---{ Trigonometria }---

Funções Trigonométricas

Prof. Eduardo Ono

Sumário

In [7]:
```
import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = [6, 4]
```

1. Objetivos

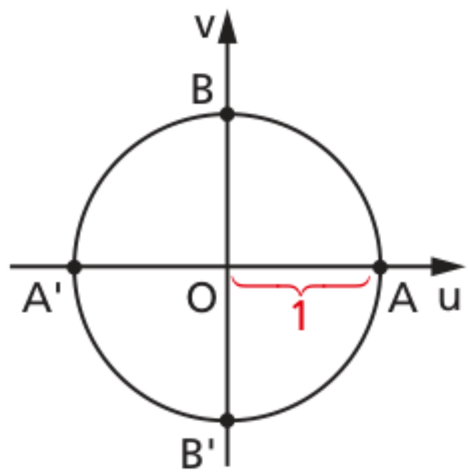
- Conceituar as funções trigonométricas;
- Definir as principais funções trigonométricas: Seno, Cosseno e Tangente;
- Construir os gráficos das principais funções trigonométricas;
- Aplicar os conceitos na resolução de problemas.

2. Funções Trigonométricas

2.1. Definição

Funções trigonométricas são aplicações de $\mathbb{R}$ sobre o ciclo trigonométrico λ, onde podemos associar a cada número real x um único ponto P de λ do seguinte modo:

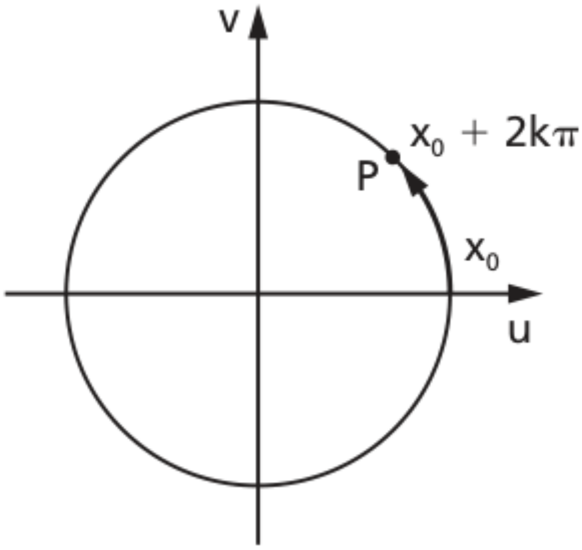
1. Se $x = 0$, então P coincide com A;
2. Se $x > 0$, então realizamos a partir de A um percurso de comprimento x, no sentido anti-horário, e marcamos P como ponto final do percurso;
3. Se $x < 0$, então realizamos a partir de A um percurso de comprimento $|x|$, no sentido horário. O ponto final do percurso é P.



Fonte: Iezzi

O ponto P é a imagem dos elementos do conjunto

$$\{x \in \mathbb{R} \mid x = x_0 + 2k\pi,\ k \in \mathbb{Z}\}$$



Fonte: Iezzi

3. Função Seno: $y = \text{sen } x$

3.1. Definição

Denominamos **função seno** a função $f : \mathbb{R} \longrightarrow \mathbb{R}$ que associa a cada real x o real $OP_1 = \text{sen } x$, isto é:

$$f(x) = \text{sen } x$$

3.2. Gráfico

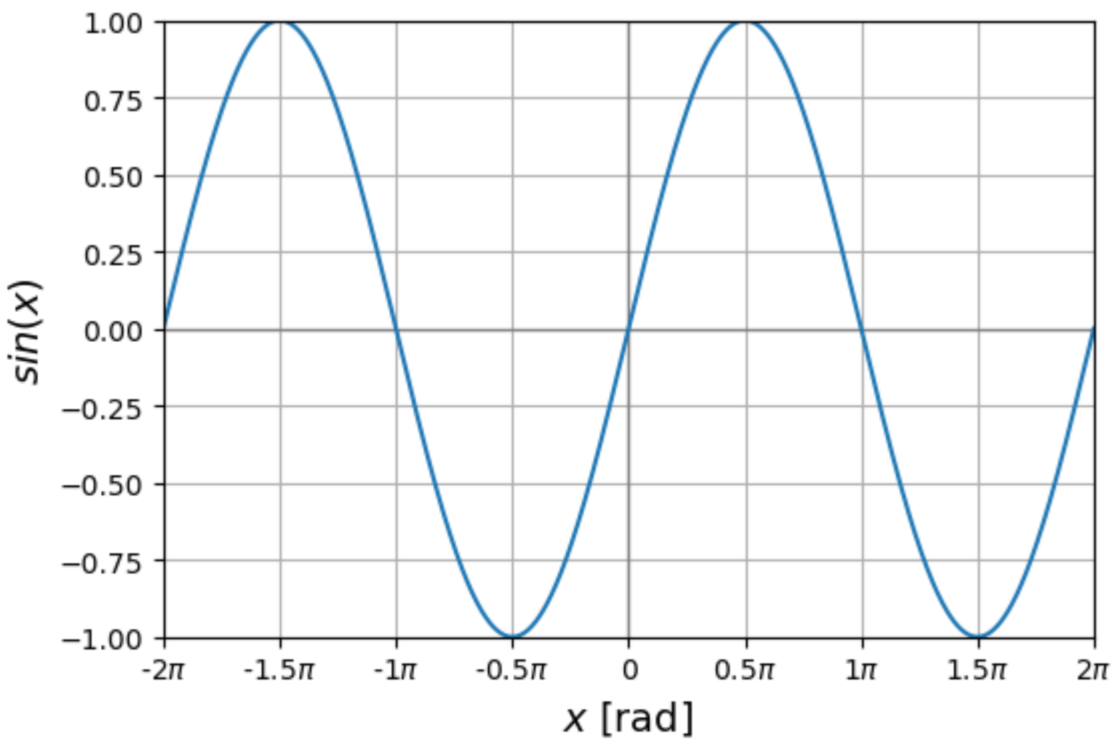
In [6]:

```python
import numpy as np
import math

x_min = -2 * math.pi
x_max = 2 * math.pi
x_step = (x_max - x_min) / 1000
y_min = -1
y_max = 1
plt.xlabel("$x$ [rad]", fontsize=14)
plt.ylabel("$sin(x)$", fontsize=14)
x_values = np.arange(x_min, x_max + x_step, x_step)
y_values = np.sin(x_values)

plt.grid()
plt.axhline(y=0.0, color='grey', linestyle='-', linewidth=1)
plt.axvline(x=0.0, color='grey', linestyle='-', linewidth=1)
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
radian_multiples = np.arange(x_min / math.pi, x_max / math.pi + 0.5, 0.5) # [-2,
radians = [n * math.pi for n in radian_multiples]
radian_labels = []
for n in radian_multiples:
    label = '0' if n == 0 else rf'{int(n)}$\pi$' if n == int(n) else rf'{n}$\pi$
    radian_labels.append(label)

plt.xticks(radians, radian_labels)
plt.plot(x_values, y_values)
plt.show()
```



4. Função Cosseno: $y = \cos x$

4.1. Definição

Denominamos **função cosseno** a função $f : \mathbb{R} \longrightarrow \mathbb{R}$ que associa a cada real x o real $OP_2 = \cos x$, isto é:

$$f(x) = \cos x$$

4.2. Gráfico

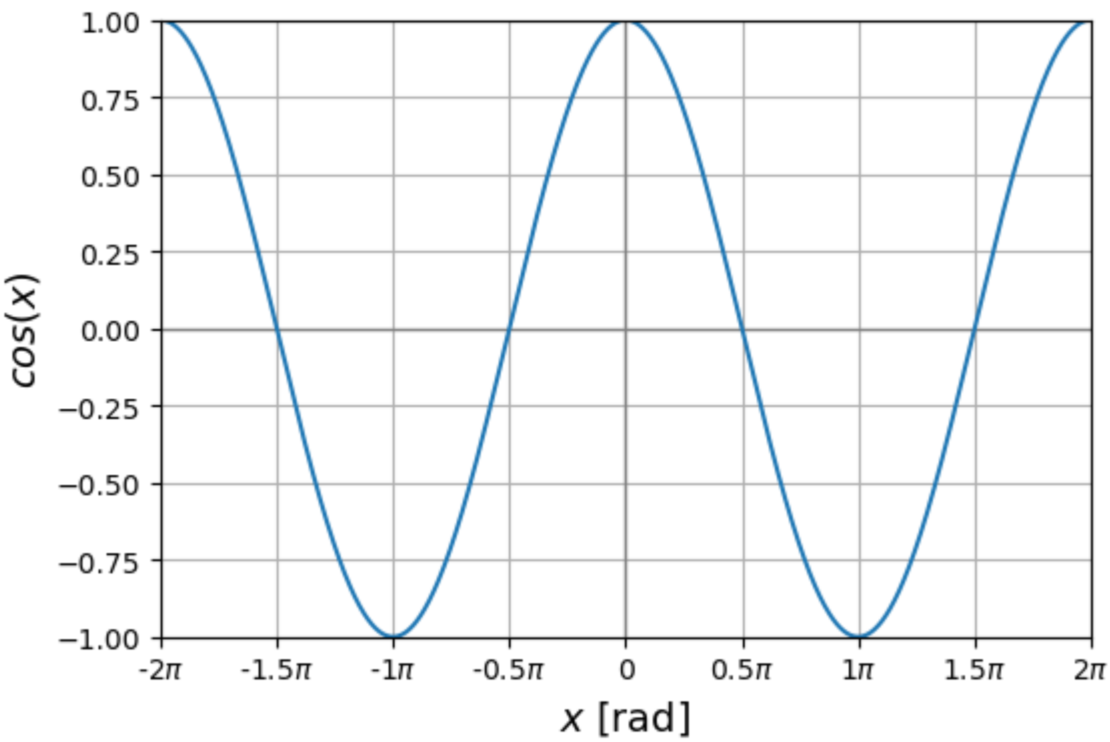
In [5]:

```
import numpy as np
import math

x_min = -2 * math.pi
x_max = 2 * math.pi
x_step = (x_max - x_min) / 1000
x_values = np.arange(x_min, x_max + x_step, x_step)
y_values = np.cos(x_values)

plt.grid()
plt.axhline(y=0.0, color='grey', linestyle='-', linewidth=1)
plt.axvline(x=0.0, color='grey', linestyle='-', linewidth=1)
plt.xlim(x_min, x_max)
plt.xlabel("$x$ [rad]", fontsize=14)
plt.ylabel("$cos(x)$", fontsize=14)
plt.ylim(-1, 1)
radian_multiples = np.arange(x_min / math.pi, x_max / math.pi + 0.5, 0.5) # [-2,
radians = [n * math.pi for n in radian_multiples]
radian_labels = []
for n in radian_multiples:
    label = '0' if n == 0 else rf'{int(n)}$\pi$' if n == int(n) else rf'{n}$\pi$
    radian_labels.append(label)

plt.xticks(radians, radian_labels)
plt.plot(x_values, y_values)
plt.show()
```



5. Função Tangente: $y = \text{tg } x$

Denominamos **função tangente** a função $f : D \longrightarrow \mathbb{R}$, que associa a cada real x, $x \neq \frac{\pi}{2} + k\pi$, o real $AT = \text{tg } x$, isto é:

$$f(x) = \text{tg } x$$

5.1. Propriedades

1. O domínio da função tangente é $D = \left\{ x \in \mathbb{R} \mid x \neq \frac{\pi}{2} + k\pi,\ k \in \mathbb{Z} \right\}$.

2. A imagem da função tangente é $\mathbb{R}$, isto é, para todo y real existe um x real tal que $\text{tg } x = y$. De fa

3. A função tangente é periódica e seu período é π.

$$\text{tg } x = \text{tg } (x + k\pi)$$

5.2. Gráfico

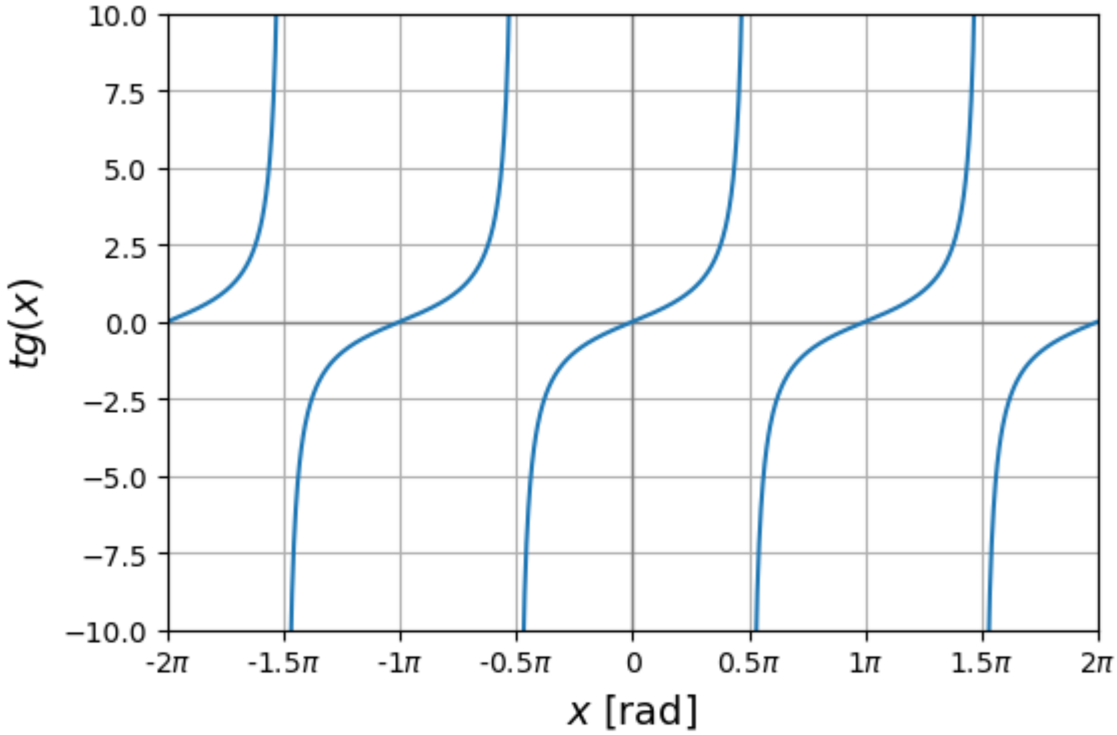
In [85]:

```
import numpy as np
import math

x_min = -2 * math.pi
x_max = 2 * math.pi
x_step = (x_max - x_min) / 1000
x_values = np.arange(x_min, x_max + x_step, x_step)
y_values = np.tan(x_values)
y_values[:-1][np.diff(y_values) < 0] = np.nan

plt.grid()
plt.axhline(y=0.0, color='grey', linestyle='-', linewidth=1)
plt.axvline(x=0.0, color='grey', linestyle='-', linewidth=1)
plt.xlim(x_min, x_max)
plt.xlabel("$x$ [rad]", fontsize=14)
plt.ylabel("$tg(x)$", fontsize=14)
plt.ylim(-10, 10)
radian_multiples = np.arange(x_min / math.pi, x_max / math.pi + 0.5, 0.5) # [-2,
radians = [n * math.pi for n in radian_multiples]
radian_labels = []
for n in radian_multiples:
    label = '0' if n == 0 else rf'{int(n)}$\pi$' if n == int(n) else rf'{n}$\pi$
    radian_labels.append(label)

plt.xticks(radians, radian_labels)
plt.plot(x_values, y_values)
plt.show()
```



10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
tg(x)
-2π
-1.5π
-1π
-0.5π
0
0.5π
1π
1.5π
2π
x [rad]