Inserting Text Into a Ruby String

660710616 วรรณวิไล เกิดศิริ

Inserting Text Into a Ruby String

ในภาษา Ruby สามารถแทรกข้อความเพิ่มได้ โดยการใช้ method ที่ชื่อว่า insert()

method insert() เป็น method ของคลาส String ในภาษา Ruby ซึ่งใช้สำหรับแทรกข้อความใหม่ลงใน string ตรงตำแหน่งที่ต้องการ

Syntax

```
str.insert(index, other_str)
```

str คือ string เดิม
index คือ ตำแหน่งที่ต้องการจะแทรก
แบ่งออกเป็น 2 กรณี
- index เป็น **จำนวนเต็มบวก** จะเริ่มแทรกจากด้านหน้าของ string
- index เป็น **จำนวนเต็มลบ** จะเริ่มแทรกจากส่วนท้ายของ string

other_str คือ ข้อความที่ต้องการจะแทรกเข้าไป

Example

Example 1 : กรณี index เป็นจำนวนเต็มบวก

```
str = "Helloworld"
str.insert(5, "my")
```

Output :

Hellomyworld

Example

Example 2 : กรณี index เป็น**จำนวนเต็มลบ**

```
str = "Helloworld"
str.insert(-2, "my")
```

Output :

Hellowormyld

เปรียบเทียบกับภาษา
C / Java / Python

ภาษา C

ภาษา C ไม่มี method insert() โดยตรงเหมือนกับ Ruby จึงต้องสร้าง function ในการแทรกข้อความขึ้นเอง

Syntax

```
void insertString(char *str, const char *insert, int index)
```

str คือ ข้อความเดิม
insert คือ ข้อความที่ต้องการแทรก
index คือ ตำแหน่งใน str ที่ต้องการแทรก insert เข้าไป

Example

กรณี index เป็นจำนวนเต็มบวก

```
#include <stdio.h>
#include <string.h>

void insertString(char *str, const char *insert, int index) {
    int lenStr = strlen(str);
    int lenInsert = strlen(insert);

    memmove(str + index + lenInsert, str + index, lenStr - index + 1);

    memcpy(str + index, insert, lenInsert);
}

int main() {
    char str[50] = "Helloworld";
    char insert[] = "my";
    int index = 5;

    insertString(str, insert, index);
    printf("%s\n", str);

    return 0;
}
```

Output :

Hellomyworld

- ใช้ **function memmove()** ในการขยับข้อความเดิม
- **function memcpy()** ในการแทรกข้อความใหม่ลงไปในข้อความเดิม

ภาษา Java

เนื่องจาก String **ในภาษา Java ไม่สามารถเปลี่ยนค่าได้โดยตรง** จึงต้องใช้ method insert() ของ class StringBuffer ใน Java ซึ่งเป็นคลาสที่ใช้จัดการกับ String ที่สามารถแก้ไขได้โดยไม่จำเป็นต้องสร้าง object ขึ้นใหม่ทุกครั้ง

Syntax

```
str.insert(int index, String text)
```

str คือ ข้อความเดิม
index คือ ตำแหน่งที่ต้องการแทรก
text คือ ข้อความที่ต้องการแทรก

Example

กรณี index เป็นจำนวนเต็มบวก

```
public class Main {
    public static void main(String[] args) {
        StringBuffer str = new StringBuffer("Helloworld");
        str.insert(5, "my");
        System.out.println(str);
    }
}
```

Output :

```
Hellomyworld
```

ภาษา Python

เนื่องจาก String ในภาษา Python **ไม่สามารถเปลี่ยนค่าได้โดยตรง ไม่มี method insert() เหมือนกับ Ruby** แต่ สามารถทำได้โดยการตัด string แล้วต่อกลับเข้าด้วยกัน

Syntax

```
new_str = str[:index] + insert_text + str[index:]
```

str[:index] คือ การตัด string ตั้งแต่ตัวแรก ถึงตำแหน่งก่อนหน้า index (* ไม่รวมตำแหน่ง index)
insert_text คือ ข้อความที่ต้องการแทรกเข้าไปใน string
str[index:] คือ การตัดstring ตั้งแต่ตำแหน่ง index ถึงตัวสุดท้าย

Example

กรณี index เป็นจำนวนเต็มบวก

```
str = "Helloworld"
index = 5
insert_text = "my"

new_str = str[:index] + insert_text + str[index:]
print(new_str)
```

Output :

```
Hellomyworld
```

Thank you