

1.สร้างฟังก์ชัน Divide by zero ให้อยู่ในรูปแบบ Try, Option และ Either

```
@main def main(): Unit =
  TryDivideByZero(1, 0)
  TryDivideByZero(1, 1)
  println(OptionDivideByZero(1, 0))
  println(OptionDivideByZero(3, 3))
  println(EitherDivideByZero(1, 0))
  println(EitherDivideByZero(8, 4))

def TryDivideByZero(x: Int, y: Int): Unit =
  try
    println(x/y) //ลองหาร x ด้วย y
  catch
    case e: ArithmeticException => println("can not divide by zero") // หากเกิด
      ArithmeticException เช่น การหารด้วย 0 ขึ้น ให้แสดงข้อความ can not divide by
      zero

def OptionDivideByZero(x: Int, y: Int): Option[Int] =
  if y != 0 then
    Some(x/y) // หาก y ไม่เท่ากับ 0 ให้คืนค่า Some(x/y)
      ซึ่งหมายความว่าทำการหารได้สำเร็จ และมีค่าที่ใช้งานได้
  else
    None // หาก y เท่ากับ 0 ให้คืนค่า None หมายความว่า ไม่มีค่าที่ใช้งานได้

def EitherDivideByZero(x: Int, y: Int): Either[String, Int] =
  if y != 0 then
    Right(x/y) // หาก y ไม่เท่ากับ 0 หมายความว่า ทำการหารสำเร็จ และให้คืนค่า Right(x
      /y)
  else
    Left("can not divide by zero") // หาก y เท่ากับ 0 หมายความว่า ทำการหารไม่สำเร็จ
      และให้คืนค่า Left("can not divide by zero")
```

ผลลัพธ์

```
can not divide by zero
1
None
Some(1)
Left(can not divide by zero)
Right(2)
```

2.สร้างฟังก์ชัน StringToInt ให้อยู่ในรูปแบบ Try, Option และ Either

Try

```
def StringToInt(s: String): Int = //ฟังก์ชันแปลงจาก String เป็น Int
  try
    s.toInt //แปลง s เป็น Int
  catch //หากมีข้อผิดพลาดเกิดขึ้นใน try จะเข้าไปที่ catch
    case e: NumberFormatException => 0
    //ถ้าไม่ได้ใส่Stringที่เป็นตัวเลขมาจะคืนค่า 0
  finally //โค้ดใน finally จะทำงานเสมอไม่ว่าจะเจอข้อผิดพลาดหรือไม่
    println("success")
```

```
println(StringToInt("5"))
println(StringToInt("five"))
```

```
success
5
success
0
```

Either

```
def StringToInt3(s: String): Either[Int, Int] = //ฟังก์ชันแปลงจาก String เป็น Int
  try //จะคืนค่าในรูปแบบ Either คือ Left เป็น Int, Right เป็น Int
    Right(s.toInt) //หากแปลงสำเร็จ จะคืนค่า Right ซึ่งเป็นค่าที่แปลงสำเร็จ
  catch { //หากมีข้อผิดพลาดเกิดขึ้นใน try จะเข้าไปที่ catch
    case e: NumberFormatException => Left(0)
    //ถ้าไม่ได้ใส่Stringที่เป็นตัวเลขมาจะคืนค่า Left ซึ่งเป็น 0
  }
```

```
println(StringToInt3("5"))
println(StringToInt3("five"))
```

```
Right(5)
Left(0)
```

Option

```
// ประกาศตัวแปร Option โดยแยกเป็น 2 cases
enum Option[+A]:
  // case Some ในกรณีที่มีค่า
  case Some(get: A)

  // case None ในกรณีที่เป็นค่าว่าง
  case None

// import Option ทำให้ไม่ต้องพิมพ์ Option.Some และ Option.None
// ทุกครั้ง
import Option.{Some, None}

def StringToInt(word: String): Option[Int] = {
  try {
    // เมื่อมีค่า ให้แปลง String to Int
    Some(word.toInt)
  } catch {
    // ถ้าแปลงไม่สำเร็จ จะคืนค่า None
    case e: NumberFormatException => None
  }
}
```

```
@main def main(): Unit = {
  println(StringToInt("123"))
  println(StringToInt("hello"))
}
```

```
Some(123)
None
```

3. บอกความแตกต่างระหว่าง Try, Option และ Either

- Try ใช้จัดการข้อผิดพลาดที่เกิดขึ้นจากการ throw exceptions มี 2 กรณี ได้แก่
 - กรณีที่สำเร็จ
 - กรณีที่ไม่สำเร็จ สามารถเป็นได้แค่ประเภทที่ Throwable
- Option ใช้เพื่อบ่งชี้ว่ามีค่าหรือไม่
 - หากมีค่า จะเป็น Some(value)
 - หากไม่มีค่า จะเป็น Noneใช้ได้ดีกับกรณีที่ค่าอาจเป็น null
- Either มี 2 การใช้งาน ได้แก่
 - Right จะเป็นกรณีที่สำเร็จ
 - Left เป็นกรณีที่ไม่สำเร็จเหมาะสมกับผลลัพธ์ที่มี 2 แบบ