



Map in Dart

630710643 จิราภา งามจันทร์

Map หรือที่เรียกกันทั่วไปว่า Dictionary หรือ Hash คือชุดของคู่ **Key-Value** ที่ไม่มีการเรียงลำดับ Maps เชื่อมโยงคีย์เข้ากับค่าบางค่าเพื่อให้เรียกค้นได้ง่าย

ในภาษาอื่น ๆ

- ภาษา C ใช้อาร์เรย์เป็นองค์ประกอบและจัดเก็บคู่ Key-Value
- ภาษา Java มี implementation ต่างๆ มากมายให้เลือกใช้ เช่น HashMap, TreeMap, CachMap, EnumMap ฯลฯ
- ภาษา Python สามารถใช้ dictionary เพื่อให้ได้ฟังก์ชันการทำงานของคู่ Key-Value

วิธีการสร้าง Map ในภาษา Dart

ตัวอย่างนี้จะสร้าง Map สำหรับ String และ String
หมายความว่า Key และ Value จะต้องเป็นประเภท
ของสตริง ซึ่งเราสามารถ Map ที่มีชนิดข้อมูลใดก็ได้
ตามที่ต้องการ

```
void main(){  
    Map<String, String> countryCapital = {  
        'USA': 'Washington, D.C.',  
        'India': 'New Delhi',  
        'China': 'Beijing'  
    };  
    print(countryCapital);  
}
```

Output

{USA: Washington, D.C., India: New Delhi, China: Beijing}.

Key จะต้องไม่ซ้ำกัน !!!

วิธีการสร้าง Map ในภาษาอื่น ๆ

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

int main() {
    char *countryNames[] = {"USA", "India", "China"};
    char *countryCapitals[] = {"Washington, D.C.", "New Delhi", "Beijing"};

    int numCountries = sizeof(countryNames) / sizeof(countryNames[0]);

    for (int i = 0; i < numCountries; i++) {
        printf("%s: %s\n", countryNames[i], countryCapitals[i]);
    }

    return 0;
}
```

- **ภาษา C** ตัวอย่างนี้จะใช้อาร์เรย์เพื่อแสดงชื่อประเทศและเมืองหลวง โค้ดจะวนซ้ำอาร์เรย์และแสดงผลประเทศ-เมืองหลวง

- **ภาษา Java** ในตัวอย่างนี้จะใช้คลาส java.util.HashMap เพื่อสร้าง Key-Value Map ซึ่งสามารถเพิ่ม Key-Value โดยใช้ put method

```
def main():
    countryCapital = {
        'USA': 'Washington, D.C.',
        'India': 'New Delhi',
        'China': 'Beijing'
    }
    print(countryCapital)

if __name__ == "__main__":
    main()
```

- **ภาษา Python** ในตัวอย่างนี้จะใช้ Dictionary เพื่อให้ได้ฟังก์ชันการทำงานของคู่ Key-Value

```
import java.util.HashMap;
import java.util.Map;

public class Main {
    public static void main(String[] args) {
        Map<String, String> countryCapital = new HashMap<>();
        countryCapital.put("USA", "Washington, D.C.");
        countryCapital.put("India", "New Delhi");
        countryCapital.put("China", "Beijing");

        System.out.println(countryCapital);
    }
}
```

การเข้าถึงค่า จาก Key

เราสามารถค้นหาค่าของ Map ได้จาก Key ของมัน
ตัวอย่างต่อไปนี้จะแสดงผล Washington, D.C.
จาก Key ของมันคือ USA

```
void main(){  
    Map<String, String> countryCapital = {  
        'USA': 'Washington, D.C.',  
        'India': 'New Delhi',  
        'China': 'Beijing'  
    };  
    print(countryCapital["USA"]);  
}
```

Output

Washington, D.C.

คุณสมบัติของ Map ในภาษา Dart

คุณสมบัติ	การทำงาน
keys	แสดง key ทั้งหมด
values	แสดง value ทั้งหมด
isEmpty	ตรวจสอบว่า map ว่างหรือไม่
isNotEmpty	ตรวจสอบว่า map ไม่ว่างหรือไม่
length	return ความยาวของ map

```
void main() {
    Map<String, double> expenses = {
        'sun': 3000.0,
        'mon': 3000.0,
        'tue': 3234.0,
    };
    print("All keys of Map: ${expenses.keys}");
    print("All values of Map: ${expenses.values}");
    print("Is Map empty: ${expenses.isEmpty}");
    print("Is Map not empty: ${expenses.isNotEmpty}");
    print("Length of map is: ${expenses.length}");
}
```

Output

All keys of Map: (sun, mon, tue)
All values of Map: (3000, 3000, 3234)
Is Map empty: false
Is Map empty: true
Length of map is: 3

การเพิ่มองค์ประกอบลงใน Map

ตัวอย่างนี้จะเพิ่ม Key เป็น 'Japan' และ Value เป็น 'Tokio'

```
void main(){
    Map<String, String> countryCapital = {
        'USA': 'Washington, D.C.',
        'India': 'New Delhi',
        'China': 'Beijing'
    };

    // Adding New Item
    countryCapital['Japan'] = 'Tokio';
    print(countryCapital);
}
```

Output

{USA: Washington, D.C., India: New Delhi, China: Beijing,
Japan: Tokio}

การอัปเดตองค์ประกอบของ Map

ตัวอย่างนี้ เราต้องการอัปเดต ค่า Value ของ Key 'USA' จาก Value 'Nothing' เป็น 'Washington, D.C.'

```
void main(){
    Map<String, String> countryCapital = {
        'USA': 'Nothing',
        'India': 'New Delhi',
        'China': 'Beijing'
    };
    // Updating Item
    countryCapital['USA'] = 'Washington, D.C.';
    print(countryCapital);
}
```

Output

{USA: Washington, D.C., India: New Delhi, China: Beijing}

Map Methods ในภาษา Dart



คณะวิทยาศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

คุณสมบัติ	การทำงาน
<code>keys.toList()</code>	แปลง Key Maps ทั้งหมดเป็น List
<code>values.toList()</code>	แปลง Value Maps ทั้งหมดเป็น List
<code>containsKey('key')</code>	ตรวจสอบว่ามี Key นั้นหรือไม่
<code>containsValue('value')</code>	ตรวจสอบว่ามี Value นั้นหรือไม่
<code>remove()</code>	ลบองค์ประกอบทั้งหมดออกจาก Map
<code>removeWhere()</code>	ลบองค์ประกอบทั้งหมดออกจาก Map หากเงื่อนไขถูกต้อง

keys.toList() values.toList()

แสดง Keys & Values
ทั้งหมด เป็น List

```
void main() {  
    Map<String, double> expenses = {  
        'sun': 3000.0,  
        'mon': 3000.0,  
        'tue': 3234.0,  
    };  
  
    // Without List  
    print("All keys of Map: ${expenses.keys}");  
    print("All values of Map: ${expenses.values}");  
  
    // With List  
    print("All keys of Map with List: ${expenses.keys.toList()}");  
    print("All values of Map with List: ${expenses.values.toList()}");  
}
```

Output All keys of Map: (sun, mon, tue)
All values of Map: (3000, 3000, 3234)
All keys of Map with List: [sun, mon, tue]
All values of Map with List: [3000, 3000, 3234]

containsKey('key')
containsValue('value')

ตรวจสอบว่ามี

Key/Value

นั้นใน map หรือไม่

```
void main() {  
  
    Map<String, double> expenses = {  
        'sun': 3000.0,  
        'mon': 3000.0,  
        'tue': 3234.0,  
    };  
  
    // For Keys  
    print("Does Map contain key sun: ${expenses.containsKey("sun")}");  
    print("Does Map contain key abc: ${expenses.containsKey("abc")}");  
  
    // For Values  
    print("Does Map contain value 3000.0: ${expenses.containsValue(3000.0)}");  
    print("Does Map contain value 100.0: ${expenses.containsValue(100.0)}");  
  
}
```

Output Does Map contain key sun: true
Does Map contain key abc: false
Does Map contain value 3000.0: true
Does Map contain value 100.0: false

remove()

การลบ item ออก
จาก Map

```
void main(){
    Map<String, String> countryCapital = {
        'USA': 'Nothing',
        'India': 'New Delhi',
        'China': 'Beijing'
    };

    countryCapital.remove("USA");
    print(countryCapital);
}
```

Output {India: New Delhi, China: Beijing}

removeWhere()

ลบองค์ประกอบ
ทั้งหมดออกจาก Map
หากเงื่อนไขถูกต้อง

```
void main() {  
    Map<String, double> mathMarks = {  
        "ram": 30,  
        "mark": 32,  
        "harry": 88,  
        "raj": 69,  
        "john": 15,  
    };  
    mathMarks.removeWhere((key, value) => value < 32);  
    print(mathMarks);  
}
```

Output {mark: 32.0, harry: 88.0, raj: 69.0}



เปรียบเทียบการ Map ในภาษา Dart กับภาษาอื่น ๆ

- **Dart และ Python** ให้โค้ดที่กระชับและอ่านง่าย มีการรองรับ Map/Dictionary ในตัว
- **Java** มีความหลากหลายของคลาส Map เพื่อตอบสนองกับความต้องการในการจัดเก็บและจัดการข้อมูล.
- **C** ขาดการรองรับ Map ในตัว ส่งผลให้มีการจัดการอาเรย์แบบ manual และมีแนวโน้มที่จะเกิดข้อผิดพลาดมากขึ้น



Thank you