición de funciones es bastante similar. Lo podemos ver como una maquina que realiza procesos, pueden ser procesos algebraicos u otros, pero algún tipo de procesamiento le van a realizar a la variable independiente:

- Tenemos nuestra variable x que la vamos a ingresar a una función $g(x)$ y entonces sale ya procesada.
- Lo que salga de $g(x)$ va a ingresar a una segunda función, en este caso $f(x)$.
- El resultado final es la composición de funciones $f \circ g$.



Pero también tenemos la definición formal siguiente.

Definición

Conocidas las funciones $f(x)$ y $g(x)$, la composición de
$f(x)$ y $g(x)$ está dada por:

$$f \circ g = (f \circ g)(x) = f(g(x))$$

In [ ]:
```python
#Importando librerías y ajustando Latex
import matplotlib.pyplot as plt
import numpy as np

#Configurando Latex
# Configuración de Matplotlib para usar LaTeX
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern Roman"],
    "text.latex.preamble": r"\usepackage{amsmath}"
})
```

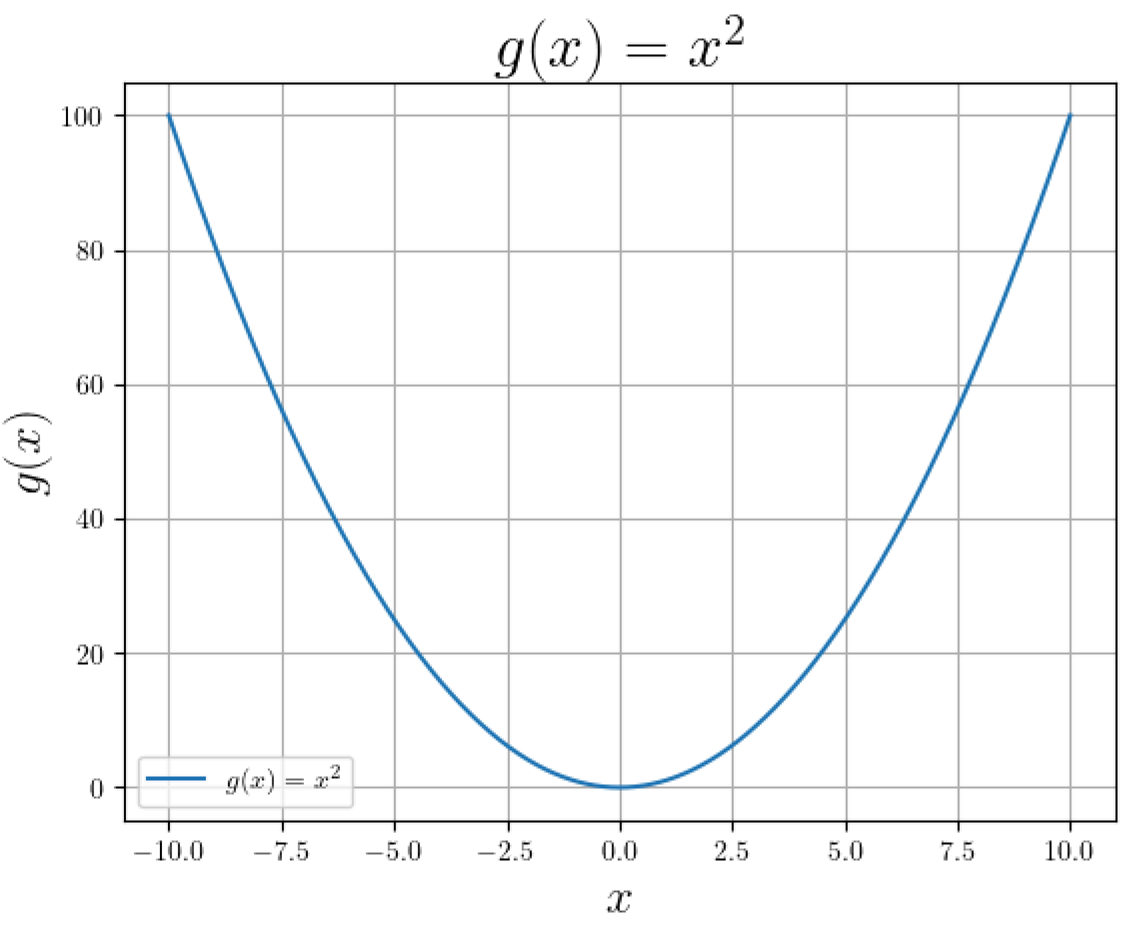
Veamos como componer 2 funciones.
Pero primero declaremos nuestra variable independiente x y su resolución de 1000 puntos.

In [ ]:
```python
N = 1000 #Resolución de la variable
x=np.linspace(-10,10,N)
```

Primero, obtengamos nuestra primera función $g(x)$, la elección será una $parábola$ con ecuación $g(x) = x^2$, además grafiquemosla en un intervalo $-10 \leq x \leq 10$.

In [ ]:
```
#Definiendo una función de parabola
#Este será mi g(x)
def g(x):
    return x**2
y = g(x)
fig, ax = plt.subplots()
ax.plot(x,y,label=r'$g(x) = x^2$')

plt.title(r'$g(x) =x^2 $',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$g(x)$',fontsize=18)
plt.grid()
plt.legend()
plt.show()
```

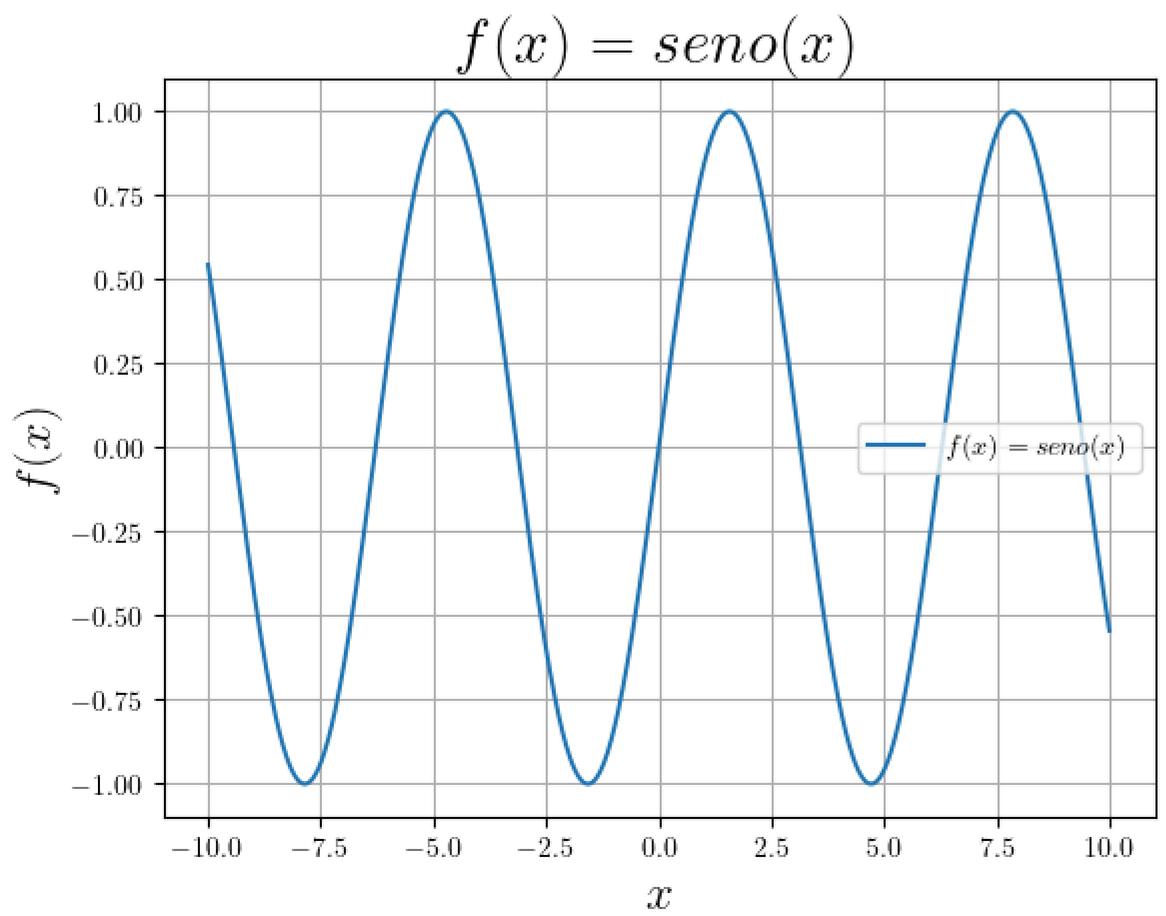


Ahora definiremos nuestra segunda función $f(x)$, esta será una función $seno(x)$, entonces $f(x) = seno(x)$, además la graficaremos en un intervalo de $-10 \leq x \leq 10$.

In [ ]:
```
#Definiendo una función seno(x)
#Este será mi f(x)
def f(x):
    return np.sin(x)

y2 = f(x)
fig, ax = plt.subplots()
ax.plot(x,y2,label=r'$f(x) = seno(x)$')

plt.title(r'$f(x)= seno(x) $',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.grid()
plt.legend()
plt.show()
```



Ahora hagamos el proceso de composición de funciones

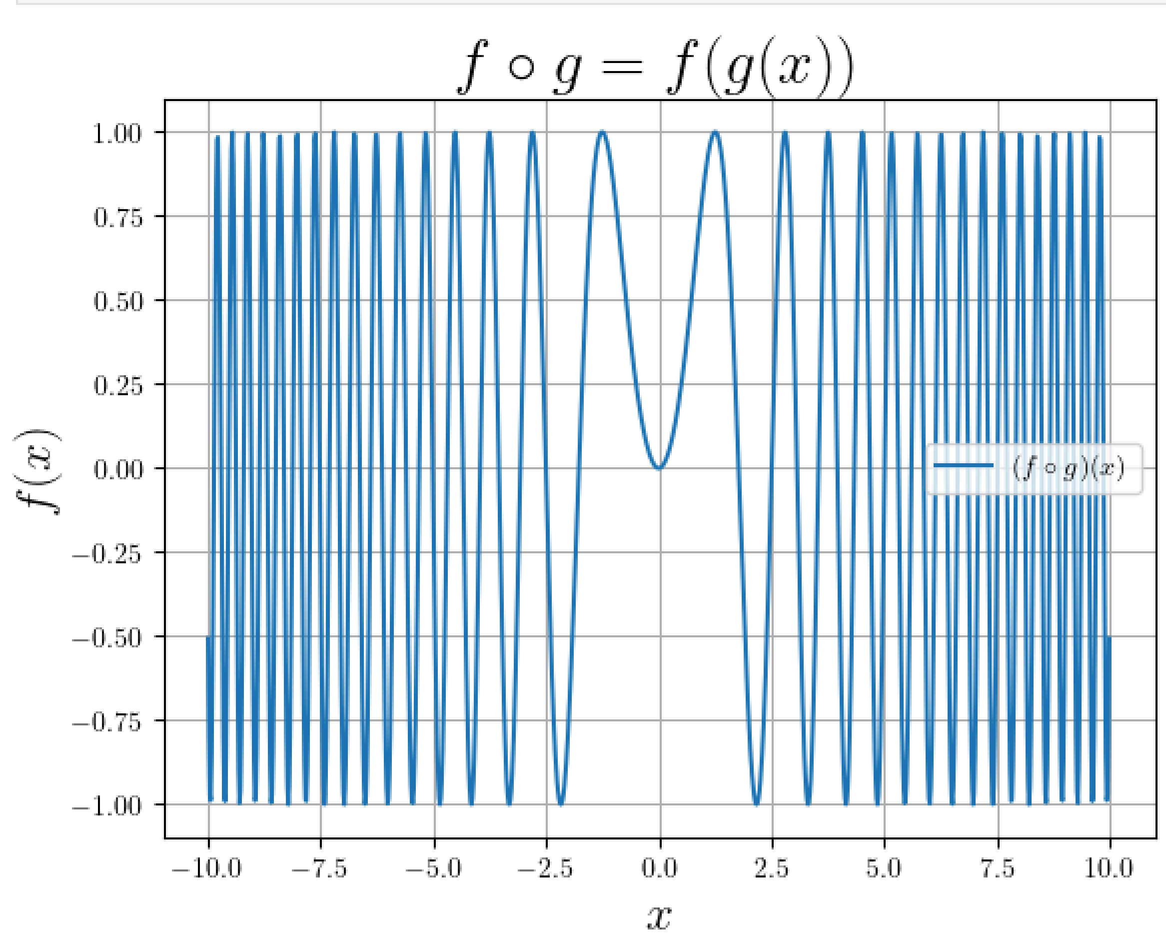
$$f \circ g$$

Es decir:

1. Evaluamos $g(x)$
2. La salida de $g(x)$ la evaluamos en $f(x)$
3. Graficamos la función

In [ ]:
```
#Composición de funciones
#Directamente
f_o_g = f(g(x))

#Graficando
fig, ax = plt.subplots()
ax.plot(x,f_o_g,label=r'$(f\circ g)(x)$')
plt.title(r'$f\circ g = f(g(x)) $',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.grid()
plt.legend()
plt.show()
```



Recuerda:
En la función que se encuentra hasta la *derecha* de

$$f \circ g = (f \circ g)(x) = f(g(x))$$

es la que lleva la x o la que se evalua **primero** y la función de hasta la *izquierda*, es la que **recibe la salida de la primera evaluación de** x.