Constant Scopes

การกำหนด Constants ในภาษา Ruby

- Ruby Constant Scope กำหนดด้วยชื่อตัวแปรด้วยตัวอักษรพิมพ์ใหญ่ทั้งหมด เป็นค่าที่เมื่อกำหนดแล้วไม่ควรเปลี่ยนแปลง “ไม่ควร” เพราะเนื่องจาก Ruby อนุญาตให้เปลี่ยนค่าได้ แต่ interpreter แต่จะแสดงคำเตือนหากโปรแกรมเปลี่ยนแปลงค่าเพื่อเตือนว่าค่าเดิมที่ถูกกำหนดไว้แล้ว

ตัวอย่างการใช้งาน Ruby

ตัวอย่าง code

```ruby
MY_NAME = "SU"
puts "Welcome to #{MY_NAME}"
MY_NAME = "Silpakorn"
puts "Welcome to #{MY_NAME}"
```

ตัวอย่าง out put

```
Welcome to SU
Welcome to Silpakorn

HelloWorld.rb:3: warning: already initialized constant MY_NAME
HelloWorld.rb:1: warning: previous definition of MY_NAME was here
```

ตัวอย่างการใช้งาน Python

ในภาษา Python ไม่มี constant หากต้องการกำหนดค่าคงที่ ต้องใช้อักษรตัวพิมพ์ใหญ่ตั้งชื่อตัวแปรและกำหนดค่านั้นๆ เนื่องจาก Python ไม่มี constant จึงสามารถแก้ไขค่าได้เสมอ และไม่มี warning อีกด้วย

ตัวอย่าง code

```
1  MY_NAME = "SU"
2  print(f"Welcome to {MY_NAME}")
3  MY_NAME = "Silpakorn"
4  print(f"Welcome to {MY_NAME}")
```

ตัวอย่าง out put

```
Output:

Welcome to SU
Welcome to Silpakorn
```

ตัวอย่างการใช้งาน Java

ในภาษา Java ใช้ final ในการกำหนดค่าคงที่ ค่าของตัวแปรจะไม่เปลี่ยนแปลงหลังจากกำหนดค่าแล้ว ตั้งชื่อโดยการใช้อักษรตัวพิมพ์ใหญ่ทั้งหมด

ตัวอย่าง code

```java
public class Main {
    public static void main(String[] args) {
        final String MY_NAME = "SU";
        System.out.println("Welcome to " + MY_NAME);
        MY_NAME = "Silpakorn";
        System.out.println("Welcome to " + MY_NAME);
    }
}
```

ตัวอย่าง out put

```
Output:

Main.java:5: error: cannot assign a value to final variable MY_NAME
        MY_NAME = "Silpakorn";
        ^
1 error
error: compilation failed
```

ตัวอย่างการใช้งาน

C

ในภาษา C ใช้ const ในการกำหนดค่าคงที่ ค่าของตัวแปรจะไม่เปลี่ยนแปลงหลังจากกำหนดค่าแล้ว

ตัวอย่าง code

```c
#include <stdio.h>
int main() {
    const char* const MY_NAME = "SU";
    printf("Welcome to %s\n", MY_NAME);
    MY_NAME = "Silpakorn";
    printf("Welcome to %s\n", MY_NAME);
    return 0;
}
```

ตัวอย่าง out put

```
Output:

Main.c: In function 'main':
Main.c:5:13: error: assignment of read-only variable 'MY_NAME'
   5 |     MY_NAME = "Silpakorn";
     |             ^
```

Constant Scope ใน Class และ Module

- ค่า Constant ที่ถูกประกาศไว้ใน class หรือ module จะถูกมองเห็นและเข้าถึงได้ภายใน scope ของ class/module นั้นทั้งหมด และถ้าต้องการเรียกใช้นอก class หรือ module ที่กำหนด ต้องใช้ ::

```ruby
class Circle
  PI = 3.14 # ประกาศใน class

  def area(r)
    PI * r * r # ใช้ได้ใน method
  end
end

puts Circle::PI # ใช้จากข้างนอกด้วย scope operator
puts Circle.new.area(5)
```

Python

```python
class Circle:
    PI = 3.14  # constant (convention)

    def area(self, r):
        return Circle.PI * r * r  # เข้าถึง class constant

print(Circle.PI)
c = Circle()
print(c.area(5))
```

Java

```java
public class Circle {
    public static final double PI = 3.14;

    public double area(double r) {
        return PI * r * r; // ใช้ constant
    }

    public static void main(String[] args) {
        System.out.println(Circle.PI);
        Circle c = new Circle();
        System.out.println(c.area(5));
    }
}
```

C

```c
#include <stdio.h>

const double PI = 3.14; // constant

typedef struct {
} Circle;

// function สำหรับ area
double area(double r) {
    return PI * r * r; // ใช้ constant
}

int main() {
    printf("%f\n", PI); // ใช้ constant
    printf("%f\n", area(5)); // เรียก function area
    return 0;
}
```

สรุปการเปรียบเทียบ

ภาษา	เปลี่ยนค่า constant	การประกาศ constant	การเข้าถึงใน method/class	การเข้าถึงจากข้างนอก
Ruby	ได้ (warning)	ตัวอักษรตัวพิมพ์ใหญ่	เรียกใช้โดยตรง	`Circle::PI`
Python	ได้	ตัวอักษรตัวพิมพ์ใหญ่	เรียกใช้ผ่าน class	`Circle.PI`
Java	ไม่ได้	`final`	เรียกใช้โดยตรง	`Circle.PI`
C	ไม่ได้	`const`	ใช้ใน function	`PI`

Thank you