



Anonymous Function in Dart

ฟังก์ชันนิรนาม

Anonymous Function in Dart หรือ พังก์ชันนิรนาม

พังก์ชันโดยส่วนใหญ่จะมีชื่อทั้งนั้น เช่น main(), printElement() ถึงอย่างนั้นคุณก็สามารถสร้างพังก์ชันที่ไม่มีชื่อได้เหมือนกัน หรือที่เรียกว่า anonymous functions บางทีก็เรียก lambda หรือบางก็เรียก closure ซึ่งหน้าตาของพังก์ชันนิรนามก็เหมือนกันพังก์ชันที่มีชื่อ ต่างกันแค่ไม่มีชื่อให้เรียก โดยปกติแล้วเราจะกำหนดมันใส่ตัวแปร หรือส่งไปเป็น Parameter ของพังก์ชันอื่น

Syntax

เปรียบเทียบกับภาษาอื่นๆ

- Dart

```
(parameter_list) {  
    statement(s)  
}
```

- C#

```
(input-parameters) => { <sequence-of-statements> }
```

- Python

```
lambda arguments : expression
```

- JavaScript

```
(function () {  
    // ...  
});
```

- ไม่มี Anonymous Function ในภาษา C หรือ Java

Example :
เขียนฟังก์ชันแบบธรรมดा

```
int add(int x, int y) {  
    return x + y;  
}
```

Example : เขียนฟังก์ชันแบบ
Anonymous Function

```
(int x, int y) {  
    return x + y;  
}
```

ตัวอย่าง

- Dart

```
void main() {  
    var sum = (int x, int y) {  
        return x + y;  
    };  
  
    print(sum(10, 20));  
}
```

output

30

- Python

```
sum = lambda x,y:x+y  
print(sum(10,20))
```

- JavaScript

```
const sum = (x, y) => {  
    return x + y;  
};  
console.log(sum(10, 20));
```



- Dart

```
void show(fn) {  
    for (var i = 0; i < 10; i++) {  
        if (fn(i)) {  
            print(i);  
        }  
    }  
  
void main() {  
    // show even numbers  
    show((int x) {  
        return x % 2 == 0;  
    });  
}
```

output

0
2
4
6
8

ตัวอย่าง

