Using Horner's technique to evaluate a *numeral* (string of digits) into a *number*

Consider the (base-10, or *decimal)* numeral: **8,192**

8,192

$$(((0 \cdot 10 + \mathbf{8}) \cdot 10 + \mathbf{1}) \cdot 10 + \mathbf{9}) \cdot 10 + \mathbf{2}$$

initial value of 0 ⇒ accumulate running left-to-right

$$0 \Rightarrow 0 \Rightarrow 8 \Rightarrow 80 \Rightarrow 81 \Rightarrow 810 \Rightarrow 819 \Rightarrow 8190 \Rightarrow 8192$$

$\cdot 10 \quad +8 \qquad \cdot 10 \quad +1 \qquad \cdot 10 \quad +9 \qquad \cdot 10 \quad +2$

Each digit d_i is "replaced by" (or "interpreted as") the operation: "multiply by 10 and add d_i"
(more generally and more accurately: "multiply by the *base* and add the *value of* the digit d_i")

These operations are then applied, starting with 0 as the initial value, from left to right.
(Of course, since $0 \cdot 10 = 0$, and $0 + d_0 = d_0$, you can also just start with initial value = d_0 and save two operations.)