In [1]:
```julia
using SymPy
```

問 2 (2)

In [2]:
```julia
let
# let の中での @variables 宣言は外に漏れないので、あとで変数を普通に使える.

  @syms x y

  f = x * sin(x*y) # 関数の定義

  fx = diff( f, x ) # 偏微分した結果を展開
  fy = diff( f, y )

  fxx = diff( fx, x )
  fxy = diff( fx, y )
  fyy = diff( fy, y )

  display(fxx)
  display(fxy)
  display(fyy)

end
```

$$-xy^2 \sin(xy) + 2y\cos(xy)$$

$$-x^2y\sin(xy) + 2x\cos(xy)$$

$$-x^3\sin(xy)$$

問 4

連鎖率により $\partial_x = r_x(\partial_r) + \theta_x(\partial_\theta)$ であるので、∂_x は r_x と θ_x が分かれば良い. ∂_y も同様だ.

幸い、$r = \sqrt{x^2+y^2}$, $\theta = \arctan(y/x)$ とわかっているので、

$r_x = \cos(\theta)$,
$r_y = \sin(\theta)$,
$\theta_x = -\sin(\theta)/r$,
$\theta_y = \cos(\theta)/r$

とわかる.

よって、

$\partial_x = \cos(\theta)\partial_r - \frac{\sin(\theta)}{r}\partial_\theta$,
$\partial_y = \sin(\theta)\partial_r + \frac{\cos(\theta)}{r}\partial_\theta$

と書ける.

だから、$\partial_{xx} = \partial_x \partial_x$ として計算すれば良いし、∂_{yy} も同様だ.

In [3]:
```
let

@syms r,t, f()

F = f(r,t)

Dx(g) = simplify( cos(t) * diff(g,r) - (sin(t)/r) * diff(g,t) )
Dy(g) = simplify( sin(t) * diff(g,r) + (cos(t)/r) * diff(g,t) )

fxx = Dx( Dx(F) )
fyy = Dy( Dy(F) )

lf = simplify( fxx + fyy )
display(lf)

end
```

$$\frac{\partial^2}{\partial r^2} f(r,t) + \frac{\frac{\partial}{\partial r} f(r,t)}{r} + \frac{\frac{\partial^2}{\partial t^2} f(r,t)}{r^2}$$

問 5 (2)

とりあえず、通常のTaylor 展開の 2階微分までで打ち切った形式を出してみよう.

In [5]:
```
let

@syms x y

function taylor2(expr, x, y, a, b, N)
    ans = 0

    for i in 0:N
        for j in 0:(N-i)
            deriv = diff(expr, x, i, y, j)
            coeff = deriv.subs(Dict(x => a, y => b)) / (factorial(i) * factorial
            ans += coeff * (x - a)^i * (y - b)^j
        end
    end

    return simplify(expand(ans))
end

f = cos(x + 2*y)

T2 = taylor2(f, x, y, 0, 0, 2)
display(T2)

end
```

$$-\frac{x^2}{2} - 2xy - 2y^2 + 1$$

結果から容易に判断できるように、

- 0次項は 1.

- 1次項は 0、つまり、無い.
- 2次項は(剰余項でない場合は) $-x^2/2 - 2xy - 2y^2$.

これは計算に手を抜いた形なので、2次項を剰余項として正しく計算しないといけない． 真面目に計算するだけだ． それはおおよそ、上の2次項と似た形のはずだ.

In [ ]: