

WEB MANAGEMENT 2

บทที่ 11 คำสั่งแบบวนซ้ำ Loops

Jenjira Jaimunk, PhD



เนื้อหาของวันนี้

- คำสั่งแบบวนซ้ำ Loops
- คำสั่ง "while"
- คำสั่ง "do...while"
- คำสั่ง "for"

คำสั่งแบบวนซ้ำ Loops

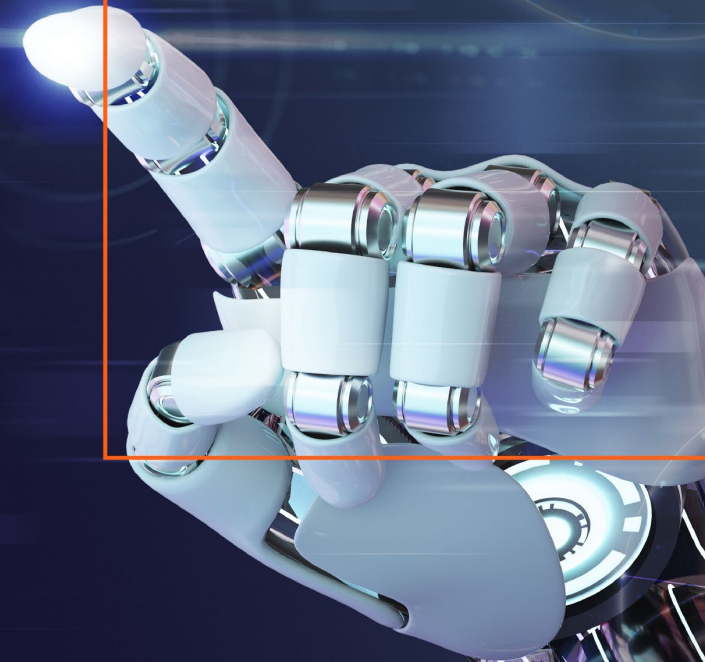
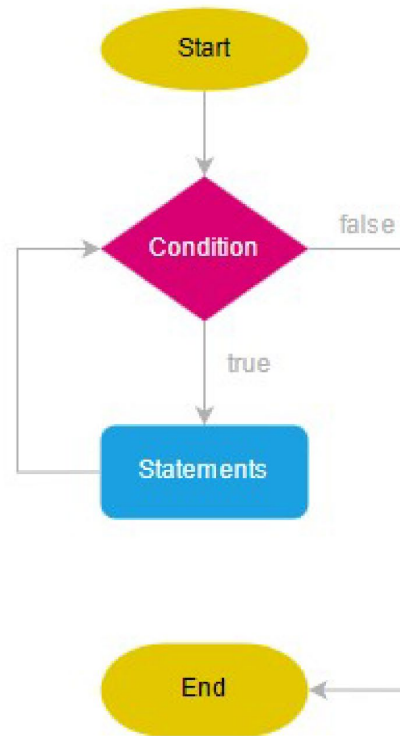
- ลูปเป็นวิธีการทำซ้ำโค้ดเดิมหลายๆ ครั้ง
- ตัวอย่างเช่น การเรียกสินค้าจากรายการที่ละรายการหรือการเรียกสินค้าจากรายการโดยใช้รหัสเดียวกัน ตั้งแต่ 1 ถึง 10 ครั้ง



คำสั่ง "while"

- คำสั่ง while จะประเมินนิพจน์ก่อนการวนซ้ำแต่ละครั้ง
- หากนิพจน์ประเมินเป็นจริง โค้ดจากเนื้อความวนรอบจะถูกดำเนินการ
- หากนิพจน์ประเมินเป็นเท็จก่อนที่จะลูปจะเข้ามา ลูป while จะไม่ดำเนินการ
- เนื่องจาก while loop ประเมินนิพจน์ก่อนการวนซ้ำแต่ละครั้ง จึงเรียกว่า pretest loop

```
while (condition) {  
    // code  
    // so-called "loop body"  
}
```



คำสั่ง "while" - ตัวอย่าง

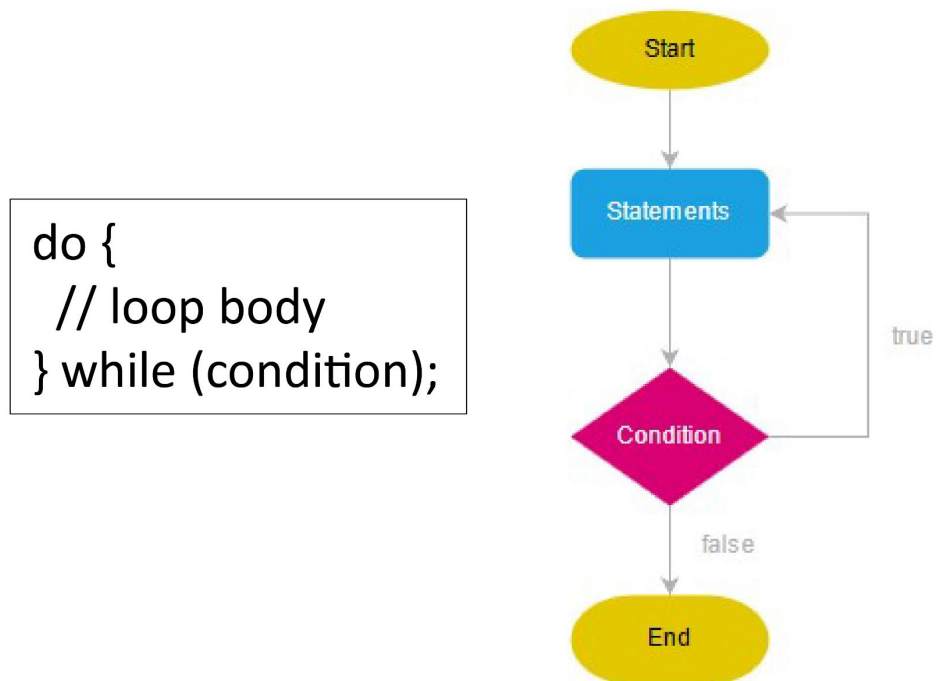
```
<script language = "javascript" type = "text/javascript">  
    let count = 1;  
    while (count < 10) {  
        console.log(count);  
        count +=2;  
    }  
</script>
```

```
>_ Console (beta) 5 0 0 0 0 0 Clear console Minimize  
JSFiddle Console (beta). Turn on/off in Editor settings.  
▲ "Running fiddle"  
▲ "Running fiddle"  
1  
3  
5  
7  
9  
>_
```



คำสั่ง "do...while"

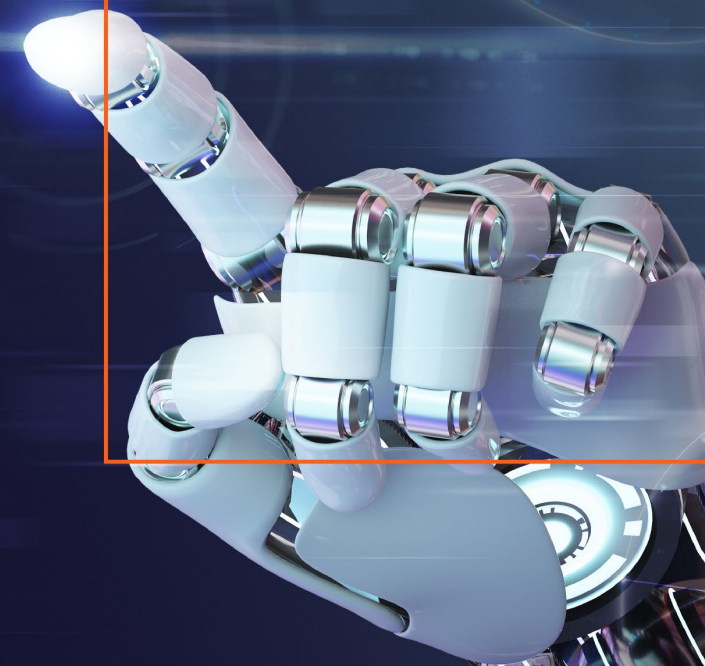
- คำสั่ง do...while จะสร้างลูปที่รันบล็อกจนกว่าเงื่อนไขจะประเมินเป็นเท็จ



คำสั่ง "do...while" - ตัวอย่าง

```
let count = 0;  
do {  
  console.log(count);  
  count++;  
} while (count < 5)
```

```
>_ Console (beta)  ⓘ 5 ⓘ 0  
☁ "Running fiddle"  
0  
1  
2  
3  
4
```



คำสั่ง "do...while" - ตัวอย่าง

```
const MIN = 1;
const MAX = 10;

let secretNumber = Math.floor(Math.random() * (MAX - MIN + 1)) + MIN;

let guesses = 0; // for storing the number of guesses
let hint = ""; // for storing hint
let number = 0;
do {
  // get input from user
  let input = prompt(`Please enter a number between ${MIN} and ${MAX}` + hint);

  // get the integer
  number = parseInt(input);

  // increase the number of guesses
  guesses++;

  // check input number with the secret number provide hint if needed
  if (number > secretNumber) {
    hint = ', and less than ' + number;
  } else if (number < secretNumber) {
    hint = ', and greater than ' + number;
  } else if (number == secretNumber) {
    alert(`Bravo! you're correct after ${guesses} guess(es).`);
  }
} while (number != secretNumber);
```

An embedded page at fiddle.jshell.net says

Please enter a number between 1 and 10

OK

Cancel

An embedded page at fiddle.jshell.net says

Please enter a number between 1 and 10, and greater than 1

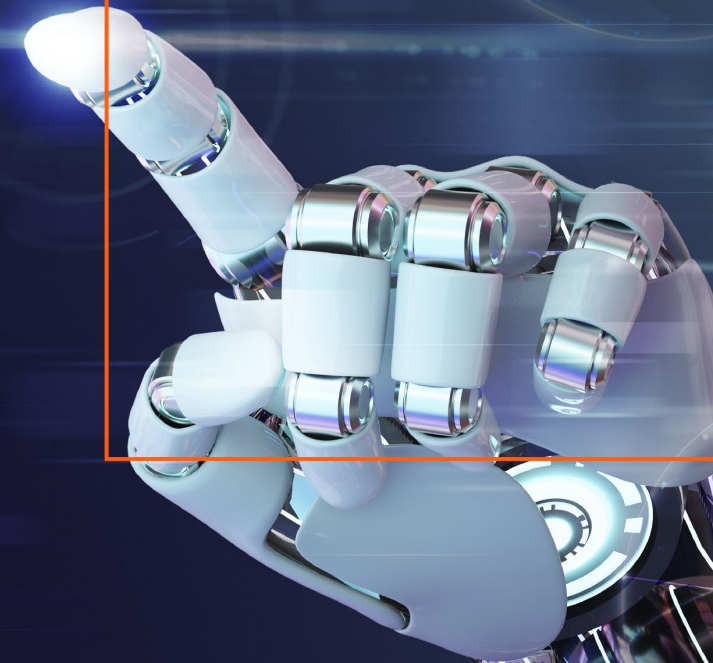
OK

Cancel

An embedded page at fiddle.jshell.net says

Bravo! you're correct after 10 guess(es).

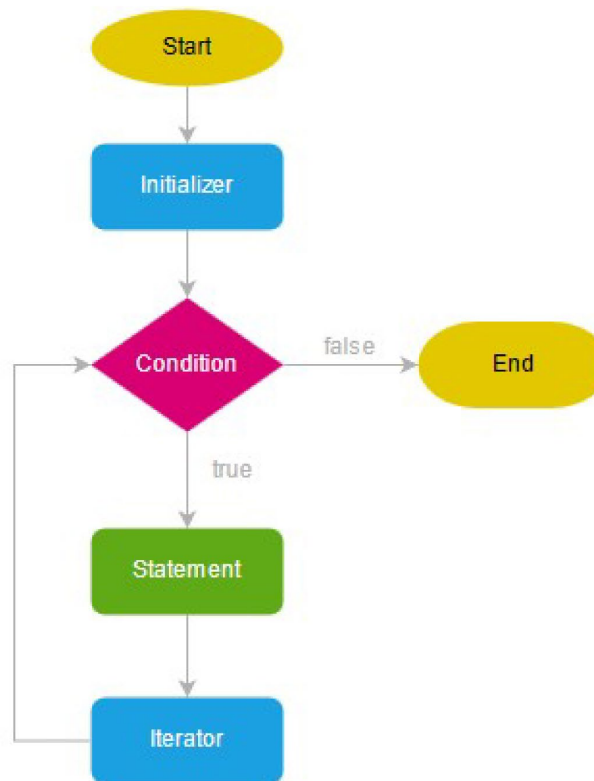
OK



คำสั่ง "for"

- คำสั่ง for สร้างลูปที่มีสามนิพจน์ทางเลือก

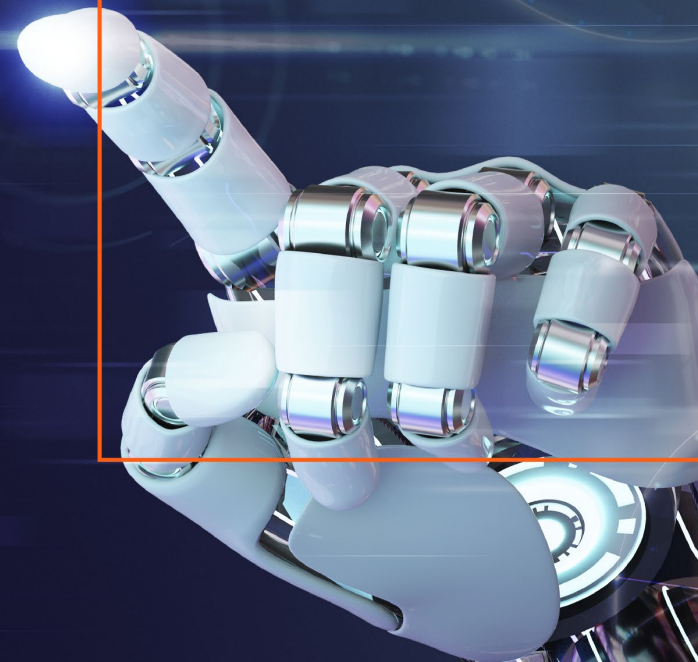
```
for (initializer; condition; iterator) {  
    // statements  
}
```



คำสั่ง "for" - ตัวอย่าง

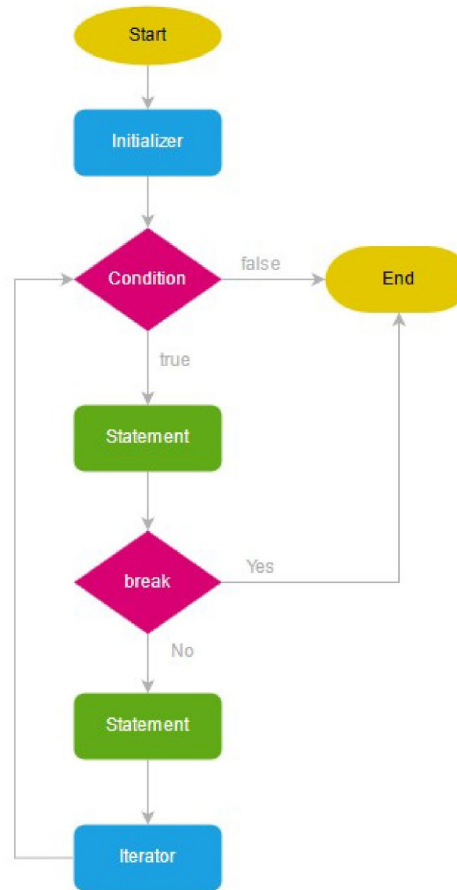
```
for (let i = 1; i < 5; i++) {  
  console.log(i);  
}
```

```
>_ Console (beta) 4 0 0 0 0  
JSFiddle Console (beta). Turn on/off in Editor settings.  
☁ "Running fiddle"  
☁ "Running fiddle"  
☁ "Running fiddle"  
1  
2  
3  
4
```



คำสั่ง Break

- โดยปกติ ลูปจะออกเมื่อเงื่อนไขของลูปเป็นเท็จ
- แต่เราสามารถบังคับให้ออกได้ตลอดเวลาโดยใช้คำสั่ง Break



คำสั่ง Break - ตัวอย่าง

```
let sum = 0;

while (true) {

    let value = +prompt("Enter a number", "");

    if (!value) break; // (*)

    sum += value;

}

alert( 'Sum: ' + sum );
```

An embedded page at fiddle.jshell.net says

Enter a number

OK

Cancel

An embedded page at fiddle.jshell.net says

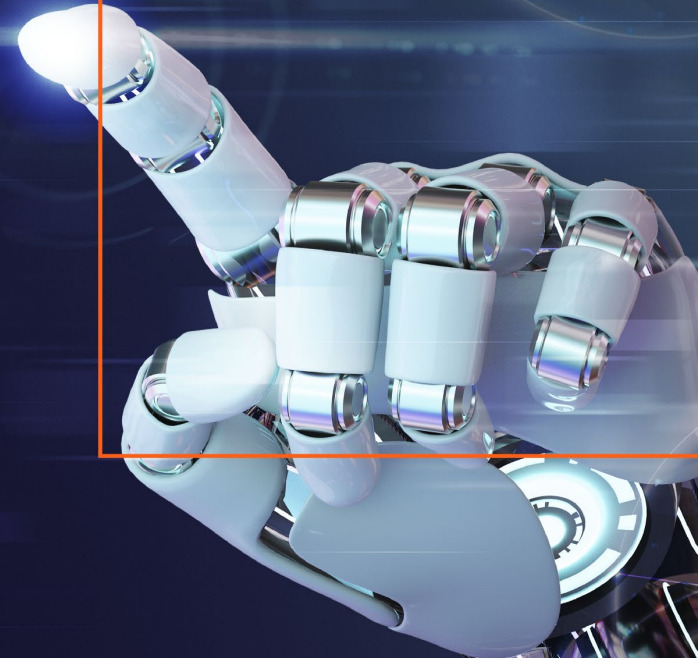
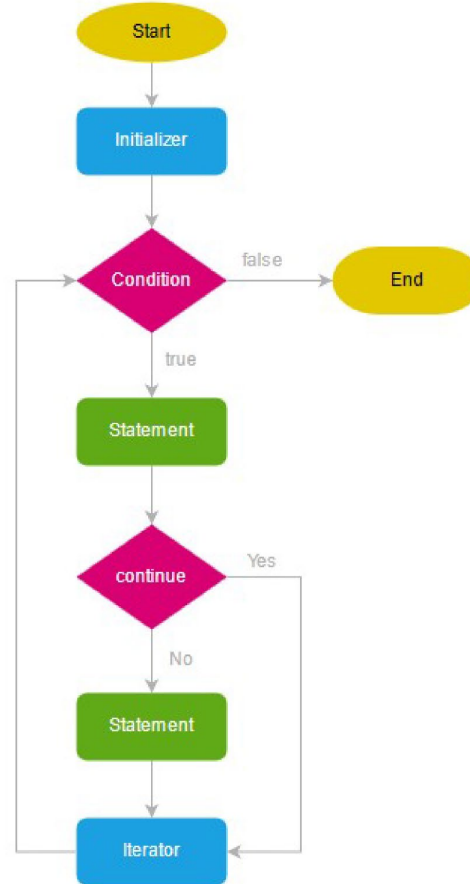
Sum: 10

OK



คำสั่ง continue

- เมื่อใช้คำสั่ง continue ใน for loop จะไม่สิ้นสุดการวนซ้ำทั้งหมด
- แต่จะข้ามไปที่นิพจน์ตัววนซ้ำแทน



คำสั่ง continue - ตัวอย่าง

```
for (let i = 0; i < 10; i++) {  
  if (i % 2 === 0) {  
    continue;  
  }  
  console.log(i);  
}
```

```
>_ Console (beta)  ⓘ 9 ⓘ 0 ⚠ 0 ⓘ 0  
☁ "Running fiddle"  
☁ "Running fiddle"  
1  
3  
5  
7  
9  
>_
```

