5. slice() & reverse()

**slice()**

slice() เป็นการหั่นส่วนของ array ตาม index ที่เราต้องการ และ return ออกมาเป็น array ใหม่

slice syntax

1. array.slice() แบบไม่รับ parameter สร้างและ return เป็น array ใหม่ที่มีข้อมูลเหมือนใน array ตัวเดิม

**console.log('////////////////SLICE////////////////');** **console.log('////////////////Ex1.1////////////////');**

**let person = ["tus", "gus", "phai", "dose" , "fern"];**  // ตัวแปร array ที่มีเก็บค่า String **console.log(person.slice()); console.log('////////////////Ex1.2////////////////');**

**let person1 = [{name:'tuskung' , age:19} , ["tus", "gus", "phai", "dose" , "fern"] , 'Phai' , 5];** // ตัวแปร array ที่มีเก็บค่า Object , Array , String และ Number **console.log(person1.slice()); console.log('////////////////Ex1.3////////////////');**

**let person2 = [(() => 'tuskung') , (function phai(name){return name;})]**

// ตัวแปร array ที่มีเก็บค่า Function

**console.log(person2.slice());**

ผลลัพธ์

Text

Description automatically generated

2. array.slice(start) array ใหม่ที่สร้างขึ้นมาจาก การหั่น array ตั้งแต่ index ที่ start เป็นต้นไป

**console.log('////////////////Ex2.1////////////////');**   //ใช้ person จากข้อ 1.1 **console.log(person.slice(2));** //เลือกเอาข้อมูลตั้งแต่ index ตัวที่ 2 ขึ้นไป **console.log('////////////////Ex2.2////////////////');**   //ใช้ person1 จากข้อ 1.2 **console.log(person1.slice(-2));** //เลือกเอาข้อมูลตั้งแต่ index ตัวที่ length-2 ขึ้นไป **console.log('////////////////Ex2.3////////////////');**   //ใช้ person2 จากข้อ 1.3 **console.log(person2.slice(person2.length));**

//เลือกเอาข้อมูลตั้งแต่ index ตัวที่ person2.length ขึ้นไปจะได้ array เปล่าๆ

ผลลัพธ์

Text

Description automatically generated

3. array.slice(start , end)

**console.log('////////////////Ex3.1////////////////');**   //ใช้ person จากข้อ 1.1 **console.log(person.slice(1,2));**

//เลือกเอาข้อมูลตั้งแต่ index ตัวที่ 1 ขึ้นไปจนถึง index ตัวที่ 2 แต่ไม่เอา index ตัวที่ 2

**console.log('////////////////Ex3.2////////////////');**   //ใช้ person1 จากข้อ 1.2 **console.log(person1.slice(0,-2));**

//เลือกเอาข้อมูลตั้งแต่ index ตัวที่ 0 ขึ้นไปแล้ว ไม่เอา index 2 ตำแหน่งท้าย

**console.log('////////////////Ex3.3////////////////');**   //ใช้ person2 จากข้อ 1.3 **console.log(person2.slice(2,person2.length));**

//เลือกเอาข้อมูลตั้งแต่ index ตัวที่ 2 ถึง person2.length

//แต่การทำแบบนี้ก็จะให้ค่าที่ไม่ต่างกับการใช้ person2.slice(2)

ผลลัพธ์

Text

Description automatically generated

**reverse()**

reverse() syntax

arr.reverse() reverse ข้อมูลจากตัวสุดท้ายไล่มาจนถึงตัวแรก

**console.log('////////////////Reverse////////////////');** **console.log('////////////////Ex1////////////////');** //ใช้ person จากตัวอย่าง slice ข้อ 1.1 **console.log(person);**    //ก่อน reverse   [ 'tus', 'gus', 'phai', 'dose', 'fern' ] **person.reverse(); console.log(person);**

//หลัง reverse   [ 'fern', 'dose', 'phai', 'gus', 'tus' ]

**console.log('////////////////Ex2////////////////');** //ใช้ person1 จากตัวอย่าง slice ข้อ 1.2 **console.log(person1);**

//ก่อน reverse   [{ name: 'tuskung', age: 19 },[ 'tus', 'gus', 'phai', 'dose', 'fern' ],'Phai',5]

**person1.reverse();**

**console.log(person1);**

//หลัง reverse   [5,'Phai',[ 'tus', 'gus', 'phai', 'dose', 'fern' ],{ name: 'tuskung', age: 19 }]

**console.log('////////////////Ex3////////////////');** //ใช้ person2 จากตัวอย่าง slice ข้อ 1.3 **console.log(person2);**   //ก่อน reverse   [ [Function (anonymous)], [Function: phai] ] **person2.reverse(); console.log(person2);**

//หลัง reverse   [ [Function: phai], [Function (anonymous)] ]

ผลลัพธ์

Text

Description automatically generated