IV Funktionen und Graphen

1. Strecken, Verschieben, Spiegeln von Graphen

Beispiel:

$$
\begin{aligned}
f(x) &= (x+2)^3 \\
&= x^3 + 3x^2 \cdot 2 + 3x \cdot 4 + 8 \\
&= x^3 + 6x^2 + 12x + 8
\end{aligned}
$$

Exkurs: Pascalsches Dreicek

$$
\begin{array}{c}
1 \\
1 \quad 1 \\
1 \quad 2 \quad 1 \\
1 \quad 3 \quad 3 \quad 1 \\
1 \quad 4 \quad 6 \quad 4 \quad 1 \\
1 \quad 5 \quad 10 \quad 10 \quad 5 \quad 1 \\
\dots
\end{array}
$$

$(a+b)^0 = 1$

$(a+b)^1 = 1a + 1b$

$(a+b)^2 = 1a^2 + 2ab + 1b^2$

$(a+b)^3 = 1a^3 + 3a^2b + 3ab^2 + 1b^3$

$(a+b)^4 = 1a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + 1b^4$

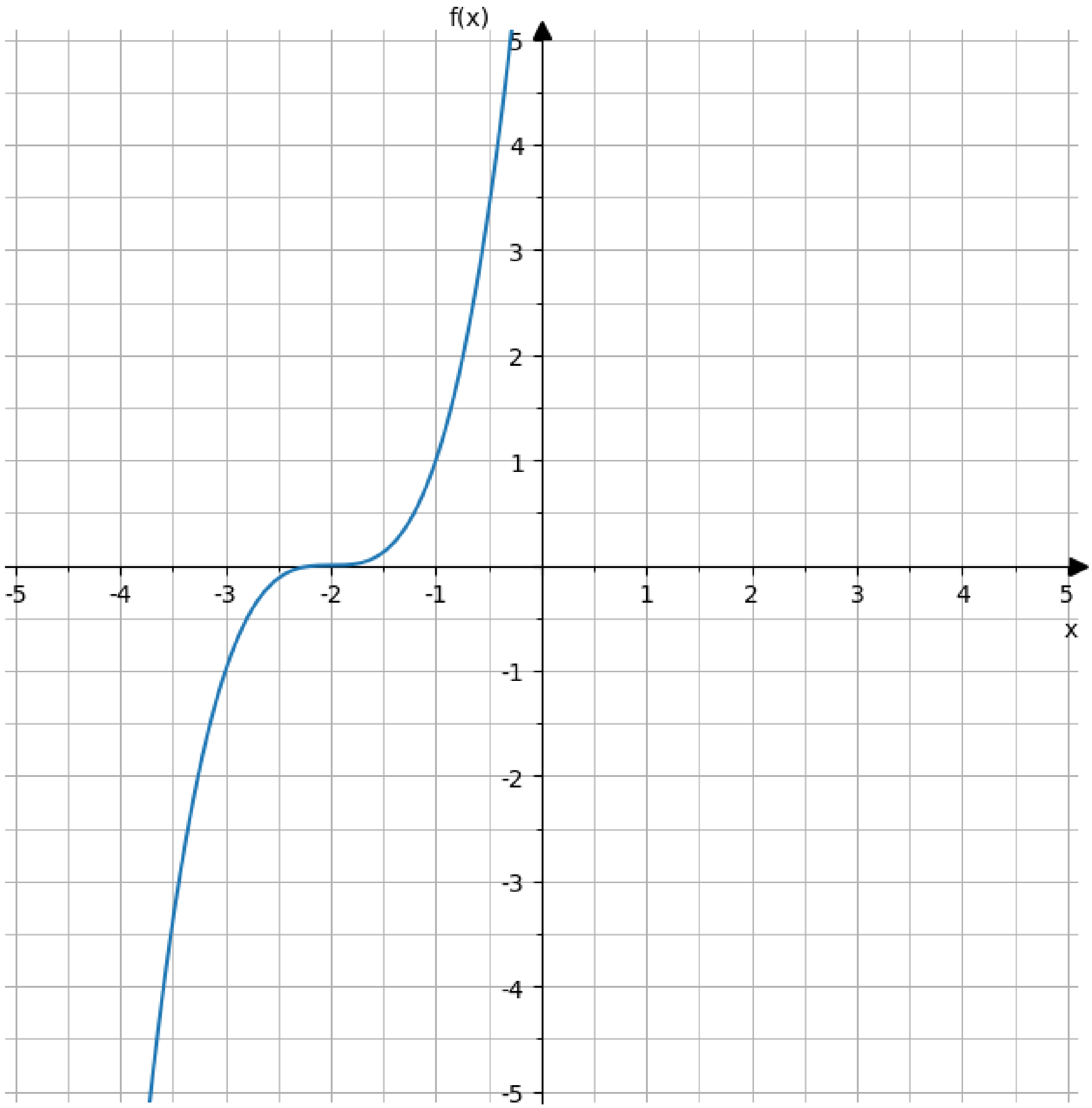
$(a+b)^5 = 1a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + 1b^5$

In [9]:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatt

# Defintionsmenge und Funktion
# -----------------------------
a= -5.1 # untere x-Intervallgrenze
b= 5.1 # obere x-Intervallgrenze
c = -5.1# untere y-Intervallgrenze
d = 5.1 # obere y-Intervallgrenze
x = np.linspace(a, b,1000)
y1= x**3+6*x**2+12*x+8
#y2=(x+2)**3
# -----------------------------

# Einstellung des Graphen
fig=plt.figure(figsize=(8,8))
ax= fig.add_subplot(1,1,1, aspect =1)

# Definiton der Haupteinheiten, reele Zahlen ohne die 0
def major_tick(x, pos):
    if x==0:
        return ""
```

```python
    return int(x)

# Achsenskalierung
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.xaxis.set_minor_locator(AutoMinorLocator(2))
ax.yaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_minor_locator(AutoMinorLocator(2))
ax.xaxis.set_major_formatter(FuncFormatter(major_tick))
ax.yaxis.set_major_formatter(FuncFormatter(major_tick))

# Position der Achsen im Schaubild
ax.spines[['top','right']].set_visible(False)
ax.spines[['bottom','left']].set_position('zero')

# Pfeile für die Achsen
ax.plot((1),(0), ls="", marker= ">", ms=7, color="k", transform=ax.get_yaxis
ax.plot((0),(1), ls="", marker= "^", ms=7, color="k", transform=ax.get_xaxis

# Achsenlänge und Beschriftung
ax.set_xlim(a,b)
ax.set_ylim(c, d)
ax.set_xlabel("x", loc="right")
ax.set_ylabel("f(x)", loc="top", rotation=0)

# Kästchen
ax.grid(linestyle="-", which="major",linewidth=0.7, zorder=-10)
ax.grid(linestyle="-", which="minor",linewidth=0.5, zorder=-10)

# Plot der Funktion
ax.plot(x,y1, zorder=10)
#ax.plot(x,y2, zorder=10)
#plt.show()
```

Out[9]: [<matplotlib.lines.Line2D at 0x13825a960>]



Wertetabelle:

x	-2	-1	0	1
$f(x)$	0	1	8	24

Fragen:

- Welche Auswirung hat es , wenn man jeden Funktionswert mit der gleichen Zahl d=-1 addiert?

x	-2	-1	0	1
	0	1	8	27

x	-2	-1	0	1
$f(x)$				
$f(x)-1$	0-1	1-1	8-1	27-1

=> Alle Punkte des Funktionsgraphen liegen um eine Einheit tiefer, als bei der Ausgangsfunktion.
=> Verschiebung des Funktionsgraphen entlang der y-Achse.

- Welche Auswirkung hat es, wenn man jeden Funktionswert mit der gleichen Zahl a=2 multipliziert?

x	-2	-1	0	1
$f(x)$	0	1	8	27
$2 \cdot f(x)$	0	2	16	54

=> Alle y-Werte der Punkte des ursprünglichen Funktionsgraphen werden mit a-vervielfacht und erhalten das entegengesetzte Vorzeichen.
=> Streckung des ursprünglichen Funktionsgraphen mit dem Faktor a.

- Welche Auswirkung hat es, wenn man jeden Funktionswert mit der gleichen Zahl a=-1 multipliziert?

x	-2	-1	0	1
$f(x)$	0	1	8	27
$-1 \cdot f(x)$	0	-1	-8	-27

=> Alle y-Werte der Punkte des ursprünglichen Funktionsgraphen erhalten das entegengesetzte Vorzeichen.
=> Speigelung des ursprünglichen Funktionsgraphen an der x-Achse.

- Welche Auswirkung hat es, wenn man von jeden x-Wert die gleiche Zahl c=2 subrahiert.

x	**-2**	**-1**	**0**	**1**
$f(x)$	0	1	8	27
$f(x-2)$	-8	-1	0	1

=> Alle Punkte des Funktionsgraphen haben den Funktionswert, den der ursprüngliche Graph schon zwei Einheiten weiter links gehabt hat.
=> Der Graph wird verschoben auf entlang der x-Achse.

Satz:
Der Graph der Funktion g mit $g(x) = a \cdot f(x-c) + d$, mit $a, c, d \in \mathbb{R}, \quad a \neq 0$ entsteht aus dem Graphen der Funktion f durch

- Streckung in y-Richtung mit dem Faktor $|a|$
- Verschiebung entlag der y-Achse um d
- Verschiebung entlang der x-Achse um c.

Beispiel:

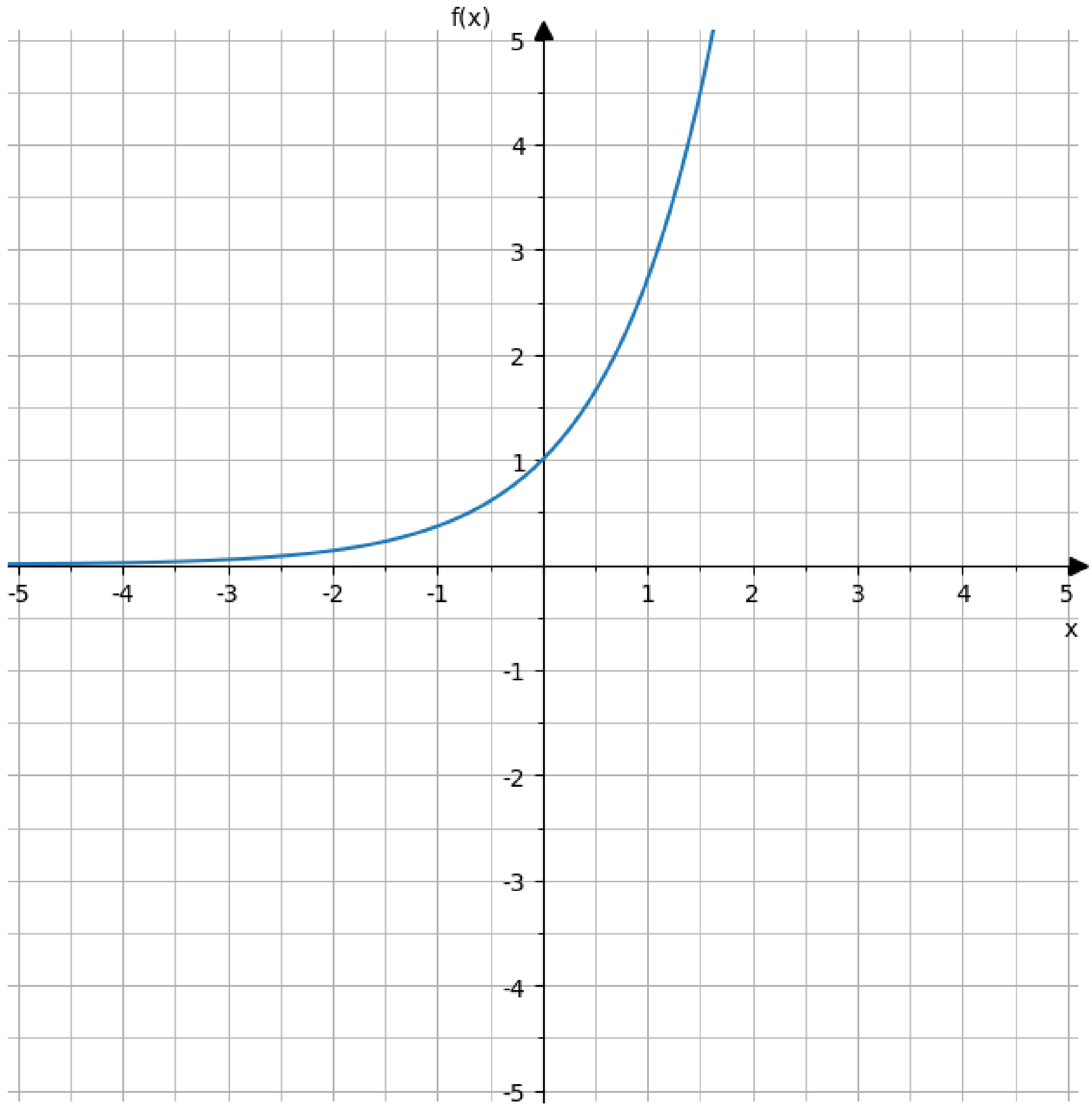
$$f(x) = e^x$$

In [10]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatt

# Defintionsmenge und Funktion
# ------------------------------
a= -5.1 # untere x-Intervallgrenze
b= 5.1 # obere x-Intervallgrenze
c = -5.1# untere y-Intervallgrenze
d = 5.1 # obere y-Intervallgrenze
x = np.linspace(a, b,1000)
y1= np.exp(x)
#y2=(x+2)**3
# ----------------------------
```

```python
# Einstellung des Graphen
fig=plt.figure(figsize=(8,8))
ax= fig.add_subplot(1,1,1, aspect =1)

# Definiton der Haupteinheiten, reele Zahlen ohne die 0
def major_tick(x, pos):
    if x==0:
        return ""
    return int(x)

# Achsenskalierung
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.xaxis.set_minor_locator(AutoMinorLocator(2))
ax.yaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_minor_locator(AutoMinorLocator(2))
ax.xaxis.set_major_formatter(FuncFormatter(major_tick))
ax.yaxis.set_major_formatter(FuncFormatter(major_tick))

# Position der Achsen im Schaubild
ax.spines[['top','right']].set_visible(False)
ax.spines[['bottom','left']].set_position('zero')

# Pfeile für die Achsen
ax.plot((1),(0), ls="", marker= ">", ms=7, color="k", transform=ax.get_yaxis
ax.plot((0),(1), ls="", marker= "^", ms=7, color="k", transform=ax.get_xaxis

# Achsenlänge und Beschriftung
ax.set_xlim(a,b)
ax.set_ylim(c, d)
ax.set_xlabel("x", loc="right")
ax.set_ylabel("f(x)", loc="top", rotation=0)

# Kästchen
ax.grid(linestyle="-", which="major",linewidth=0.7, zorder=-10)
ax.grid(linestyle="-", which="minor",linewidth=0.5, zorder=-10)

# Plot der Funktion
ax.plot(x,y1, zorder=10)
#ax.plot(x,y2, zorder=10)
#plt.show()
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0x12fdf1400>]
```



Wertetabelle:

x	-2	-1	0	1
$f(x)$	0,135	0,368	1	e

Fragen:

- Welche Auswirung hat es , wenn man jeden Funktionswert mit -1 ?

x	-2	-1	0	1
	0,135	0,368	1	e

x	-2	-1	0	1
$f(x)$				
$-f(x)$	-0,135	-0,368	-1	$-e$

=> Die y-Koordinaten aller Punkte des Grapen werden negativ. => Spiegelung des Funktionsgraphen an der x-Achse.

- Welche Auswirung hat es , wenn man die Funktionsvariable mit -1 multipliziert?

x	-2	-1	0	1
$f(x)$	0,135	0,368	1	e
$f(-x)$	7,389	e	1	0,386

=> Alle Punkte des Grahen erhalten die y-Koordinaten ihrer negativen Pendants. => Spiegelung des Funktionsgraphen an der y-Achse.

- Welche Auswirung hat es , wenn man die Funktionsvariable mit -1 multipliziert und den Funktionswert auch mnit -1?

x	-2	-1	0	1
$f(x)$	0,135	0,368	1	e
$f(-x)$	-7,389	-e	-1	-0,386

=> Alle Punkte des Grahen erhalten die y-Koordinaten ihrer negativen Pendants.

=> Alle y-kooridnaten der Punkte erhalten das entegegensgesetzte Vorzeichen.

=> Spiegelung des Funktionsgraphen am Ursprung $O(0|0)$

Satz:
Der Graph der Funktion g entsteht aus dem Graphen der Funktion f durch

- $g(x) = f(-x)$ mit einer Spiegelung an der y-Achse.
- $g(x) = -f(x)$ mit einer Spiegelung an der x-Achse.
- $g(x) = -f(-x)$ mit einerSpiegelung am Ursprung $O(0|0)$

Nachweis einer Achsensymmetrie zur y-Achse bzw. einer Punktsymmetrie zum Ursprung:

Satz:

Der Graph einer Funktion f ist genau dann

- achsensymmetrisch zur y-Achse, wenn für alle $x \in D_f$ gil: $f(-x) = f(x)$
- punktsymmetrisch zum Ursprung, wenn für alle $x \in D_f$ gil: $f(-x) = -f(x)$

Beispiel:

$$\begin{aligned} f(x) &= x \cdot \sin(x) \\ f(-x) &= -x \cdot \sin(-x) \\ &= -(x \cdot \sin(-x)) \\ &= -(x \cdot (-\sin(x)) \\ &= x \cdot \sin(x) \end{aligned}$$

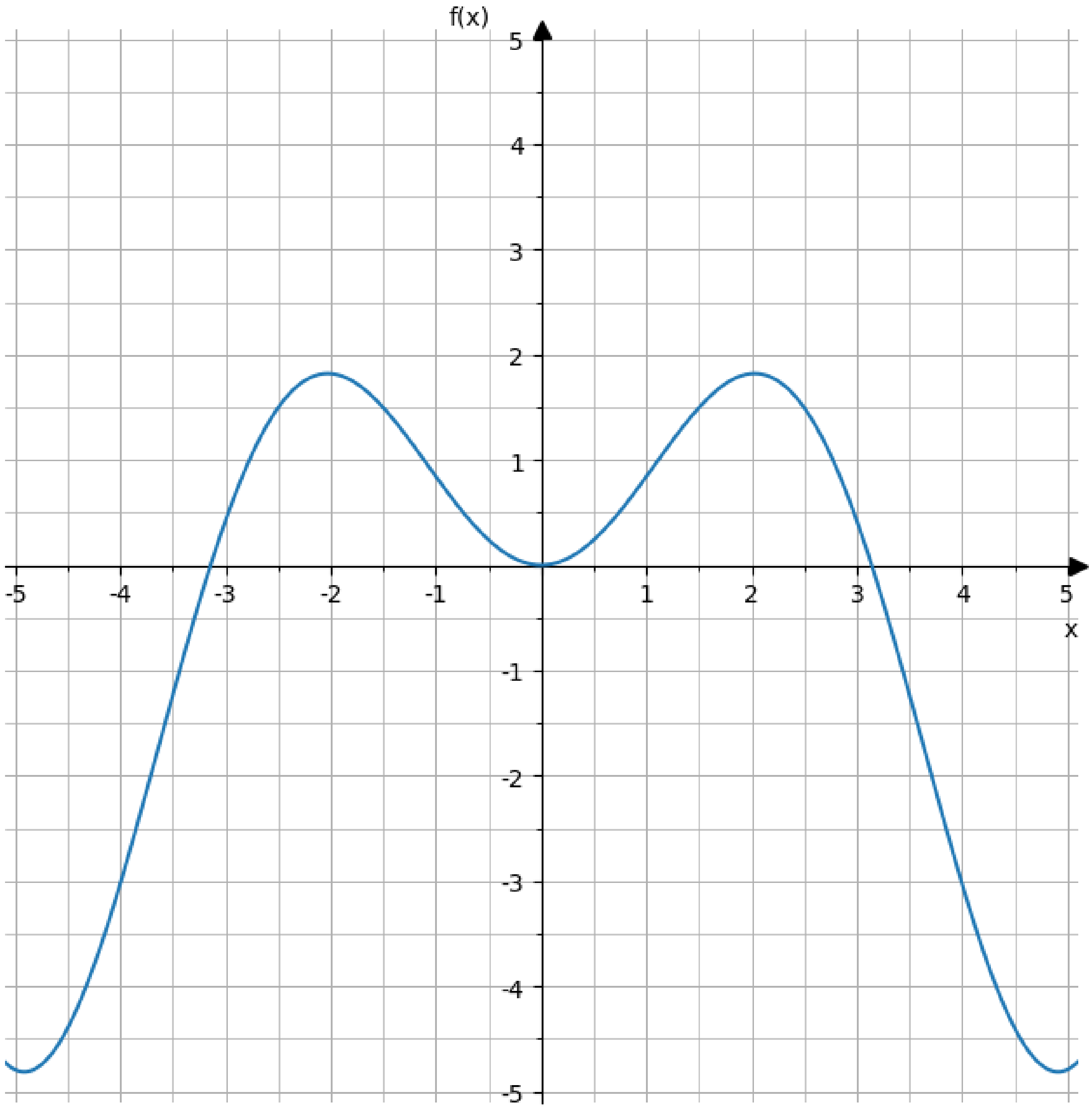
*

⇒ Achsensymmetrie zur y-Achse.

In [11]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatt

# Defintionsmenge und Funktion
# ----------------------------
a= -5.1 # untere x-Intervallgrenze
b= 5.1 # obere x-Intervallgrenze
c = -5.1# untere y-Intervallgrenze
d = 5.1 # obere y-Intervallgrenze
x = np.linspace(a, b,1000)
y1= x*np.sin(x)
# ----------------------------

# Einstellung des Graphen
fig=plt.figure(figsize=(8,8))
ax= fig.add_subplot(1,1,1, aspect =1)

# Definiton der Haupteinheiten, reele Zahlen ohne die 0
def major_tick(x, pos):
    if x==0:
        return ""
    return int(x)

# Achsenskalierung
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.xaxis.set_minor_locator(AutoMinorLocator(2))
ax.yaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_minor_locator(AutoMinorLocator(2))
ax.xaxis.set_major_formatter(FuncFormatter(major_tick))
ax.yaxis.set_major_formatter(FuncFormatter(major_tick))

# Position der Achsen im Schaubild
ax.spines[['top','right']].set_visible(False)
ax.spines[['bottom','left']].set_position('zero')

# Pfeile für die Achsen
ax.plot((1),(0), ls="", marker= ">", ms=7, color="k", transform=ax.get_yaxis
ax.plot((0),(1), ls="", marker= "^", ms=7, color="k", transform=ax.get_xaxis

# Achsenlänge und Beschriftung
ax.set_xlim(a,b)
ax.set_ylim(c, d)
ax.set_xlabel("x", loc="right")
ax.set_ylabel("f(x)", loc="top", rotation=0)

# Kästchen
ax.grid(linestyle="-", which="major",linewidth=0.7, zorder=-10)
ax.grid(linestyle="-", which="minor",linewidth=0.5, zorder=-10)

# Plot der Funktion
```

```
ax.plot(x,y1, zorder=10)
#plt.show()
```

Out[11]: [<matplotlib.lines.Line2D at 0x1383d3a40>]



In [ ]: