Recall from Calculus, the linear approximation:

\[ L(x) = y_0 + y'(x_0)(x - x_0) \]

for a function \( y(x) \).

Now suppose \( y(x) \) is a solution of a 1st order I. V. P.

\[
\begin{cases}
  y' = f(x, y) \\
  y(x_0) = y_0 
\end{cases}
\]

Then \( L(x) = y_0 + f(x_0, y_0) \Delta x \)

and \( L(x) \) approximates the solution \( y(x) \) well for small \( \Delta x \).
we want to approximate 
\[ y \] the soln to 
\[
\begin{align*}
y' &= f(x,y) \\
y(x_0) &= y_0
\end{align*}
\]
"far away" from \((x_0, y_0)\)

**strategy:** At steps of \(\Delta x\),...

**re-anchor the tangent line approximation — again and again —**

**Benefit**

**The problem...**
2.b. Program

\[ y_{n+1} = y_n + f(x_n, y_n) \cdot h \]

\[ y_1 = y_0 + f(x_0, y_0) \cdot h \]
\[ y_2 = y_1 + f(x_1, y_1) \cdot h \]
\[ y_3 = y_2 + f(x_2, y_2) \cdot h \]
\[ x_{n+1} - x_n = \Delta x = h \]

continue until you get where you want to be.
2.6

(4) **Ex:** Use Euler Method to approximate \( y(1.5) \) if \( y \) is the solution of

\[
\begin{align*}
  y' &= 2xy \\
  y(1) &= 1
\end{align*}
\]

a) Let \( h = 0.1 \) (See attached example)

b) Let \( h = 0.05 \) (I will not do this.)
Section 2.6 A Numerical Method

Exercise 4

> reset;

> x := 1; y := 1; h := 0.1;

\[ x := 1 \]
\[ y := 1 \]
\[ h := 0.1 \]

> dydx := (x, y) -> 2 * x * y;

\[ dydx := (x, y) \rightarrow 2xy \]

> for i from 1 to 5 do
>   y := y + h * dydx(x, y);
>   x := x + h;
>   printf("x=%f Approx y=%f True Value=%f Abs Error=%f\n", x, y, exp(x^2)/exp(1), abs(y-exp(x^2)/exp(1)));
> od:

x=1.100000 Approx y=1.200000 True Value=1.233678 Abs Error=0.033678
x=1.200000 Approx y=1.464000 True Value=1.552707 Abs Error=0.088707
x=1.300000 Approx y=1.815360 True Value=1.993716 Abs Error=0.178356
x=1.400000 Approx y=2.287354 True Value=2.611696 Abs Error=0.324343
x=1.500000 Approx y=2.927813 True Value=3.490343 Abs Error=0.562530

> dsolve({diff(z(t), t) = 2*t*z(t), z(1) = 1});

\[ z(t) = \frac{e^t}{c} \]

> diff(z(t), t) = 2*t*z(x);

\[ \frac{d}{dt} z(t) = 2t z(1.5) \]