

โครงการ

เรื่อง NodeJS

นายภาณุพงศ์ คงเสน่ห์ 1610701334

นายพลวิรุ์ คงอ่อน 1610701995

1451

อาจารย์ที่ปรึกษา : ผศ. สิรินคร จิยาศักดิ์

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชาหลักการเขียนโปรแกรมคอมพิวเตอร์ ๑ เพื่อเป็นประโยชน์แก่ผู้อ่านหรือผู้ที่สนใจที่จะศึกษาเกี่ยวกับภาษา NodeJS ซึ่งผู้จัดทำขึ้นเพื่อศึกษาสื่อเพื่อการศึกษา

ผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานฉบับนี้จะเป็นประโยชน์แก่ผู้อ่านหรือผู้ที่สนใจที่จะศึกษาเกี่ยวกับสื่อเพื่อการศึกษา หากมีข้อบกพร่องประการใด ผู้จัดทำขออภัยไว้ ณ โอกาสนี้ด้วย

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
คำนำ	ก
ภาษา NodeJS	1
วิธีติดตั้ง Node.js	1
การ run โปรแกรม	3
Module คืออะไร	3
รู้จักกับ npm	4
การประกาศตัวแปร	4
การแสดงผลและการรับค่าข้อมูลทางแป้นพิมพ์	4
-ฟังก์ชัน console.log();	4
-ฟังก์ชัน rl.question	5
การใช้ var	5
การใช้ let	6
การเขียนโปรแกรมแบบมีเงื่อนไข	7
-คำสั่ง If else if	7
-คำสั่ง Switch case	10
คำสั่ง For loop	10
คำสั่ง While loop	11
คำสั่ง do while	12
ภาคผนวก	13

ภาษา NodeJS

การเขียนโปรแกรมด้วยภาษา JavaScript ที่ฝั่ง server แทนที่ปกติแล้วจะเป็นฝั่ง client แต่จริงๆ แล้ว Node.js นั้นจะรวมไปถึง environment ต่างๆ ที่ทำขึ้นเพื่อให้เราเขียน JavaScript เอาไว้ที่ฝั่ง server ได้ด้วย (webserver, runtime และอื่นๆ) เรียกได้ว่ามันก็คือ platform เนื่องจาก Node.js นั้นขึ้นชื่อในด้านความเร็วของการประมวลผล จึงทำให้ application ที่เขียนด้วย Node.js นั้นมีจำนวนเพิ่มขึ้นอย่างรวดเร็ว ซึ่งรวมไปถึง

application ที่จะช่วยให้การพัฒนาเว็บไซต์เป็นไปอย่างราบรื่นมากขึ้นด้วยโดยบทความนี้จะขอเน้นไปที่วิธีการนำ application เหล่านั้นมาใช้ให้เกิดประโยชน์ในการทำเว็บไซต์

วิธีติดตั้ง Node.js

การจะนำ application ที่เขียนด้วย Node.js มาใช้งาน เราจะต้องติดตั้ง Node.js เสียก่อน => Download



Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.exe)	32-bit	64-bit
Mac OS X Installer (.pkg)	Universal	
Mac OS X Binaries (.tar.gz)	32-bit	64-bit
Linux Binaries (.tar.gz)	32-bit	64-bit
SunOS Binaries (.tar.gz)	32-bit	64-bit
Source Code	node-v0.10.29.tar.gz	

ให้เราเลือก download แบบ installer เพราะขั้นตอนการติดตั้งจะง่ายและสะดวกรวดเร็ว

Welcome to the Node.js Setup Wizard



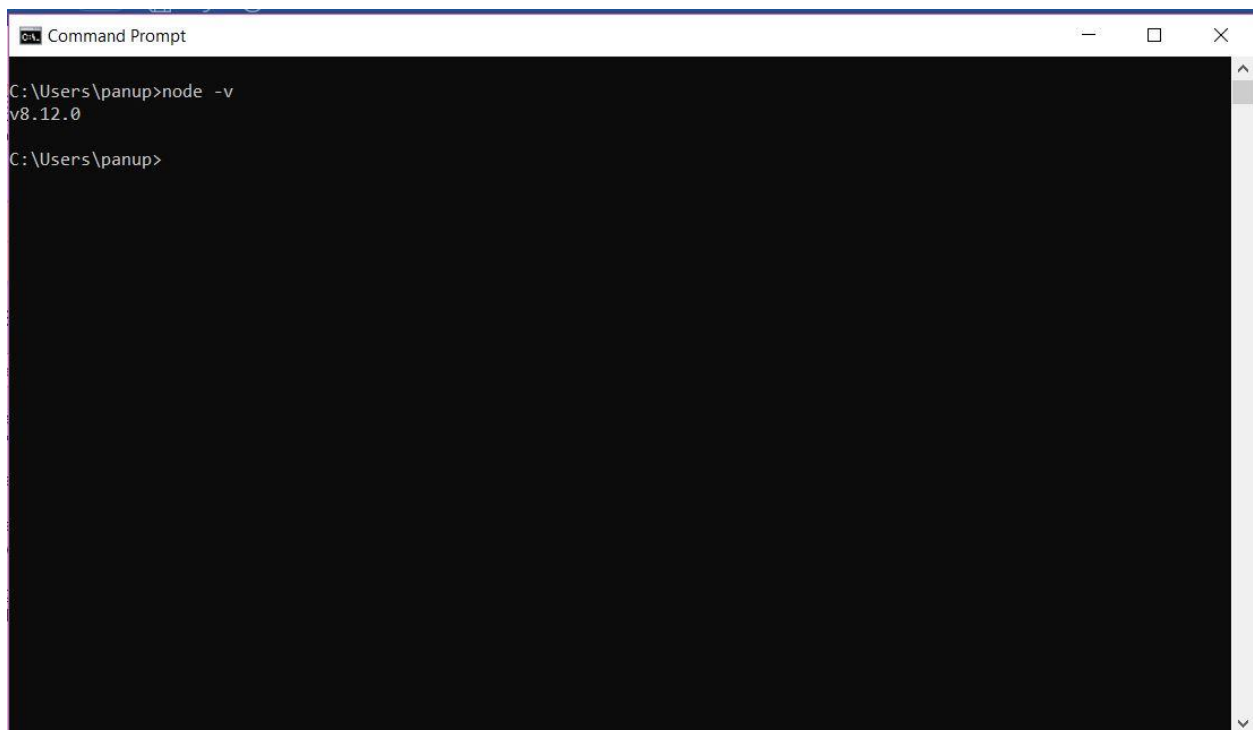
The Setup Wizard will install Node.js on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Back

Next

Cancel

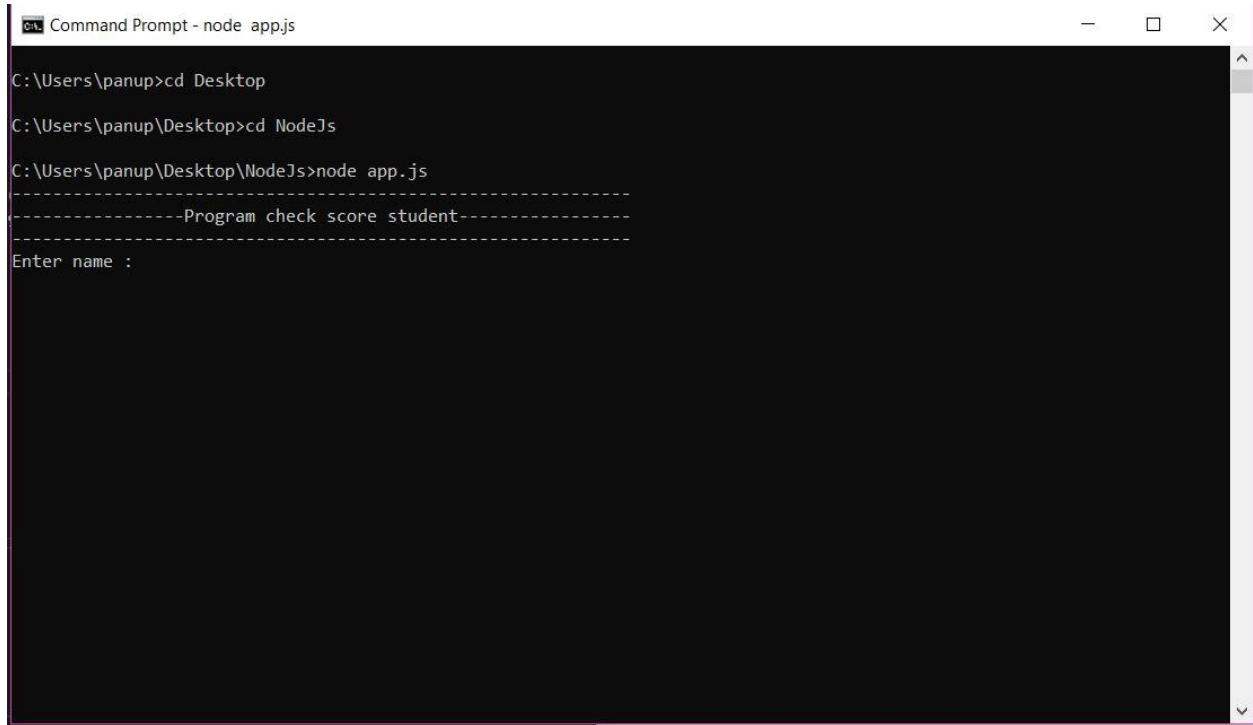
เมื่อติดตั้งเสร็จแล้ว ให้เราเปิด Command-line Interface(Command Prompt, Terminal) ขึ้นมา และพิมพ์คำสั่ง **node -v** หากติดตั้งสำเร็จก็จะแสดงเลขเวอร์ชันของ Node.JS แสดงขึ้นมา

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The command prompt shows the user's current directory as "C:\Users\panup>". The user has entered the command "node -v", and the output displayed is "v8.12.0". The prompt is now "C:\Users\panup>".

```
Command Prompt
C:\Users\panup>node -v
v8.12.0
C:\Users\panup>
```

การ run โปรแกรม

การ run โปรแกรมสามารถทำได้ โดยใช้ cmd ขึ้นมา และพิมพ์คำสั่ง node <ชื่อไฟล์> ก็จะสามารถรันโปรแกรมได้ ตามตัวอย่าง



```
Command Prompt - node app.js
C:\Users\panup>cd Desktop
C:\Users\panup\Desktop>cd NodeJs
C:\Users\panup\Desktop\NodeJs>node app.js
-----Program check score student-----
Enter name :
```

โดยที่ cd เพื่อเข้าโฟลเดอร์ที่เก็บไฟล์ไว้ จากนั้นก็ใช้คำสั่ง node ชื่อไฟล์ ก็จะสามารถรันได้

Module คืออะไร

โมดูล (Module) คือไฟล์หรือส่วนของโปรแกรมที่ใช้สำหรับกำหนดตัวแปร ฟังก์ชัน หรือคลาส โดยแบ่งย่อยอีกหน่วยหนึ่งจากโปรแกรมหลัก และในโมดูลยังสามารถประกอบไปด้วยคำสั่งประมวลผลการทำงานได้ ยกตัวอย่างเช่น เมื่อคุณเขียนโปรแกรมในภาษา Javascript คุณอาจจะมีฟังก์ชันสำหรับทำงานและจัดการกับตัวเลขเป็นจำนวนมาก และในขณะเดียวกัน คุณไม่ต้องการให้โปรแกรมหลักนั้นมีขนาดใหญ่เกินไป นั่นหมายความว่าคุณสามารถนำฟังก์ชันเหล่านี้มาสร้างเป็นโมดูล และในการใช้งานนั้นจะต้องนำเข้ามาในโปรแกรมโดยวิธีที่เรียกว่า Import

รู้จักกับ npm

npm นั้นจะถูกติดตั้งมาพร้อมกับ Node.js เพื่อทำหน้าที่จัดการ package เสริมต่างๆ ไม่ว่าจะเป็นการติดตั้ง application หรือการติดตั้ง module ต่างๆ ที่เป็น dependency ของ application เพียงแค่เราระบุชื่อ package ที่ต้องการจะใช้ มันก็จะไปตรวจสอบชื่อ package นั้นใน registry เมื่อพบแล้ว มันก็จะดาวน์โหลด package นั้นๆ มาให้เราทันที นอกจากนี้ การนำ application ที่เราเขียนเองไปเพิ่มไว้ใน registry ของ npm ก็สามารถทำได้เช่นกัน

สำหรับวิธีใช้ npm นั้นก็ไม่มีอะไรยากเลย เพียงแค่เราเปิด Command-line Interface ขึ้นมา แล้วเข้าไปยัง path ที่ต้องการจะติดตั้ง (ที่อยู่ของ project เรา) จากนั้นให้พิมพ์คำสั่งนี้

```
npm install ชื่อแพ็คเกจ
```

และ package ที่พวกเราใช้ในการเขียนโปรแกรมคือ package readline-sync

การประกาศตัวแปร

การประกาศตัวแปรของภาษา NodeJS จะมีการประกาศ 2 แบบ คือ การประกาศตัวแบบ let และ var และ const

การแสดงผลและการรับค่าข้อมูลทางแป้นพิมพ์

ฟังก์ชัน console.log();

เป็นฟังก์ชันที่ใช้ในการแสดงผลข้อมูลออกทางจอภาพ มีรูปแบบดังนี้

```
console.log("Hello" + name);
```

เป็นคำสั่งที่ไว้ใช้แสดงผลโดย สามารถเขียนข้อความ ใน “ ” ได้และสามารถแสดงตัวแปรโดยการ + (ชื่อตัวแปร) ก็จะสามารถแสดงค่าตัวแปรที่ถูกเก็บไว้ได้ ตัวอย่างเช่น

```
console.log("-----Program check score student-----");
```

จะเห็นได้ว่าโค้ดบรรทัดนี้แสดงถึงชื่อโปรแกรม Program check score student

ฟังก์ชัน rl.question

```
var name ;  
name = rl.question("Enter");
```

rl.question เป็นโค้ดที่ใช้ในการรับค่าผ่านทางแป้นพิมพ์เพื่อมาเก็บไว้ในตัวแปรจากตัวอย่างโค้ดข้างต้น จะเห็นได้ว่ามี การประกาศตัวแปรและได้ทำการเขียน (ชื่อตัวแปร) = rl.questionชื่อชนิดข้อมูล (“ข้อความของ input ที่จะแสดง”) ตามตัวอย่างข้างต้นดังนี้

```
name = rl.question("Enter name : ");  
grade = rl.questionInt("Enter score : ");  
  
if(grade>=80 && grade<=100) //condition expression  
{  
    console.log(name +" got grade A"); //statement  
}
```

โดย ตัวแปร name ก็ย้ข้อมูลมาเป็น string และตัวแปร grade ก็ย้ข้อมูลเป็นเลขจำนวนเต็ม หลังจากก็ย้ข้อมูลเสร็จ ก็จะเข้าเงื่อนไข ถ้าเกรดมากกว่าเท่ากับ 80 และเกรดน้อยกว่าเท่ากับ 100 ก็จะแสดง ชื่อที่ก็ย้และก็ย้ข้อความ

การใช้ var

```
9  
10  
11 for(var i = 0; i < 3; i++) {  
12     setTimeout(function() { console.log(i) }, 100)  
13 }  
14 console.log(i);|
```

ผลลัพธ์ที่ออกมา :

พิมพ์ 3 ออกมา 3 ครั้ง

เนื่องจาก loop ได้ทำงานจนสิ้นสุดแล้ว ทำให้ในขณะนั้น i มีค่าเป็น 3 และตัวแปร ชนิด var เป็น global scope ทำให้ในบรรทัด 14 สามารถทำงานได้และผลักตัวเองออกมานอก for หรือ อธิบายง่ายๆคือ โคนแทนค่าไปในแต่ละรอบเข้าตัวแปรเดิม

การใช้ let

```
9
10
11 for(let i = 0; i < 3; i++) {
12   setTimeout(function() { console.log(i) }, 100)
13 }
14 console.log(i);
```

ผลลัพธ์ที่ออกมา :

สำหรับค่าใน for จะได้ผลลัพธ์ออกมา คือ 0 1 และ 2

เนื่องจากการใช้ let ประกาศตัวแปร ทำให้เป็น ตัวแปรนั้นๆเป็น block scoped นั่นก็มียกเว้นอยู่ในขอบเขตของตัวเอง ในบรรทัด 14 ผลที่ออกมาจึงไม่สามารถอ้างอิงตัวแปรได้นั่นเอง

ความต่างระหว่าง let กับ const คือ let เมื่อประกาศตัวแปรแล้ว สามารถเปลี่ยนแปลงค่าได้ เมื่อเรากำหนดค่าให้กับตัวแปรนั้นๆใหม่ แต่ถ้าเป็นตัวแปรชนิด const จะไม่สามารถ ประกาศค่าใหม่ให้กับตัวแปรนั้นได้ จึงเหมาะจะใช้สำหรับประกาศตัวแปรค่าคงที่เช่น ค่า PI เป็นต้น

ยกตัวอย่างเช่น

```
9
10
11 let a = 0
12 a = 1
13
14 const PI = 3.14
15 PI = 1
16
17
18 |
```

ค่า PI เมื่อเป็นค่าคงที่ จะไม่สามารถเปลี่ยนแปลงค่าได้อีก

สำหรับบทความนี้ ฝากไว้ให้นักพัฒนาได้เปลี่ยนมาใช้ ชนิดตัวแปรให้ตรงกับขอบเขตของการใช้งาน เพื่อที่จะสามารถอ่านแล้วเข้าใจง่าย เข้าใจชนิดของตัวแปรมากยิ่งขึ้น

การเขียนโปรแกรมแบบมีเงื่อนไข

1.if else if

2.switch case

If else if

If เป็นโครงสร้างภาษาที่สำคัญอย่างหนึ่ง ใช้ในการตรวจสอบเงื่อนไขการทำงาน และกำหนดทิศทางในการทำงานที่เหมาะสมกับเงื่อนไขนั้น ซึ่งโครงสร้างของ if ก็เป็นเช่นเดียวกับภาษา C คือ

```
if(grade>80) //expression|
{
    //statement
}
```

โดย if จะตรวจสอบว่า เงื่อนไขใน expression เป็น จริง (TRUE) หรือไม่ ถ้าเป็นจริง ก็จะเข้าไปทำคำสั่งใน statement ถ้า ไม่เป็นจริง (FALSE) ก็จะ ไม่ทำคำสั่งใน statement ซึ่ง statement นี้ อาจจะเป็นคำสั่งเพียง คำสั่งเดียว หรือว่า เป็นชุดคำสั่งก็ได้ ซึ่งถ้าเป็นชุดคำสั่ง ก็ต้องมี วงเล็บปีกกา ครอบ ชุดคำสั่งนั้นด้วย จากตัวอย่าง

ELSE

ในกรณีที่ต้องการให้มีทางเลือก มากกว่าหนึ่งทาง โดยทำเงื่อนไขเป็นจริง ก็ให้ทำ ทางเลือกหนึ่ง ถ้าไม่เป็นจริง ก็ให้ทำ อีกทางเลือกหนึ่ง ก็สามารถทำได้โดยใช้ else เข้ามาช่วยใน if ดังนี้

```
if(grade>80) //expression
{
    //statement
}else
{
    //statement
}
```

```
if(grade>=80 && grade<=100)
{
    console.log("You got Grade : A");
}else
{
    console.log("Try again");
}
```

จากตัวอย่าง ถ้าเงื่อนไขเป็นจริง ก็จะทำคำสั่งชุดแรก ถ้าเงื่อนไขไม่เป็นจริง ก็จะไปทำใน else หรือว่า คำสั่งชุดที่สอง แทน

ELSEIF

elseif เป็นการนำเอา else มารวมกับ if โดยเมื่อ เงื่อนไขแรกไม่เป็นจริง และต้องมาทำคำสั่งใน else ก็จะทำการ ตรวจสอบเงื่อนไขใน if ที่อยู่ก่อน else ทันที ซึ่งจะมีรูปแบบดังนี้

```
if(expression - 1 )
{
    steaement 1
}else if (expression - 2)
{
    statement 2
}else
{
    statement 3
}
```

```
if(grade>=80 && grade<=100)
{
    console.log(" got grade A");
}else if (grade>=70 && grade<=79)
{
    console.log(" got grade B");
}else
{
    console.log("try agian");
}
```

จากตัวอย่าง จะเห็นว่ามีการตรวจสอบเงื่อนไขว่า grade>=80 และ grade<=100 ว่าเป็นจริงหรือไม่ ถ้าเป็นจริง ก็จะทำคำสั่งชุดแรก แล้วก็ออกจากเงื่อนไข if เลย แต่ถ้าไม่จริง ก็จะไปตรวจสอบเงื่อนไขที่สองคือ grade>=70 และ grade<=79 ว่าเป็นจริงหรือไม่ ถ้าเป็นจริงก็จะทำคำสั่งชุดที่สอง แต่ถ้าเป็นเท็จ ก็จะไปทำคำสั่งในชุดที่สามเลย

การใช้ else if นั้น จะใช้ซ้อนกันกี่ชั้นก็ได้ ซึ่งการตรวจสอบเงื่อนไข ก็จะทำแบบไล่เรื่อยๆ ถ้าเจอเงื่อนไขที่เป็นจริงก็จะแสดงผล และออกจากเงื่อนไข

คำสั่ง Switch case

คำสั่ง switch-case นั้นคล้ายกับ คำสั่ง If-else เป้าหมายของมันเพื่อตรวจสอบกับค่าคงที่ นี่เป็นตัวอย่างการใช้คำสั่ง switch

```
switch (n) {  
  case 1:  
    cout << "n is 1";  
    break;  
  case 2:  
    cout << "n is 2";  
    break;  
  default:  
    cout << "Unknown n";  
}
```

Switch expression สามารถมีได้แค่หนึ่งค่าเพื่อประเมิน คำสั่ง case เป็นคำสั่งเงื่อนไขเพื่อเปรียบเทียบค่า ในตัวอย่าง case 1: จะทำงานเมื่อ n มีค่าเท่ากับ 1 หลังจากคำสั่งด้านล่างเราต้องใส่คำสั่ง break เพื่อหยุดสำหรับแต่ละ case ไม่เช่นนั้น โปรแกรมจะทำงานไปจนกว่าจะพบคำสั่ง break หรือสิ้นสุดบล็อกคำสั่งของ switch และคำสั่ง default นั้นเป็นทางเลือกเมื่อโปรแกรมไม่ตรงกับเงื่อนไขใดๆ ก่อนหน้าเช่นเดียวกับคำสั่ง else

คำสั่งวนซ้ำ

คำสั่งวนซ้ำ (loop statements) ถูกใช้เพื่อควบคุมโปรแกรมเพื่อให้โปรแกรมทำบางส่วนของโค้ดซ้ำๆ มีรูปแบบหลายประเภทที่เราสามารถใช้ได้ในภาษา Nodejs โดยคำสั่ง while , do while ,for

คำสั่ง For loop

```
for(initialize; condition; iterator)  
{  
  statements ;  
}
```

ในการใช้ for loop เราสามารถกำหนดค่าเริ่มต้น สร้างเงื่อนไข (expression) และทำการกำหนดการเพิ่มค่าหรือลดค่าของตัวทำซ้ำได้ก่อนที่จะเริ่มทำงาน มันจะมีประโยชน์ที่จะให้สะดวกเมื่อเรารู้จำนวนรอบที่แน่นอนในการวนซ้ำของโปรแกรม

```
for(var i= 1; i<=10; i++)
{
  console.log(i);
}
```

ในตัวอย่างเป็นการใช้ for loop เพื่อแสดงตัวเลข 1 ถึง 10 เช่นกัน สังเกตว่าการกำหนดค่าเริ่มต้น การสร้างเงื่อนไขและการเพิ่มค่าจะถูกใส่ไว้ที่ส่วนหัวของโปรแกรม

คำสั่ง While loop

```
while (expression) {
  statements
}
```

while loop นั้นจะใช้ในการทำซ้ำส่วนของโปรแกรมในขณะที่ expression เป็น true จนกว่า expression เป็น false โปรแกรมจะออกจาก while loop และทำงานต่อไปหลังจากลูป

```
var n = 1;
while (n <=10) {
  console.log("Loop" + n);
  n++;
}
```

ในตัวอย่าง เป็นตัวอย่างในการใช้ while loop เพื่อนับเลข โดยโปรแกรมจะนับเริ่มจาก 1 ถึง 10 เราได้ลูปของ while ครบใดที่ค่าของตัวแปร n ยังน้อยกว่าหรือเท่ากับ 10 ในลูปก็จะแสดงค่าของ n และเพิ่มค่าขึ้นในแต่ละรอบโดยคำสั่ง n++ เพื่อไม่ให้โปรแกรมอยู่ในลูปตลอดไป

คำสั่ง do while

```
do {  
    statements  
} while (expression) ;
```

สิ่งที่แตกต่างของ do-while loop คือการตรวจสอบ expression หลังจากการทำงานรอบแรกเสร็จสิ้น นั่น

หมายความว่า do-while loop จะต้องทำงานอย่างน้อย 1 รอบ สังเกตว่า expression จะอยู่ที่ตอนท้ายของลูป มาดูตัวอย่าง

```
var n ;  
console.log("Enter 0 to exit the loop")  
do {  
    n = rl.questionInt("Enter number : ");  
    console.log("Number you entered is " + n);  
} while (n != 0) ;  
console.log("Loop ended");|
```

ในตัวอย่าง เป็นการใช้ do-while loop เพื่อถามให้ผู้ใช้ป้อนตัวเลข (0) เพื่อออกจากลูป ถ้าตัวเลขที่ใส่เข้ามานั้นไม่ใช่ศูนย์ โปรแกรมก็จะถามให้ใส่ใหม่เรื่อยไป และเมื่อใส่เลขศูนย์ มันทำให้ expression ของเราเป็น false และทำให้โปรแกรมออกจากลูป

ภาคผนวก

Codeในการเขียน NodeJs โดยใช้ npm package readline-sync


```
Command Prompt
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\panup>cd Desktop

C:\Users\panup\Desktop>cd NodeJs

C:\Users\panup\Desktop\NodeJs>node app.js
-----
-----Program check score student-----
-----
Enter name : pasawee kongon
Enter score : 90
pasawee kongon got grade A

Do you want to exit Y/N : n

Enter name : panupong kongsanae
Enter score : 75
panupong kongsanae got grade B

Do you want to exit Y/N : y

-----
-----
```

ภาพในการ run โปรแกรม และ คีย์ข้อมูลเกรด