Piecewise Function

In [1]:
```
import sympy as sp

x = sp.symbols("x")
p = sp.Piecewise((0, x < -1), (x**2, x <= 1), (sp.log(x), True))
p
```

Out[1]:
$$\begin{cases} 0 & \text{for}\: x < -1 \\ x^{2} & \text{for}\: x \leq 1 \\ \log{\left(x \right)} & \text{otherwise} \end{cases}$$

In [4]:
```
p = sp.Piecewise((0, x < 0), (1, x < 1), (2, True))
p
# sp.piecewise_exclusive(p)
```

Out[4]:
$$\begin{cases} 0 & \text{for}\: x < 0 \\ 1 & \text{for}\: x < 1 \\ 2 & \text{otherwise} \end{cases}$$

In [5]:
```
p.integrate(x)  # continuous antiderivative
```

Out[5]:
$$\begin{cases} 0 & \text{for}\: x < 0 \\ x & \text{for}\: x < 1 \\ 2 x - 1 & \text{otherwise} \end{cases}$$

In [13]:
```
p.piecewise_integrate(x)  # piecewise antiderivative
```

Out[13]:
$$\begin{cases} 0 & \text{for}\: x < 0 \\ x & \text{for}\: x < 1 \\ 2 x & \text{otherwise} \end{cases}$$

Limit

In [6]:
```
x = sp.symbols("x")

sp.limit(sp.sin(x) / x, x, 0)
```

Out[6]: 1

In [7]:
```
sp.limit(1 / x, x, 0)  # the default direction is right and can be specif
sp.limit(1 / x, x, 0, dir="+")
```

Out[7]: ∞

In [8]:
```
# sp.limit(1 / x, x, 0, dir="-")
sp.limit(1 / x, x, 0, dir="+-")
```

Out[8]: $\tilde{\infty}$

In [9]:
```
print(sp.zoo.__doc__)
```

```
Complex infinity.

    Explanation
    ===========

    In complex analysis the symbol `\tilde\infty`, called "complex
    infinity", represents a quantity with infinite magnitude, but
    undetermined complex phase.

    ComplexInfinity is a singleton, and can be accessed by
    ``S.ComplexInfinity``, or can be imported as ``zoo``.

    Examples
    ========

    >>> from sympy import zoo
    >>> zoo + 42
    zoo
    >>> 42/zoo
    0
    >>> zoo + zoo
    nan
    >>> zoo*zoo
    zoo

    See Also
    ========

    Infinity
```

In [10]:

```
sp.Limit(sp.sin(x) / x, x, 0)
```

Out[10]:

$$\lim_{x \to 0^+}\left(\frac{\sin\left(x\right)}{x}\right)$$

In [11]:

```
k, n = sp.symbols("k n", integer=True)
A = sp.Sum(sp.Integer(-1) ** (k + 1) / k, (k, 1, n))
A
# A.doit()
```

Out[11]:

$$\sum_{k=1}^{n} \frac{\left(-1\right)^{k+1}}{k}$$

In [12]:

```
i, n, m = sp.symbols("i n m", integer=True)

sp.Sum(2 * i - 1, (i, 1, n))
# sp.Sum(2 * i - 1, (i, 1, n)).doit()
```

Out[12]:

$$\sum_{i=1}^{n} \left(2i - 1\right)$$

In [12]:

```
i, n, m = sp.symbols("i n m", integer=True)

sp.summation(2 * i - 1, (i, 1, n))
```

Out[12]: n^2

In [49]:
```
sp.Sum(x**n / sp.factorial(n), (n, 0, sp.oo))
# sp.summation(x**n / sp.factorial(n), (n, 0, sp.oo))
```

Out[49]:

$$\sum_{n=0}^{\infty} \frac{x^n}{n!}$$

In [17]:
```
sp.summation(i, (i, 0, n), (n, 0, m))
# sp.Sum(i, (i, 0, n), (n, 0, m))
```

Out[17]:

$$\frac{m^3}{6} + \frac{m^2}{2} + \frac{m}{3}$$

In [28]:
```
k, n = sp.symbols("k n", integer=True, positive=True)
y = sp.symbols("y")

sum_result = sp.summation(sp.binomial(n, k) * x**k * y ** (n - k), (k, 0,
sum_result

# domain = sp.Abs(x) / sp.Abs(y) <=1

# sum_result.refine(domain).simplify()
```

Out[28]:

$$\begin{cases} y^n \left(\frac{x}{y} + 1\right)^n & \text{for } \left|\frac{x}{y}\right| \leq 1 \\ \sum_{k=0}^{n} x^k y^{-k+n} \binom{n}{k} & \text{otherwise} \end{cases}$$

In [20]:
```
sp.product(i, (i, 1, k))
# sp.Product(i, (i, 1, k))
```

Out[20]: $k!$

In [23]:
```
sp.product(i, (i, 1, k), (k, 1, n))
```

Out[23]:

$$\prod_{k=1}^{n} k!$$

Sequences

In [53]:
```
n = sp.Symbol("n")

s = sp.SeqFormula(n**2)
# s = sp.SeqFormula(n**2, (n, 0, 15))

s
# s[:]
# s.formula
# s.coeff(3)
```

Out[53]: $[0, 1, 4, 9, \ldots]$

In [42]:
```
from sympy.series.sequences import RecursiveSeq

y = sp.Function("y")
n = sp.symbols("n")
```

```python
fib = RecursiveSeq(y(n - 1) + y(n - 2), y(n), n, initial=[0, 1])
fib
# fib.recurrence
# fib.degree
```

Out[42]: $\operatorname{RecursiveSeq}(y(n-2)+y(n-1), y(n), n, (0,\ 1), 0)$

In [40]:
```python
for idx, val in zip(range(10), fib):
    print(idx, val)
```

```
0 0
1 1
2 1
3 2
4 3
5 5
6 8
7 13
8 21
9 34
```

Series

In [3]:
```python
f = sp.tan(x)
sp.series(f, x, 2, 6)
# sp.series(f, x, 2, 6, dir="+")
```

Out[3]:
$$\tan(2)+\left(1+\tan^2(2)\right)(x-2)+(x-2)^2\left(\tan^3(2)+\tan(2)\right)$$
$$+(x-2)^3\left(\frac{1}{3}+\frac{4\tan^2(2)}{3}+\tan^4(2)\right)+(x-2)^4\left(\tan^5(2)+\frac{5\tan^3(2)}{3}+\frac{2\tan(2}{3}\right.$$
$$+(x-2)^5\left(\frac{2}{15}+\frac{17\tan^2(2)}{15}+2\tan^4(2)+\tan^6(2)\right)+O\left((x-2)^6; x\to 2\right)$$

In [4]:
```python
sp.series(f, x, 2, 6, dir="-")
```

Out[4]:
$$\tan(2)+(2-x)\left(-\tan^2(2)-1\right)+(2-x)^2\left(\tan^3(2)+\tan(2)\right)$$
$$+(2-x)^3\left(-\tan^4(2)-\frac{4\tan^2(2)}{3}-\frac{1}{3}\right)+(2-x)^4\left(\tan^5(2)+\frac{5\tan^3(2)}{3}+\frac{2\tan}{3}\right.$$
$$+(2-x)^5\left(-\tan^6(2)-2\tan^4(2)-\frac{17\tan^2(2)}{15}-\frac{2}{15}\right)+O\left((x-2)^6; x\to 2\right)$$

In [2]:
```python
# formal power series
sp.fps(sp.tanh(x)).series(n=10)
```

Out[2]:
$$x-\frac{x^3}{3}+\frac{2x^5}{15}-\frac{17x^7}{315}+\frac{62x^9}{2835}+O\left(x^{10}\right)$$

In [44]:
```python
sp.fps(sp.ln(1 + x)).series()
# sp.fps(sp.ln(1 + x)).series().truncate(10)
```

Out[44]: $$\sum_{k=1}^{\infty} - \frac{(-1)^{-k} x^k}{k}$$

In [3]:
```
n = sp.symbols("n", integer=True, positive=True)
# sp.fps(x**n * sp.sin(x**2), x)
sp.fps(x**n * sp.sin(x**2), x).series(x)
```

Out[3]: $$x^{n+2} + \left(\sum_{k=3}^{\infty} \begin{cases} \frac{\left(-\frac{1}{4}\right)^{\frac{k}{4}-\frac{1}{2}} x^{k+n}}{\left(\frac{3}{2}\right)^{\left(\frac{k}{4}-\frac{1}{2}\right)} \left(\frac{k}{4}-\frac{1}{2}\right)!} & \text{for } k \bmod 4 = 2 \\ 0 & \text{otherwise} \end{cases} \right)$$

In [8]:
```
sp.fps(sp.ln(1 + x)).series().truncate(10).integrate()
```

Out[8]: $$\frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{12} - \frac{x^5}{20} + \frac{x^6}{30} - \frac{x^7}{42} + \frac{x^8}{56} - \frac{x^9}{72} + \frac{x^{10}}{90} + O\left(x^{11}\right)$$