Dart Tutorial

Constructor in Dart (Part 2)

ตัวอย่างที่ 5 การเขียน Constructor แบบย่อ

- ตัวอย่างการเขียน Constructor ในรูปแบบเต็ม

```
class Person{
 String? name;
 int? age;
 String? subject;
 double? salary;

 // Constructor
Person(String name,int age,String subject,double salary){
  this.name = name;
  this.age = age;
  this.subject = subject;
  this.salary = salary;
 }

void display(){
  print("Name: ${this.name}");
  print("Age: ${this.age}");
  print("Subject: ${this.subject}");
  print("Salary: ${this.salary}");
  }
}

void main(){
  Person person = Person("John", 30, "Maths", 50000.0);
  person.display();
}
```

- สามารถเขียนในรูปแบบย่อดังนี้

```
class Person{
 String? name;
 int? age;
 String? subject;
 double? salary;

 // Constructor in shorthand
Person(this.name,this.age,this.subject,this.salary);

void display(){
  print("Name: ${this.name}");
  print("Age: ${this.age}");
  print("Subject: ${this.subject}");
  print("Salary: ${this.salary}");
  }
}

void main(){
  Person person = Person("John", 30, "Maths", 50000.0);
  person.display();
}
```

- หมายเหตุ ในภาษา java ไม่สามารถเขียน Constructor แบบย่อได้

ตัวอย่างที่ 6 Constructor แบบมี Optional Parameters

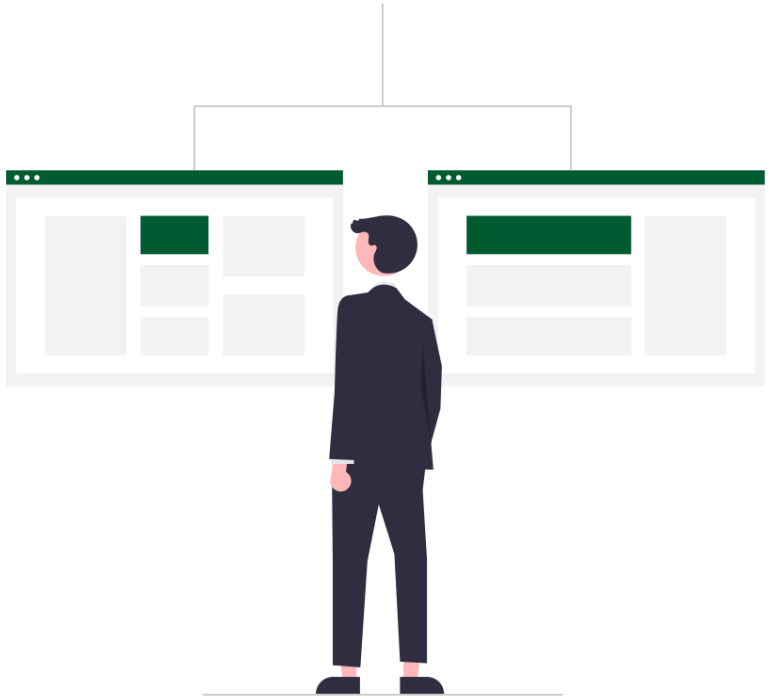
- Optional Parameters คือ Parameters ที่เราจะสามารถส่งหรือไม่ส่งค่ามาก็ได้ Constructor ก็ยังสามารถทำงานได้ตามปกติ ซึ่งปกติถ้าส่งค่าพารามิเตอร์มาไม่ครบตามที่ประกาศไว้ก็จะไม่สามารถสร้าง Object ได้

```
class Person {
 String? name;
 String? lastName;
 int? age;

 // Constructor
 Person([this.name, this.lastName, this.age]);

 // Method
 void display() {
   print("Name: ${this.name}");
   print("Last Name: ${this.lastName}");
   print("Age: ${this.age}");
 }
}

void main(){
 Person person = Person("David", "Carter");
 person.display();
}
```

```
Output

Name : David
LastName : Carter
Age : null
```

ตัวอย่าง Optional Parameters ในภาษา Java

- ในภาษา java จะไม่มี Optional Parameters เป็นการประยุกต์ใช้จากวิธี Constructor Overloading เพื่อรองรับการรับ Parameters ที่มีหลากหลายรูปแบบ



```java
class Person{
  String name;
  String lastName;
  int age;

  Person(String name,String lastName,int age){
    this.name = name;
    this.lastName = lastName;
    this.age = age;
  }
  Person(String name,String lastName){
    this.name = name;
    this.lastName = lastName;
    this.age = 0;
  }
  Person(String name){
    this.name = name;
    this.lastName = null;
    this.age = 0;
  }
  Person(){
    this.name = null;
    this.lastName = null;
    this.age = 0;
  }
```

ตัวอย่างที่ 7 Parameters ที่มีชื่อ (Named Parameters)

- Parameters ที่มีชื่อทำให้สามารถส่งค่าโดยการเรียกชื่อของ Parameters โดยไม่ต้องคำนึงถึงลำดับหรือจำนวนของ Parameters ที่ประกาศไว้ใน Constructor ก็ได้

```
class ShowMyDetails{
  String? name;
  String? lastName;
  int? age;

ShowMyDetails({this.name,this.lastName,this.age});

  void display(){
    print("Name: ${this.name}");
    print("Last Name: ${this.lastName}");
    print("Age: ${this.age}");
  }
}

void main(){
    ShowMyDetails smd = ShowMyDetails(name: "Jay", age: 24 ,lastName: "Tillu",);
    smd.display();
}
```

```
Output

Name : jay
LastName : Tillu
Age : 24
```

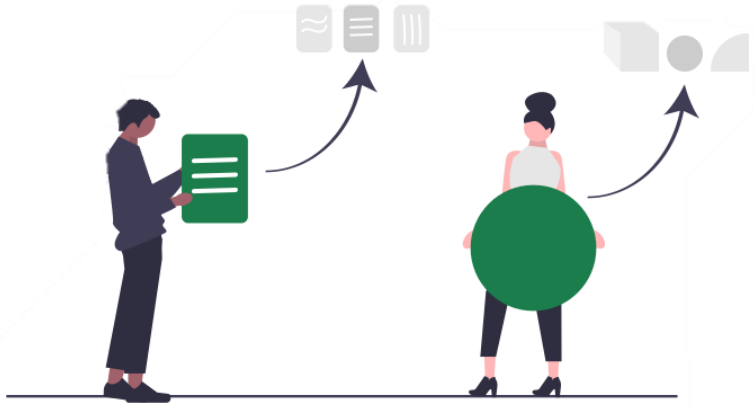
- หมายเหตุ ในภาษา java ไม่รองรับ Parameters ที่มีชื่อ

ตัวอย่างที่ 8 การกำหนด Default Values ใน Constructor

- เป็นการกำหนดค่าเริ่มต้นหากไม่มีการส่งค่าผ่านการเรียกใช้ Constructor ตอนสร้าง Object

```
class Square {
  int? height;
  int? width;

  // Assign Default Values in Constructor
  Square({this.height = 2, this.width = 2});

  // Method
  void display() {
    print("Square height: ${this.height}");
    print("Square width: ${this.width}");

  }
}

void main(){
  Square sq1 = Square();
  sq1.display();
}
```

```
Output

Square height: 2
Square width: 2
```



ตัวอย่างการกำหนด Default Values ใน Constructor ในภาษา Java

- เป็นการกำหนดค่าเริ่มต้นหากไม่มีการส่งค่าผ่านการเรียกใช้ Constructor ตอนสร้าง Object โดยการประยุกต์ใช้วิธี Constructor Overloading

```java
class Square{
    int height;
    int width;

    public Square(int height, int width){
        this.height = height;
        this.width = width;
    }
    public Square(int height){
        this.height = height;
        this.width = 2;
    }
    public Square(){
        this.height = 2;
        this.width = 2;
    }

    void display(){
        System.out.println("Square height : "+this.height);
        System.out.println("Square width : "+this.width);
    }

    Run | Debug
    public static void main(String arg[]){
        Square sqr = new Square();
        sqr.display();
    }

}
```

```
Output

Square height: 2
Square width: 2
```

Key Points

- Constructor ต้องมีชื่อเดียวกับคลาส
- Constructor ไม่ต้องมี return type
- Constructor ถูกเรียกใช้เพียงครั้งเดียว ณ เวลาสร้าง Object
- Constructor จะถูกเรียกใช้อย่างอัตโนมัติ ณ เวลาสร้าง Object
- Constructor ถูกใช้เพื่อ initialize ค่า Properties ของ Class

Reference(s)

•https://dart-tutorial.com

•https://dart.dev

•https://www.educative.io