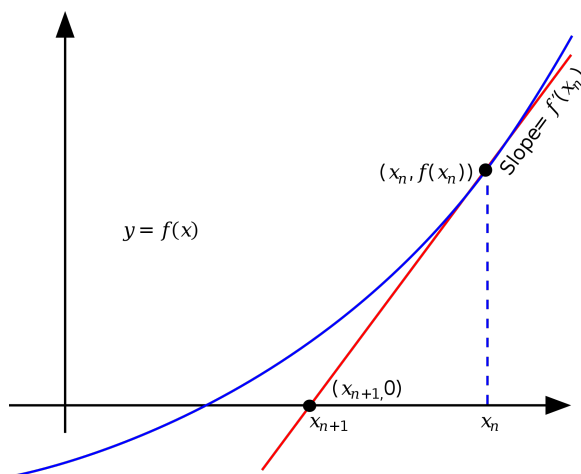


Basic Python - While-Loop Exercise

Hoàng-Nguyên Vũ

1. Mô tả:

- Phương pháp Newton (Newton's Method), còn được gọi là phương pháp Newton-Raphson, là một phương pháp số học để tìm gần đúng của các nghiệm của một hàm số thực. Cụ thể, nó thường được sử dụng để tìm gần đúng của các nghiệm của phương trình $f(x) = 0$.
- Ngoài ứng dụng trong tìm nghiệm của một hàm số, phương pháp Newton còn có ứng dụng trong máy học (Machine learning) trong việc tìm nghiệm của đạo hàm của hàm loss. Tuy nhiên đây là phương pháp không phổ biến bằng thuật toán gradient descent.



- Ở bài này, chúng ta sẽ dùng phương pháp Newton để tính căn bậc hai cho một số dương a . Chúng ta thực hiện các bước sau:
 - Khởi tạo giá trị $x_0 = a$, $n = 0$ và cho trước giá trị ε (thực ra x_0 có thể nhận bất kỳ giá trị dương nào). Tiếp đó, ta sẽ đi xây dựng hàm $f(x) = x^2 - a$. Ở đây, ta xem x_n (ở bước hiện tại đang là x_0) chính là lời giải cho bài toán tính căn bậc hai của a .
 - Cải thiện xấp xỉ x_n bằng xấp xỉ x_{n+1} theo công thức tổng quát như sau:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$
 - So sánh xấp xỉ x_{n+1} và x_n . Nếu $|x_{n+1} - x_n| < \varepsilon$, ta sẽ dừng việc cải thiện xấp xỉ, và trả về kết quả căn bậc hai của a là x_{n+1} . Ngược lại thực hiện tiếp bước (b) với $n = n + 1$.

2. **Bài tập:** Cài đặt hàm `find_squared_root(a)` tìm căn bậc hai cho một số a bất kì với $\varepsilon = 0.001$.

```
1 def find_squared_root(a):  
2     """Find the squared root of number a"""  
3     EPSILON = 0.001  
4  
5     #TODO: Your code here  
6
```

Ví dụ:

- Test case 1: `find_squared_root(2)` → 1.4142135623746899
- Test case 2: `find_squared_root(3)` → 1.7320508100147276