

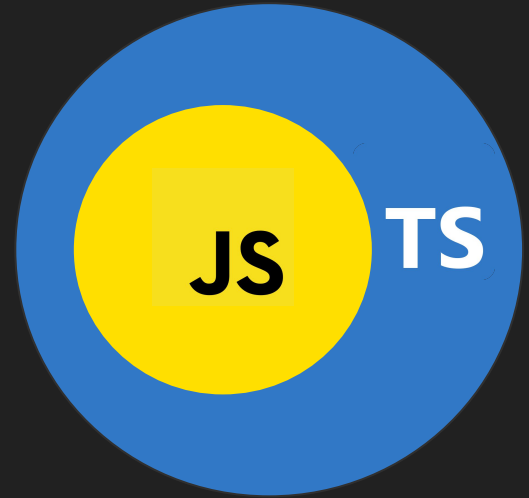


รู้จักกับ TypeScript



TypeScript คืออะไร

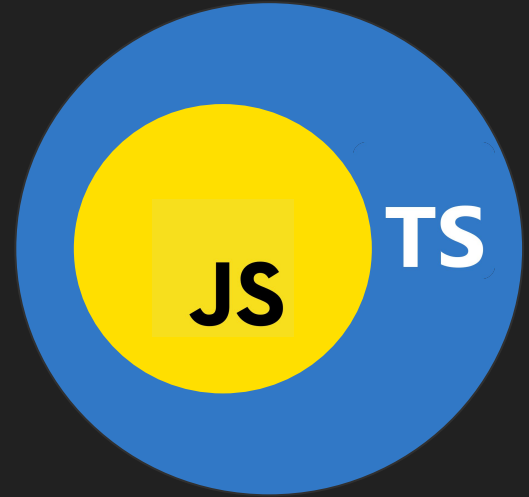
TypeScript เป็นภาษาที่ถูกออกแบบมาเพื่อช่วยให้การเขียนโปรแกรมด้วย JavaScript นั้นมีประสิทธิภาพมากยิ่งขึ้น โดยโค้ดที่ถูกเขียนด้วย TypeScript นั้นจะถูกแปลงให้เป็น JavaScript ในเวอร์ชันที่เราต้องการได้





TypeScript คืออะไร

เพราะฉะนั้นหากมีการเปลี่ยนแปลง
เวอร์ชันของ JavaScript เราก็ไม่จำเป็นต้องไป
ศึกษาใหม่ สามารถใช้ TypeScript แปลงให้เป็น
JavaScript เวอร์ชันนั้นๆ ได้ทันที



ทำไมต้องใช้ TypeScript



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



ทำไมต้องใช้ TypeScript

1. Superset ของ JavaScript
2. Compile ก่อนใช้งาน
3. Static Type



ทำไมต้องใช้ TypeScript

- **Superset** ของ JavaScript หมายถึงสามารถใช้ทุกฟีเจอร์ใหม่ๆของ JavaScript ได้ (JavaScript ทำอะไรได้ TypeScript ก็ทำได้เหมือนกัน)
- **Compile ก่อนใช้งาน** ถ้าใช้งาน JavaScript จะไม่รู้เลยว่าโค้ดทำงานถูกต้องหรือไม่ จนกว่าจะนำโปรแกรมไปรันบน Browser แต่ TypeScript จะคอมไพล์เพื่อตรวจสอบข้อผิดพลาดที่อาจจะเกิดขึ้นก่อนที่จะรันโปรแกรม ทำให้ทราบได้ว่าโปรแกรมของเรานั้นทำงานผิดพลาดตรงจุดใด



ทำไมต้องใช้ TypeScript

- **Static Type** หมายถึง เป็นภาษาที่มีชนิดข้อมูลสืบเนื่องมาจาก JavaScript นั้นเป็นภาษาในรูปแบบ Dynamic Type คือ ตัวแปรที่เก็บข้อมูลนั้นสามารถเปลี่ยนแปลงชนิดข้อมูลได้ตลอด (ทำให้เกิดบัคได้ง่าย) จึงได้มีการพัฒนา TypeScript ขึ้นมาเพื่อกำหนดชนิดหรือประเภทข้อมูลลงในตัวแปรได้





ทำไมต้องใช้ TypeScript

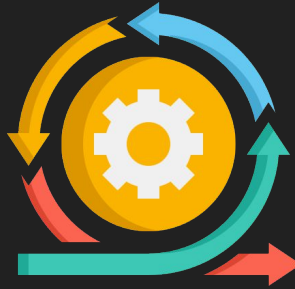
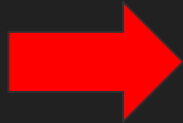
ในการใช้งาน TypeScript นั้น เราต้องการผลลัพธ์สุดท้ายเป็น JavaScript เนื่องจาก Browser เข้าใจแค่ JavaScript เฉพาะฉะนั้นเราต้องเขียนโปรแกรมด้วย TypeScript ก่อนและคอมไพล์เพื่อแปลงโค้ดดังกล่าวให้เป็น JavaScript จากนั้นนำผลลัพธ์ที่ได้จากการแปลงไปทำงานบน Browser



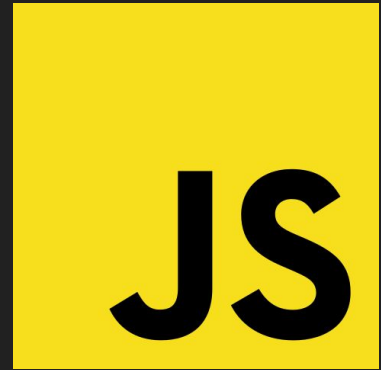
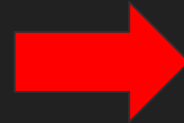
ทำไมต้องใช้ TypeScript



TypeScript



Compile



JavaScript



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ต้องมีพื้นฐานอะไรบ้าง ?



<https://www.youtube.com/c/KongRuksiamOfficial/>

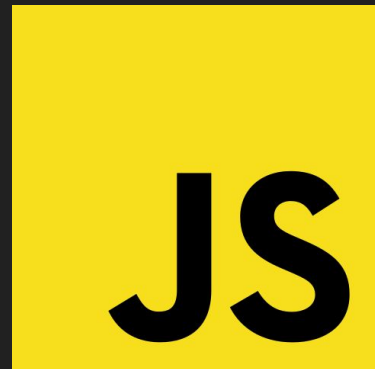


<https://www.facebook.com/KongRuksiamTutorial/>



ต้องมีพื้นฐานอะไรบ้าง

- JavaScript เบื้องต้น
- JavaScript ES6



เครื่องมือพื้นฐาน



- Visual Studio Code
- Node.js
- TypeScript *



ติดตั้ง Node.js



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Nodejs คืออะไร

Node.js เป็นชุดเครื่องมือในการแปลคำสั่งของ JavaScript และ เป็น JavaScript Runtime Environment กล่าวคือ สามารถ นำ JavaScript ไปรันใน Windows , Mac , Linux ได้ ส่งผลให้ สามารถรันโค้ด JavaScript ด้วย Nodejs ได้เลย โดยไม่จำเป็นต้องสร้างเป็นเว็บเพจแล้วนำเว็บเพจไปรันใน Web Browser นั้นเอง (ไม่จ้อ Web Browser)



ติดตั้ง Visual Studio Code



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ปัญหาของ JavaScript



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ติดตั้ง TypeScript



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



ติดตั้ง TypeScript

```
npm install -g typescript
```

ตรวจสอบเวอร์ชัน

```
tsc -v
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

เริ่มต้นใช้งาน TypeScript



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



สร้างไฟล์ TypeScript

ชื่อไฟล์.ts

แปลง TypeScript เป็น JavaScript

tsc ชื่อไฟล์.ts

เช่น tsc app.ts (TypeScript Compiler)

สร้างโปรเจกต์ TypeScript



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



สร้างโปรเจกต์ TypeScript

คำสั่ง

```
tsc --init
```

****tsconfig.json** คือไฟล์สำหรับตั้งค่าการทำงานของ
TypeScript Compiler



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

รู้จักกับไฟล์ tsconfig.json



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

tsconfig.json คืออะไร



คือ ไฟล์สำหรับตั้งค่าการทำงานของ TypeScript Compiler
ประกอบด้วยค่าพื้นฐานที่ควรทราบดังนี้

- target
- rootDir
- outDir
- include

tsconfig.json คืออะไร

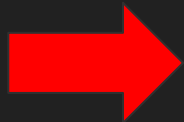
TS

1. **"target": "es2016"** เป็นการกำหนดให้คอมไพล์ TypeScript ออกมาเป็น JavaScript es2016
2. **"rootDir": "./"** ตำแหน่งโฟลเดอร์ที่เก็บกลุ่มไฟล์ TypeScript (.ts)
3. **"outDir": "./dist/"** ตำแหน่งโฟลเดอร์ที่เก็บกลุ่มไฟล์ JavaScript (.js) ที่คอมไพล์เรียบร้อยแล้ว เช่น โฟลเดอร์ที่ชื่อว่า dist เป็นต้น
4. **"include": ["src"]** กำหนดขอบเขตของไฟล์ที่จะให้คอมไพล์ เช่น ไฟล์ที่ถูกจัดเก็บในโฟลเดอร์ src เป็นต้น

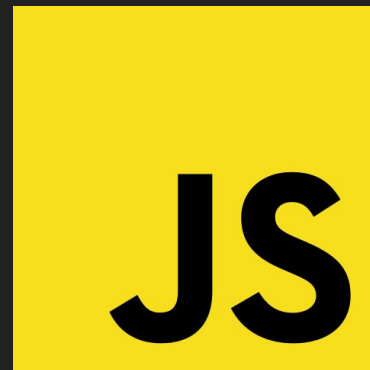
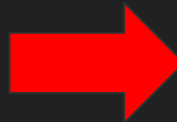
tsconfig.json คืออะไร



src/app.ts



compile...



dist/app.js

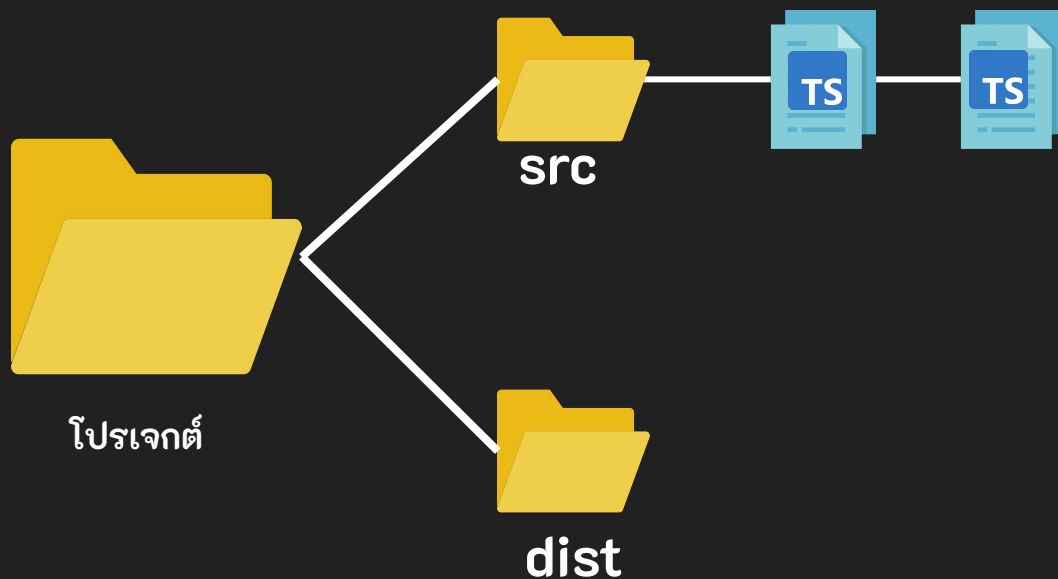


<https://www.youtube.com/c/KongRuksiamOfficial/>

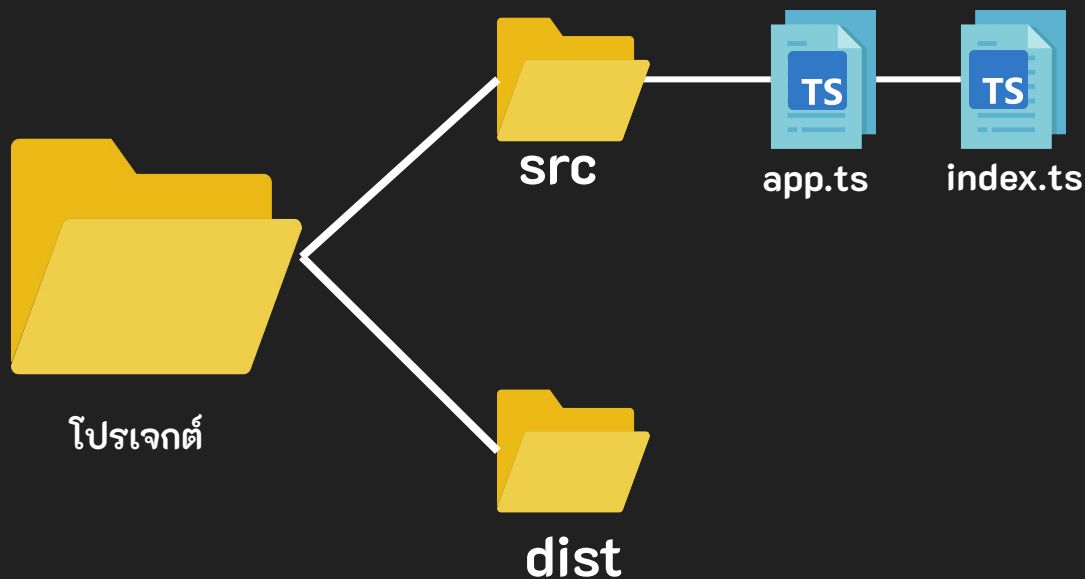


<https://www.facebook.com/KongRuksiamTutorial/>

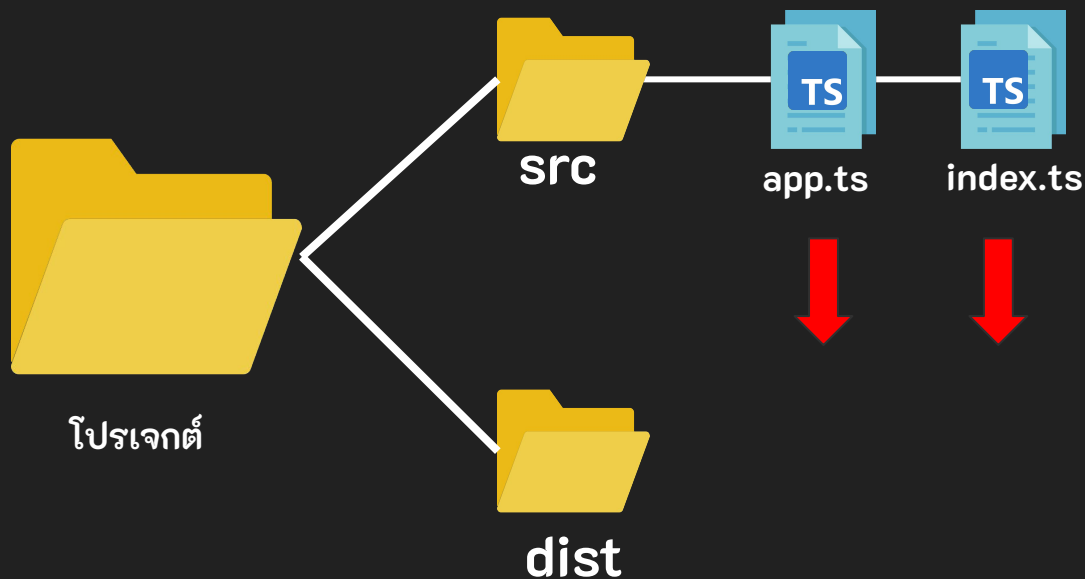
tsconfig.json คืออะไร



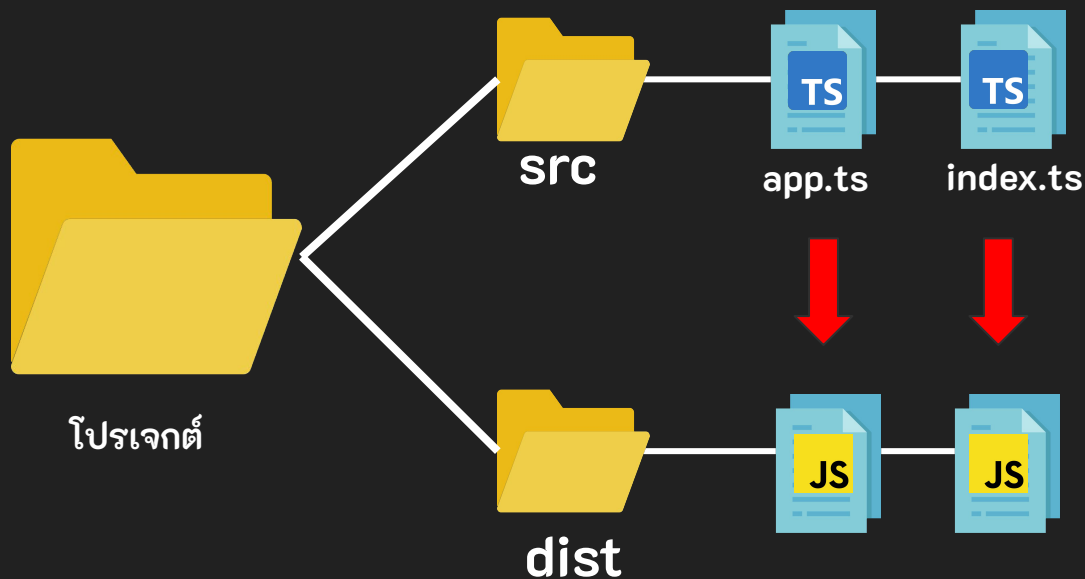
tsconfig.json คืออะไร



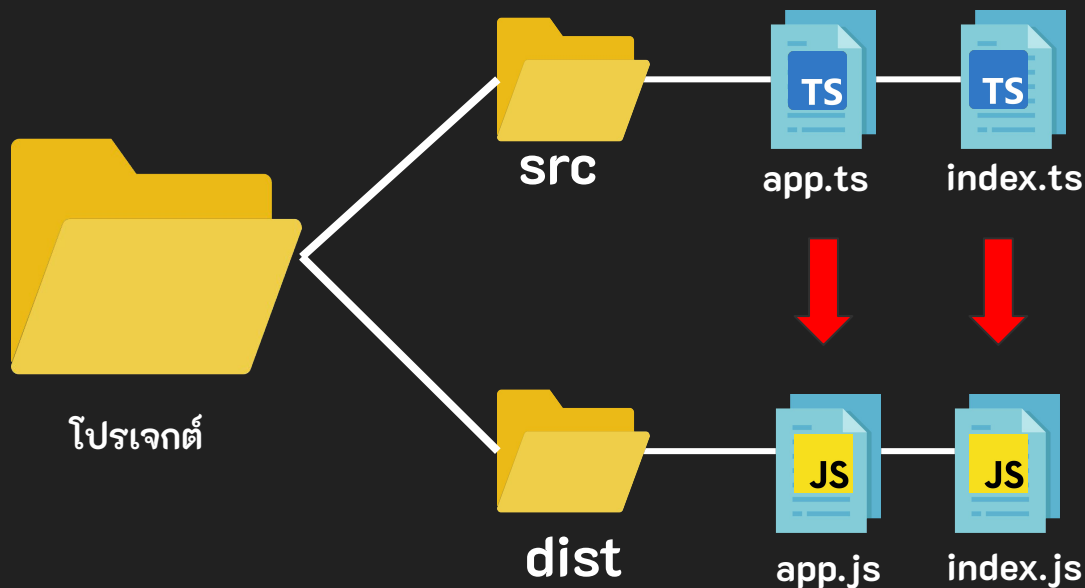
tsconfig.json คืออะไร



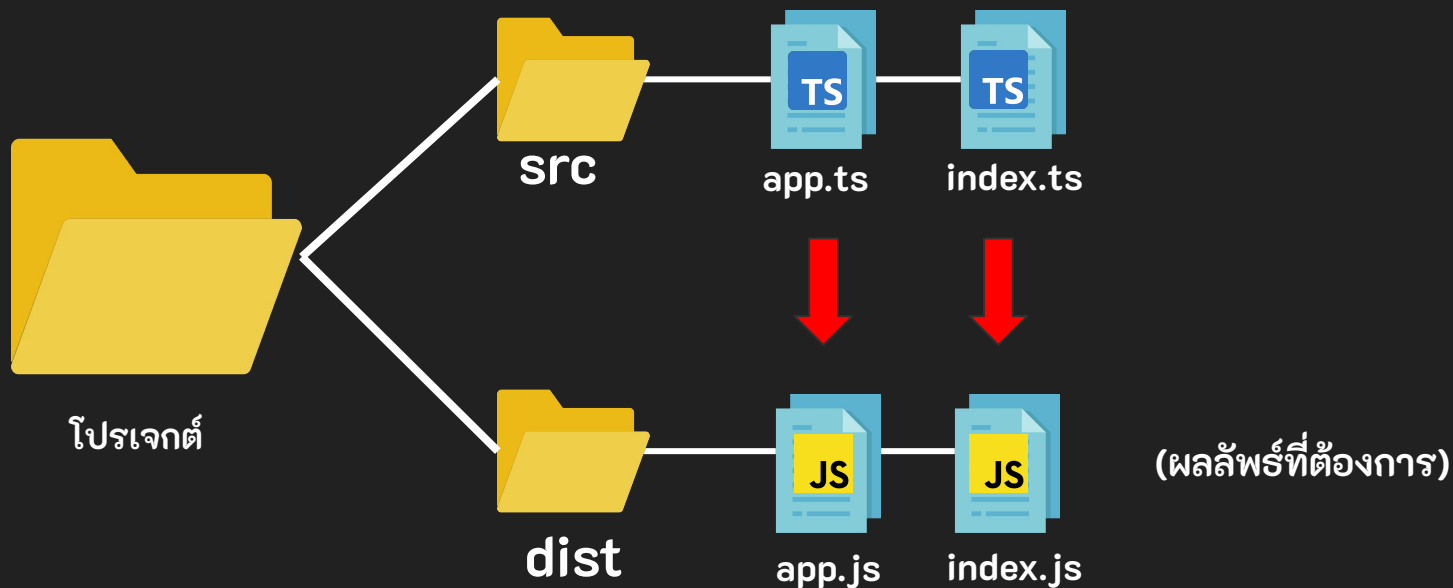
tsconfig.json คืออะไร



tsconfig.json คืออะไร



tsconfig.json คืออะไร



Watch Mode



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Watch Mode คืออะไร



เป็นการสั่งคอมไพล์พร้อมกับตรวจสอบไฟล์ ทุกๆไฟล์ที่อยู่ในโปรเจกต์ถ้ามีการเปลี่ยนแปลงคำสั่งจะให้คอมไพล์ใหม่ทันที โดยไม่ต้องสั่งคอมไพล์หลายๆครั้ง (คอมไพล์แบบเรียลไทม์)

คำสั่ง

`tsc -w` หรือ `tsc --watch`

ชนิดข้อมูลพื้นฐาน



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ชนิดข้อมูลพื้นฐาน



Data Type	คำอธิบาย	ตัวอย่าง
boolean	ค่าทางตรรกศาสตร์	True / False
number	ตัวเลขที่ไม่มีจุดทศนิยม	20
	ตัวเลขที่มีจุดทศนิยม	30.15
string	ข้อความ	"kongruksiam"



ตัวแปร (Variable)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

การนิยามตัวแปร



`const` , `let` ชื่อตัวแปร : ชนิดข้อมูล = ค่าเริ่มต้น

ตัวอย่าง

```
let fname : string = "kongruksiam"
```

```
let age:number = 30;
```

```
let isWorking : boolean = false;
```

Any & Unknown



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Any คืออะไร

any คือ การนิยามตัวแปรที่มีชนิดข้อมูลแบบใดก็ได้
กล่าวคือ ตัวแปรใดๆที่ TypeScript ไม่สามารถบอกได้
ว่าเป็นข้อมูลแบบใด จะทำการกำหนดให้ตัวแปรนั้นมี
ชนิดข้อมูลเป็น any

ตัวอย่างการใช้ Any

```
let myVar:any = "kongruksiam"; // เป็น string
```

```
myVar = 10; // เป็น number ก็ได้
```

```
myVar = true; // เป็น boolean ก็ได้
```

ประโยชน์ของ Any

“ การกำหนด any นั้นจะสามารถทำได้
เมื่อต้องการอยากให้ข้อมูลมีการ
เปลี่ยนแปลงค่าได้ ”

ข้อควรระวัง

- ควรระบุชนิดข้อมูลให้เรียบร้อย ไม่ควรปล่อยให้ชนิดข้อมูลเป็น any เพราะอาจทำให้ข้อผิดพลาดได้ เนื่องจาก any นั้นสามารถทำให้ค่าข้อมูลเปลี่ยนเนื่องจากการไม่มีการตรวจสอบชนิดข้อมูล ถ้าเสี่ยงไม่ได้ให้เปลี่ยนจากการใช้ any ไปเป็น **unknown** แทน



Unknown คืออะไร

มีลักษณะคล้าย Any คือ การนิยามตัวแปรที่มีชนิดข้อมูลแบบใดก็ได้ แต่จะมีการตรวจสอบชนิดข้อมูลก่อนเริ่มทำงาน

ตัวอย่างการใช้ Unknown

```
let myVar:unknown = "kongruksiam"; // เป็น string  
if(typeof myVar=== "ชนิดข้อมูล"){  
    //กระบวนการทำงาน  
}
```



สรุป Any & Unknown

ตัวแปรของเราสามารถเปลี่ยนชนิดข้อมูลได้ โดย

- any ไม่มีการตรวจสอบชนิดข้อมูล
- unknown มีการตรวจสอบชนิดข้อมูล



Type Assertion



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Type Assertion คืออะไร

กระบวนการระบุชนิดข้อมูลด้วยตนเอง ซึ่งสามารถ
ระบุได้ 2 รูปแบบ ดังนี้

1. as
2. <ชนิดข้อมูล>

โครงสร้างควบคุมพื้นฐาน



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

แบบมีเงื่อนไข (Condition)

กลุ่มคำสั่งที่ใช้ตัดสินใจในการเลือกทำงานตามเงื่อนไขต่างๆ
ภายในโปรแกรม

- if..else
- Switch..Case

if...else

```
if(เงื่อนไข){  
    คำสั่งเมื่อเงื่อนไขเป็นจริง ;  
}  
else{  
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ ;  
}
```

Ternary Operator (If..else ลดรูป)

```
if(เงื่อนไข){  
    //คำสั่งเมื่อเงื่อนไขเป็นจริง  
}else{  
    //คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
}
```

ตัวแปร = (เงื่อนไข) ? คำสั่งเมื่อเงื่อนไขเป็นจริง : คำสั่งเมื่อเงื่อนไขเป็นเท็จ;

Switch..Case



```
switch (ค่าที่เก็บในตัวแปรควบคุม) {  
    case ค่าที่ 1 : คำสั่งที่ 1;  
        break;  
    case ค่าที่ N : คำสั่งที่ N;  
        break;  
    default : คำสั่งเมื่อไม่มีค่าที่ตรงกับที่ระบุใน case  
        break;  
}
```



โจทย์ปัญหา : เลือกใช้บริการระบบธนาคาร

ให้ป้อนหมายเลขเพื่อใช้บริการ

- เลข 1 คือ สอบถามยอดคงเหลือในบัญชี
- เลข 2 คือ ฝากเงิน
- เลข 3 คือ ถอนเงิน
- ตัวเลขอื่นๆ คือ หมายเลขไม่ถูกต้อง



แบบทำซ้ำ (Loop)

กลุ่มคำสั่งที่ใช้ในการวนรอบ (loop)

โปรแกรมจะทำงานไปเรื่อยๆจนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จจึงจะหยุดทำงาน

ฟังก์ชัน (Function)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ฟังก์ชัน (Function)

ชุดคำสั่งที่นำมาเขียนรวมกันเป็นกลุ่มเพื่อให้เรียกใช้งานตามวัตถุประสงค์ที่ต้องการและลดความซ้ำซ้อนของคำสั่งที่ใช้งานบ่อยๆ ฟังก์ชันสามารถนำไปใช้งานได้ทุกที่และแก้ไขได้ในภายหลัง ทำให้โค้ดในโปรแกรมมีระเบียบและใช้งานได้สะดวกมากยิ่งขึ้น



ฟังก์ชัน (Function)

- ฟังก์ชันที่ไม่มีการรับและส่งค่า (void)
- ฟังก์ชันที่มีการรับค่า (Parameter)
- ฟังก์ชันที่มีส่งค่า (Return)
- ฟังก์ชันที่มีการรับและส่งค่า (Parameter & Return)



ฟังก์ชันที่ไม่มีการรับและส่งค่า (void)

```
function ชื่อฟังก์ชัน() {  
    // คำสั่งต่างๆ  
}
```

การเรียกใช้งานฟังก์ชัน

ชื่อฟังก์ชัน ();

```
function ชื่อฟังก์ชัน() void {  
    // คำสั่งต่างๆ  
}
```

การเรียกใช้งานฟังก์ชัน

ชื่อฟังก์ชัน ();



ฟังก์ชันที่มีการรับค่า (Parameter)

```
function ชื่อฟังก์ชัน(พารามิเตอร์ : ชนิดข้อมูล,...){
```

```
// กลุ่มคำสั่งต่างๆ
```

```
}
```

- อาร์กิวเมนต์ คือ ตัวแปรหรือค่าที่ต้องการส่งมาให้กับฟังก์ชัน
- พารามิเตอร์ คือ ตัวแปรที่ฟังก์ชันสร้างไว้สำหรับรับค่าที่จะส่งเข้ามาให้กับฟังก์ชัน

การเรียกใช้งานฟังก์ชัน

ชื่อฟังก์ชัน (argument1,argument2,.....)

ฟังก์ชันที่มีส่งค่า (Return)

```
function ชื่อฟังก์ชัน() ชนิดข้อมูล{  
    return value; // อ้างอิงตามชนิดข้อมูลที่ส่งออกไป  
}
```

ฟังก์ชันที่มีการรับและส่งค่า

```
function ชื่อฟังก์ชัน(พารามิเตอร์ : ชนิดข้อมูล,...) ชนิดข้อมูล{  
    return value; // อ้างอิงตามชนิดข้อมูลที่ส่งออกไป  
}
```

Arrow Function

เป็นรูปแบบการเขียน Function ให้มีความกระชับมากยิ่งขึ้น

โครงสร้างคำสั่ง

```
const ชื่อฟังก์ชัน=(พารามิเตอร์ : ชนิดข้อมูล,..) : ชนิดข้อมูล =>{  
    return value; // อ้างอิงตามชนิดข้อมูลที่ส่งออกไป  
}
```

Default Parameter

เป็นการสร้างฟังก์ชันโดยกำหนดค่าเริ่มต้นให้กับพารามิเตอร์

โครงสร้างคำสั่ง

```
function ชื่อฟังก์ชัน(พารามิเตอร์ : ชนิดข้อมูล = ค่าเริ่มต้น){  
    // กลุ่มคำสั่งต่างๆ  
}
```


Object



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Object คืออะไร

ตัวแปรที่สามารถเก็บค่าได้หลายค่า โดยเขียนอยู่ในรูปของคู่
property : value (ใช้เครื่องหมาย colon ในการจำแนก)

- **property** คือ ชื่อในการอ้างอิงข้อมูล
- **value** คือ ข้อมูลที่จัดเก็บใน property



การสร้าง Object แบบกำหนดค่า

โครงสร้างคำสั่ง

```
const ชื่อวัตถุ = {  
    property:value  
}
```

ตัวอย่าง

```
const person = {  
    name:"kongruksiam",  
    age:20  
}
```



การสร้าง Object แบบระบุชนิดข้อมูล

โครงสร้างคำสั่ง

```
const ชื่อวัตถุ : { property:type } = {ค่าเริ่มต้น}
```

ตัวอย่าง

```
const position : {x:number,y:number} = {x:10,y:20}
```



ฟังก์ชันรับค่า Object

โครงสร้างคำสั่ง

```
const ชื่อฟังก์ชัน = (object) : void=>{  
  
    //คำสั่งต่างๆ  
}
```



ฟังก์ชันส่งค่า Object

โครงสร้างคำสั่ง

```
const ชื่อฟังก์ชัน = () : object =>{  
  
    return object  
}
```



Excess Properties Check

การตรวจสอบ Properties ที่ส่งเข้ามาทำงานในฟังก์ชัน

ตัวอย่าง

ชื่อฟังก์ชัน (parameter object)

เปลี่ยนเป็น

ชื่อฟังก์ชัน ({parameter object})

Type Aliases



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Type Aliases คืออะไร

นอกจากชนิดข้อมูลพื้นฐานที่อยู่ในภาษา TypeScript
เราสามารถนิยามหรือสร้างชนิดข้อมูลในรูปแบบของเราได้
เรียกว่า **Type Aliases (Custom Type)**

Type Aliases คืออะไร

โครงสร้างคำสั่ง

```
type ชื่อTypeAliases = {  
    property:type,  
    ....  
}
```

ตัวอย่าง

```
type Employee = {  
    name:string,  
    salary:number  
}
```

การใช้งาน Type Aliases

โครงสร้างคำสั่ง

```
let ชื่อตัวแปร : ชื่อTypeAliases = {property:value,property:value}
```

ตัวอย่าง

```
let emp1 : Employee = {name:"KongRuksiam",salary:50000}
```

การเข้าถึงข้อมูลใน Type Aliases

ตัวอย่าง

```
let emp1 : Employee = {name:"KongRuksiam",salary:50000}
```

```
emp1.name = "ก้องรักสยาม"
```

```
emp1.salary = 100000
```

```
console.log(emp1)
```



ตัวอย่าง Type Aliases (ข้อมูลพนักงาน)

สร้าง Type Aliases เพื่อจัดการเกี่ยวกับข้อมูลพนักงาน
โดยมีโครงสร้างดังนี้

- ID - รหัสพนักงาน (number)
- Name - ชื่อพนักงาน (string)
- Salary - เงินเดือน (number)
- Phone - เบอร์ติดต่อ (string)



Type Aliases (Optional Properties)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Type Aliases (Optional Properties)

เป็นการระบุว่า Property ที่ทำงานอยู่ใน Type Aliases นั้นสามารถกำหนดค่าได้หรือไม่กำหนดค่าก็ได้ โดยใช้เครื่องหมาย ?

โครงสร้างคำสั่ง

```
type ชื่อTypeAliases = {  
    property ? : type,  
}
```

Type Aliases (Readonly Modifier)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Type Aliases (Readonly Modifier)

เป็นการระบุว่า Property ที่ทำงาน
อยู่ใน Type Aliases นั้นหลังจากที่
กำหนดค่าแล้วจะไม่สามารถแก้ไข
ค่าได้ (อ่านได้อย่างเดียว) โดยใช้
readonly modifier

โครงสร้างคำสั่ง

```
type ชื่อTypeAliases = {  
    readonly property : type,  
}
```

อาร์เรย์ (Array)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

อาร์เรย์ คืออะไร

1. ชุดของตัวแปรที่อยู่ในรูปลำดับใช้เก็บค่าข้อมูลให้อยู่ในกลุ่มเดียวกัน โดยข้อมูลภายในอาร์เรย์จะถูกเก็บในตำแหน่งที่ต่อเนื่องกัน
2. เป็นตัวแปรที่ใช้ในการเก็บข้อมูลที่มีลำดับที่ต่อเนื่อง ซึ่งข้อมูลมีค่าได้หลายค่าโดยใช้ชื่ออ้างอิงได้เพียงชื่อเดียว และใช้หมายเลขกำกับ (**index**) ให้กับตัวแปรเพื่อจำแนกความแตกต่างของค่าตัวแปรแต่ละตัว

คุณสมบัติของอาร์เรย์

1. ใช้เก็บกลุ่มของข้อมูล
2. ข้อมูลที่อยู่ในอาร์เรย์จะเรียกว่าสมาชิก หรือ อิลิเมนต์ (element)
3. แต่ละอิลิเมนต์ (element) จะเก็บค่าข้อมูล (value) และ อินเด็กซ์ (Index)
4. Index หมายถึงคีย์ของอาร์เรย์ใช้อ้างอิงตำแหน่งของ element เริ่มต้นที่ 0
5. สมาชิกในอาร์เรย์ต้องมีชนิดข้อมูลเหมือนกัน
6. สมาชิกในอาร์เรย์จะถูกค้นด้วยเครื่องหมายคอมม่า



การสร้างอาร์เรย์

แบบไม่กำหนดค่าเริ่มต้น

```
const ชื่อตัวแปร : ชนิดข้อมูล []=[];
```

แบบกำหนดค่าเริ่มต้น

```
const ชื่อตัวแปร : ชนิดข้อมูล [] = [element,...]
```



การเข้าถึงสมาชิก ในอาร์เรย์ (Array)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

รูปแบบการเข้าถึงสมาชิก

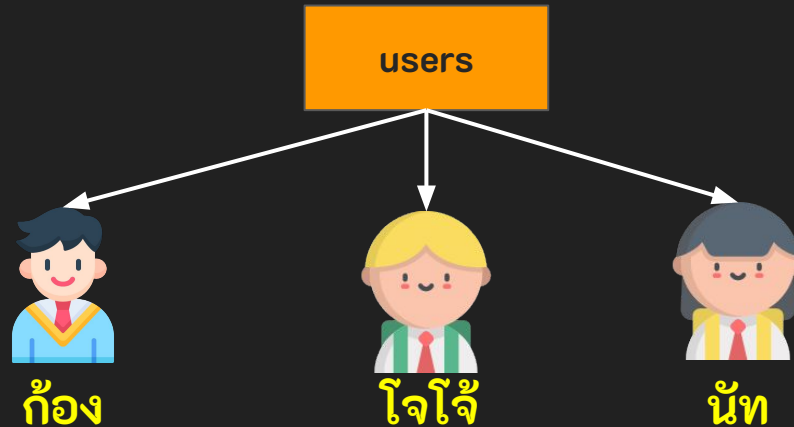
1. อ้างอิงผ่าน Index
2. ใช้ For Loop ร่วมกับ length
3. ใช้ ForEach



เข้าถึงสมาชิกในอาร์เรย์

TS

```
const users:string []=["ก้อง","โจโจ้","นัท"]
```



<https://www.youtube.com/c/KongRuksiamOfficial/>

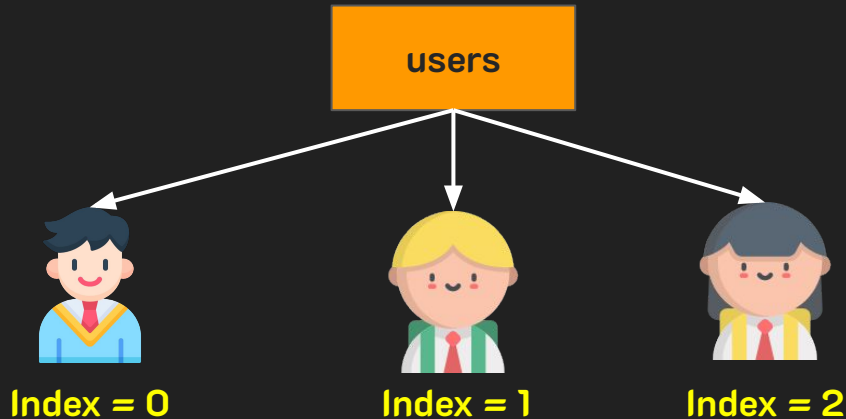


<https://www.facebook.com/KongRuksiamTutorial/>

เข้าถึงสมาชิกในอาร์เรย์

TS

```
const users:string []=["ก้อง","โจโจ้","นัท"]
```



จำนวนสมาชิกในอาร์เรย์ (Length)

```
const users:string []=["ก้อง","โจโจ้","นัท"]
```

ก้อง	โจโจ้	นัท
------	-------	-----



Length = 3

เข้าถึงสมาชิกในอาร์เรย์ (For Loop)

```
const users:string []=["ก้อง","โจโจ้","นัท"]  
  
for (let i = 0; i < users.length ; i++) {  
    console.log(users[i]);  
}
```



เข้าถึงสมาชิกในอาร์เรย์ (ForEach)

```
const users:string []=["ก้อง","โจโจ้","นัท"]  
users.forEach(element=>{  
    console.log(element)  
})
```



Array & Type Aliases

Function Overloading

Function Overloading คืออะไร

การสร้างฟังก์ชันที่มีชื่อเหมือนกันแต่สามารถรับ
พารามิเตอร์จำนวนต่างกันพร้อมคืนค่าที่แตกต่างกันได้

ตัวอย่างที่ 1

1. `function sayHi() :string`
2. `function sayHi(name:string) :string`
3. `function sayHi(name?:unknown) :unknown`



ตัวอย่างที่ 1

1. `function sayHi() :string`
2. `function sayHi(name:string) :string`
3. `function sayHi(name?:unknown) :unknown`



ตัวอย่างที่ 1

1. `function sayHi() :string`
2. `function sayHi(name:string) :string`
3. `function sayHi(name?:unknown) :unknown`



ตัวอย่างที่ 2

1. `function total (a:number,b:number) :number`
2. `function total (a:string,b:string) :string`
3. `function total (a:unknown,b:unknown)
:unknown`



ตัวอย่างที่ 2

1. function total (**a:number,b:number**) :number
2. function total (**a:string,b:string**) :string
3. function total (**a:unknown,b:unknown**)
:unknown



ตัวอย่างที่ 2

1. `function total (a:number,b:number) :number`
2. `function total (a:string,b:string) :string`
3. `function total (a:unknown,b:unknown) :unknown`

Spread Operator



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Spread Operator

เป็นรูปแบบการกระจายสมาชิกในอาร์เรย์ออกมาใช้งาน
โดยเติมเครื่องหมาย ... ด้านหน้าตัวแปรอาร์เรย์

Rest Parameter



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Rest Parameter

ใช้ในการส่งค่าพารามิเตอร์ (Array) เข้าไปทำงานในฟังก์ชันโดยไม่จำกัดจำนวนโดยใช้เครื่องหมาย ...

Destructuring



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Destructuring (การสลายโครงสร้าง)

คือ การกำหนดค่าที่อยู่ภายใน Array หรือ Object ให้กับตัวแปร โดยใช้วิธีการจับคู่ตัวแปรกับค่าใน Array หรือ Object เช่น

```
const colors = ["ขาว", "แดง"]  
const a = colors[0] //a=ขาว  
const b = colors[1] //b=แดง
```

การใช้งาน

```
const colors = ["ขาว", "แดง"]  
const [a,b]=colors
```

Tuple



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Tuple คืออะไร

1. ชุดของตัวแปรที่อยู่ในรูปลำดับใช้เก็บค่าข้อมูลให้อยู่ในกลุ่มเดียวกัน
2. เป็นตัวแปรที่ใช้ในการเก็บข้อมูลที่มีลำดับที่ต่อเนื่อง ซึ่งข้อมูลมีค่าได้หลายค่าโดยใช้ชื่ออ้างอิงได้เพียงชื่อเดียว และใช้หมายเลขกำกับ (**index**) ให้กับตัวแปรเพื่อจำแนกความแตกต่างของค่าตัวแปรแต่ละตัว



คุณสมบัติของ Tuple

1. ใช้เก็บกลุ่มของข้อมูล
2. ข้อมูลที่อยู่ใน Tuple จะเรียกว่าสมาชิก หรือ อิลิเมนต์ (element)
3. แต่ละอิลิเมนต์ (element) จะเก็บค่าข้อมูล (value) และ อินเด็กซ์ (Index)
4. Index หมายถึงคีย์ใช้อ้างอิงตำแหน่งของ element **เริ่มต้นที่ 0**
5. สมาชิกใน Tuple มีชนิดข้อมูลต่างหรือเหมือนกันก็ได้
6. สมาชิกใน Tuple จะถูกค้นด้วยเครื่องหมายคอมม่า

คล้ายกับอาร์เรย์เลย!



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

ความแตกต่างของ Tuple และ Array

Array

- ขนาดมีความยืดหยุ่น (Dynamic Size)
- สมาชิกใน Array ต้องมีชนิดข้อมูลเหมือนกัน

Tuple

- มีขนาดตายตัว (Fixed Size)
- สมาชิกใน Tuple มีชนิดข้อมูลต่างกันก็ได้ โดยต้องระบุชนิดข้อมูลกำกับ

การสร้าง Tuple

```
const ชื่อตัวแปร : [type1,type2,...] = [value1,value2,...]
```

- สร้างตัวแปร Tuple เพื่อเก็บพิกัดแกน x ,y

```
const point:[number,number] = [5,10]
```



ตัวอย่างการสร้าง Tuple

- สร้างตัวแปร Tuple เพื่อเก็บพิกัดแกน x ,y



```
const point:[number,number] = [5,10]
```

ตัวอย่างการสร้าง Tuple

- สร้างตัวแปร Tuple เพื่อเก็บพิกัดแกน x ,y



```
const point:[number,number] = [5,10]
```

ตัวอย่างการสร้าง Tuple

- สร้างตัวแปร Tuple เพื่อเก็บพิกัดแกน x ,y



```
const point:[number,number] = [5,"kong"]
```

ตัวอย่างการสร้าง Tuple

สร้างตัวแปร Tuple เพื่อเก็บค่า HTTP Status Code

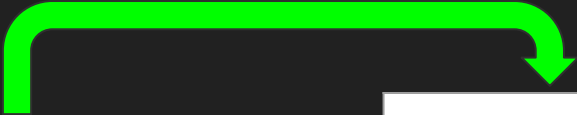
```
const status:[number , string]
```

Status Code	ความหมาย
200	OK
404	Not Found

ตัวอย่างการสร้าง Tuple

สร้างตัวแปร Tuple เพื่อเก็บค่า HTTP Status Code

`const status:[number , string]`



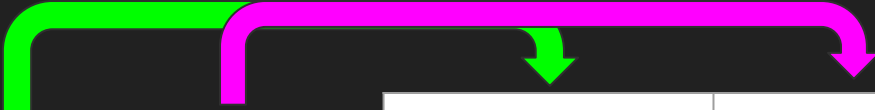
Status Code	ความหมาย
200	OK
404	Not Found



ตัวอย่างการสร้าง Tuple

สร้างตัวแปร Tuple เพื่อเก็บค่า HTTP Status Code

```
const status:[number, string]
```



Status Code	ความหมาย
200	OK
404	Not Found



Tuple Label



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

Tuple Label คืออะไร

การกำหนดชื่อกำกับภายในชนิดข้อมูลที่ทำงานอยู่ใน Tuple

- `const point:[number,number] = [5,10]`
- `const good:[number,string]=[200,"OK"]`

กำหนด Tuple Label

```
const point:[number,number] = [5,10]
```

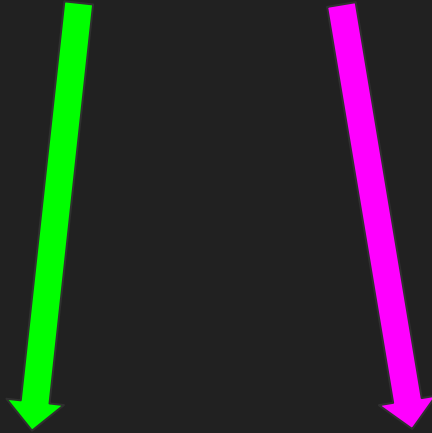


```
const point:[x:number,y:number] = [5,10]
```



กำหนด Tuple Label

```
const good:[number,string]=[200,"OK"]
```



```
const good:[code : number,status:string]=[200,"OK"]
```

การสร้าง Type แบบ Tuple



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

เก็บข้อมูล HTTP Status Code ในรูปแบบ Tuple

Status Code	ความหมาย
200	OK
400	Bad Request (คำขอไม่ถูกต้อง)
404	Not Found (ไม่พบ)

คอร์สเรียนเจาะลึก TypeScript

วิทยากร

Kong Ruksiam

Programmer , Developer

ผู้เรียนทั้งหมด รีวิว


1,912 **450**

เกี่ยวกับฉัน

โปรแกรมเมอร์และนักพัฒนาเกม รวมถึงสอนเกี่ยวกับการเขียนโปรแกรมในช่องยูทูป KongRuksiam , KongRuksiam Official และเป็นเจ้าของแฟนเพจ KongRuksiam มีผู้ติดตามมากกว่า 50,000 คน



 Facebook

 Youtube

<https://www.udemy.com/course/typescript-basic/>