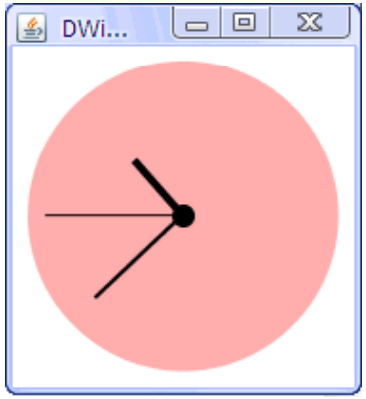
การทำงานแบบวงวนและเลือกทำ

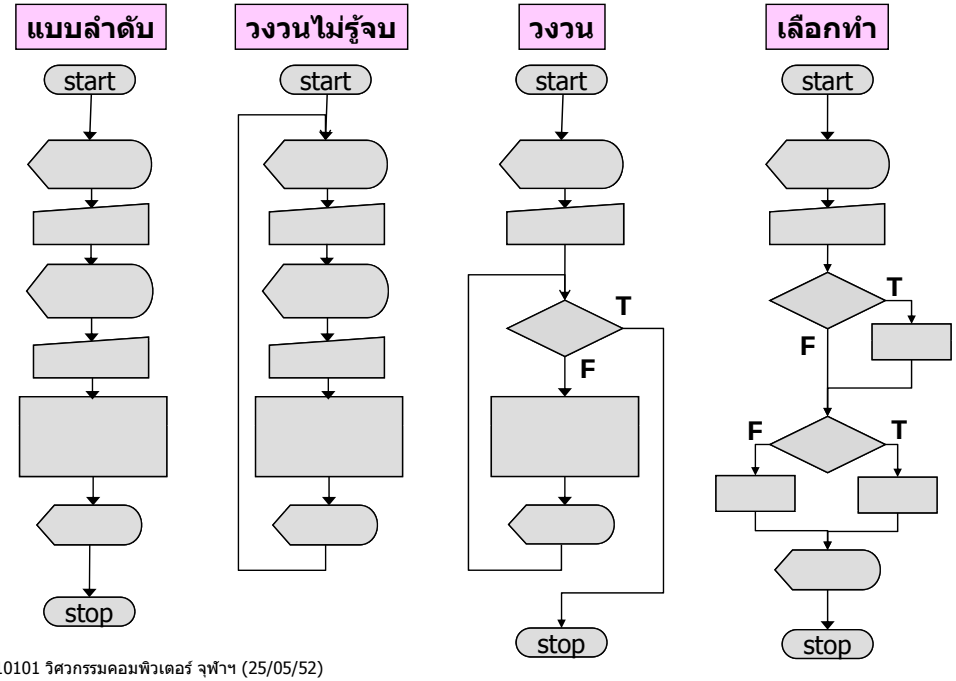
วงวน (loop)



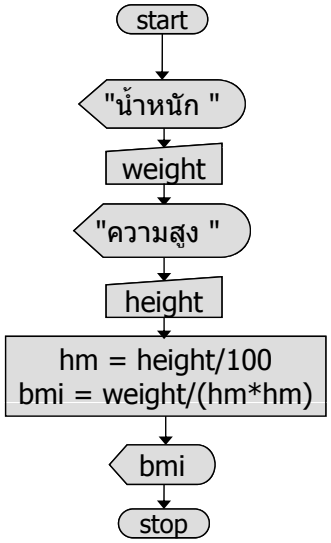
หัวข้อ

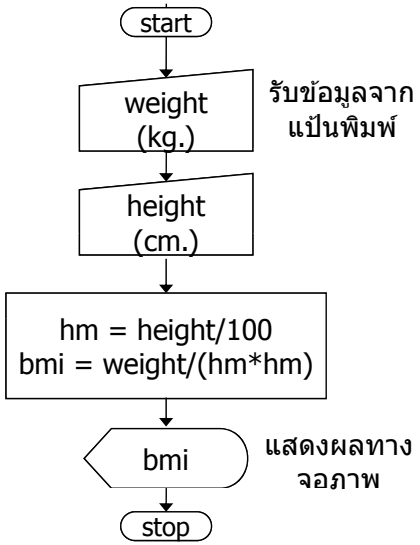
- ผังงาน
- วงวนไม่รู้จบ : `while(true)`
- ตัวดำเนินการ : `++ -- % < == > !=`
- การกระโดดออกจากวงวน : `if (...) break;`
- วงวนมาตรฐาน : `while`
- การเลือกทำ : `if, if-else`
- ตัวอย่าง ๆ ๆ

ผังงาน (Flowchart)

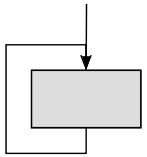


ผังงาน : บรรยายการทำงานหลัก





วงวนไม่รู้จบ : while(true) { ... }



```
while ( true ) {
  คำสั่ง 1
  คำสั่ง 2
  คำสั่ง 3
}
```

```
public static void main(String[] args) {
  int k = 0;
  while (true) {
    System.out.println("k = " + k);
    k = k + 1;
  }
}
```

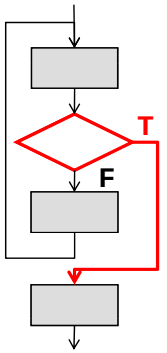

ดัชนีมวลกาย

```
import java.util.Scanner;
public class BodyMassIndex {
  public static void main(String[] args) {
    Scanner kb = new Scanner(System.in);
    while ( true ) {
      System.out.print("น้ำหนัก (kg.) = ");
      double weight = kb.nextDouble();
      System.out.print("ความสูง (cm.) = ");
      double height = kb.nextDouble();
      double hm = height / 100.0;
      double bmi = weight / (hm * hm);
      System.out.println("ดัชนีมวลกาย = " + bmi);
    }
  }
}
```

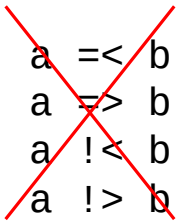
$$BMI = \frac{weight(kg.)}{height^2(m^2)}$$

ทำงานในวงวนไม่รู้จบ
ผู้ใช้ต้องหยุดโปรแกรมเอง

การกระโดดออกจากวงวน




```
while ( true ) {
  คำสั่ง
  if ( เงื่อนไข ) break;
  คำสั่ง
}
คำสั่ง
...
```



บรรยายเงื่อนไขด้วย นิพจน์ตรรกะ (Boolean expression)
ใช้การเปรียบเทียบในการกำหนดเงื่อนไข
ด้วยตัวเปรียบเทียบข้อมูล < > <= >= != ==

```
if ( a == b  ) break;
```

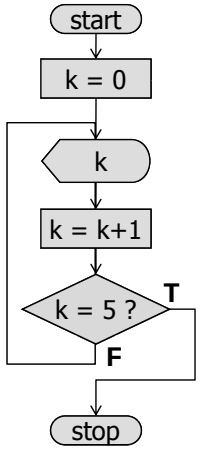
ผลการเปรียบเทียบจะได้ค่า true หรือ false

ดัชนีมวลกาย : เลิกโปรแกรมเมื่อน้ำหนักเป็นลบ

```
import java.util.Scanner;
public class BodyMassIndex {
  public static void main(String[] args) {
    Scanner kb = new Scanner(System.in);
    while ( true ) {
      System.out.print("น้ำหนัก (kg.) = ");
      double weight = kb.nextDouble();
      if (weight < 0) break;
      System.out.print("ความสูง (cm.) = ");
      double height = kb.nextDouble();
      double hm = height / 100.0;
      double bmi = weight / (hm * hm);
      System.out.println("ดัชนีมวลกาย = " + bmi);
    }
  }
}
```


ตัวอย่าง : แสดงเลขนับขึ้น

```
public static void main(String[] args) {
  int k = 0;
  while (true) {
    System.out.println("k = " + k);
    k = k + 1;
    if (k == 5) break;
  }
}
```



โปรแกรมนี้แสดงอะไร

```
k = 1
k = 2
k = 3
k = 4
k = 5
```

```
k = 0
k = 1
k = 2
k = 3
k = 4
k = 5
```

```
k = 0
k = 1
k = 2
k = 3
k = 4
```


เขียนได้หลายแบบ

```
int k = 0;
while (true) {
  System.out.println("k = " + k);
  k = k + 1;
  if (k == 5) break;
}
```

```
int k = 0;
while (true) {
  if (k == 5) break;
  System.out.println("k = " + k);
  k = k + 1;
}
```

```
int k = 0;
while (true) {
  if (k >= 5) break;
  System.out.println("k = " + k);
  k = k + 1;
}
```


ใช้บ่อย : เพิ่มอีกหนึ่ง (++) ลดลงหนึ่ง (--)

- `c = c + 1; c += 1; c++; ++c;`
- `c = c - 1; c -= 1; c--; --c;`
- `++` กับ `--` ใช้กับตัวแปรเท่านั้น
  `(a + b)++` ไม่ได้

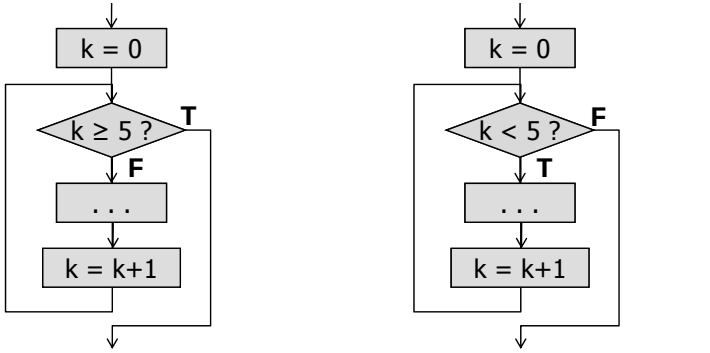
```
int k = 0;
while (true) {
  if (k >= n) break;
  // ...
  k++;
}
```

วงวนแบบ while

```
int k = 0;
while (true) {
  if (k >= 5) break;
  ...
  k++;
}
```

```
int k = 0;
while (k < 5) {
  ...
  k++;
}
```

ส่วนใหญ่เขียนแบบนี้



ตัวอย่าง : ค่าเฉลี่ยของข้อมูล 10 ตัว

```
import java.util.Scanner;
public class Average {
  public static void main(String[] args) {
    Scanner kb = new Scanner(System.in);
    double sum = 0;
    int k = 0;
    while (k < 10) {
      System.out.print("ข้อมูล = ");
      double data = kb.nextDouble();
      sum = sum + data;
      k = k + 1;
    }
    double average = sum / 10.0;
    System.out.println("ค่าเฉลี่ย = " + average);
  }
}
```


ตัวอย่าง : ค่าเฉลี่ยของข้อมูล *n* ตัว

```
import java.util.Scanner;
public class Average {
  public static void main(String[] args) {
    Scanner kb = new Scanner(System.in);
    System.out.print("จำนวนข้อมูล = ");
    int n = kb.nextInt();
    double sum = 0;
    int k = 0;
    while ( k < n ) {
      System.out.print("ข้อมูล = ");
      double data = kb.nextDouble();
      sum = sum + data;
      k = k + 1;
    }
    double average = sum / n ;
    System.out.println("ค่าเฉลี่ย = " + average);
  }
}
```


ตัวอย่าง : ค่าเฉลี่ยของชุดข้อมูล (เลิกเมื่อพบลบ)

```
import java.util.Scanner;
public class Average {
  public static void main(String[] args) {
    Scanner kb = new Scanner(System.in);
    double sum = 0;
    int k = 0;
    while ( true ) {
      System.out.print("ข้อมูล = ");
      double data = kb.nextDouble();
      if (data < 0) break;
      sum = sum + data;
      k = k + 1;
    }
    double average = sum / k ;
    System.out.println("ค่าเฉลี่ย = " + average);
  }
}
```

ตัวอย่าง : หารากที่สอง (บาบิโลน) แบบใช้วงวน

```
public static void main(String[] args) {
  Scanner kb = new Scanner(System.in);
  System.out.print("a = ");
  double a = kb.nextDouble();
  double x = 1.0;
  while (true) {
    x = (x + a/x) / 2.0;
    if ( Math.abs(x*x - a) <= 1e-5   ) break;
  }
  System.out.println("รากที่สอง = " + x );
}
```

$$x_k = \frac{1}{2}\left(x_{k-1} + \frac{a}{x_{k-1}} \right)$$

$$| x^2 - a | < 10^{-5}$$

ต้องระวังเรื่องความเท่ากันของ double
เช่น 0.1+0.1+0.1 ไม่เท่ากับ 0.3
(0.1+0.1+0.1 – 0.3) ≈ 5.6×10⁻¹⁷

double : ละเอียดแต่ไม่ 100%

```
public static void main(String[] args) {
  double a = 0.1;
  double b = a + a + a;
  System.out.println(b – 0.3);
}
```

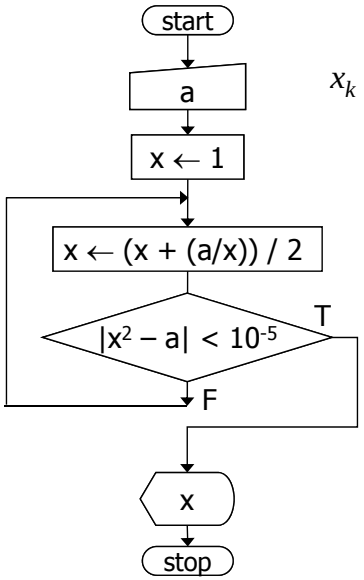
5.551115123125783E-17

- คอมพิวเตอร์เก็บ double หนึ่งตัวด้วยเนื้อที่ 64 บิต
- มีขนาดจำกัด จึงมีบางค่าเก็บได้ไม่แม่นยำ 100%
- จึงต้องระวังการเปรียบเทียบความเท่ากันของ double
- ควรใช้การเปรียบเทียบการมีค่าใกล้กันมาก ๆ แทน

$$| a - b | < 10^{-10}$$

$$| a - b | / \max(|a|, |b|) < 10^{-10}$$

ผังงาน : การหารากที่สองของ a



$$x_k = \frac{1}{2}\left(x_{k-1} + \frac{a}{x_{k-1}} \right)$$

ปรับวงวนของการหารากที่สอง

```
while (true) {
  x = (x + a/x) / 2.0;
  if (Math.abs(x*x – a) <= 1e-5) break;
}
```

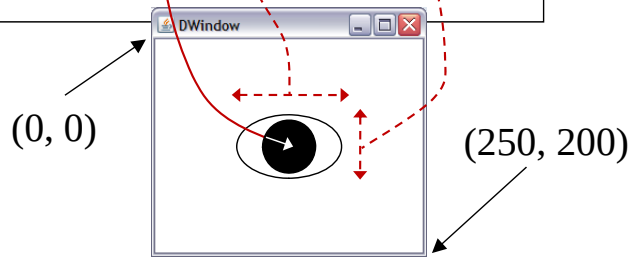
```
while (true) {
  if (Math.abs(x*x – a) <= 1e-5) break;
  x = (x + a/x) / 2.0;
}
```

```
while (Math.abs(x*x – a) > 1e-5) {
  x = (x + a/x) / 2.0;
}
```

DWindow

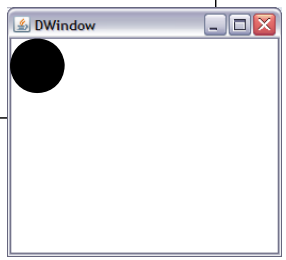
คลาสเสริมของ JLab ช่วยวาดเส้น วงรี สี่เหลี่ยม

```
import jlab.graphics.DWindow;
public class Eye {
  public static void main(String[] args) {
    DWindow w = new DWindow(250, 200);
    w.drawEllipse(125, 100, 100, 60);
    w.fillEllipse(125, 100, 50, 50);
  }
}
```



ลูกบอลเคลื่อนที่

```
import jlab.graphics.DWindow;
public class MovingBall {
  public static void main(String[] args) {
    DWindow w = new DWindow(250, 200);
    double x = 0, y = 0;
    while (true) {
      w.fade(1);      // ภาพในวินโดว์จางลง 100% (ลบหมด)
      w.fillEllipse(x, y, 50, 50);
      x = x + 2;      // ปรับตำแหน่ง x เพื่อใช้ในรอบถัดไป
      y = y + 1;      // ปรับตำแหน่ง y เพื่อใช้ในรอบถัดไป
      w.sleep(100);   // หยุดการทำงาน 100msec.
    }
  }
}
```



ลอง `w.fade(0.1)` ดู

ตัวดำเนินการเสริม

- เศษจากการหาร : %

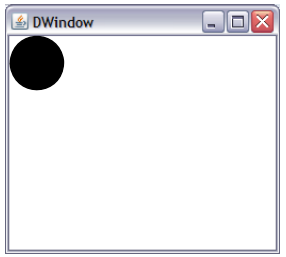
```
  5 % 2  ได้  1
5.2 % 2  ได้  1.2
```

```
import jlab.graphics.DWindow;
public class MovingBall {
  public static void main(String[] args) {
    DWindow w = new DWindow(250, 200);
    double x = 0, y = 0;
    while (true) {
      w.fade(1);
      w.fillEllipse(x, y, 50, 50);
      x = (x + 2) % 250;
      y = (y + 1) % 200;
      w.sleep(100);
    }
  }
}
```

ต้องการให้ลูกบอลออกขวา วนกลับมาทางซ้าย

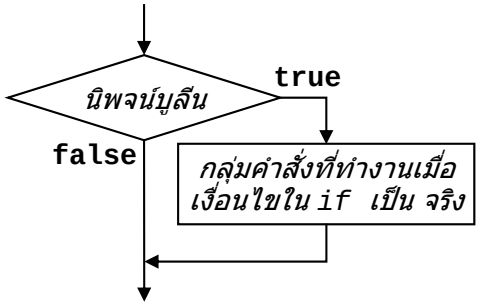
```
x = 246
x = (246 + 2) % 250 = 248 % 250 = 248
x = (248 + 2) % 250 = 250 % 250 = 0
```


ทำอย่างไรให้ลูกบอลเด้งขอบวินโดว์



ต้องใช้คำสั่งทดสอบ
ให้กลับทิศการเคลื่อนที่เมื่อชนขอบ

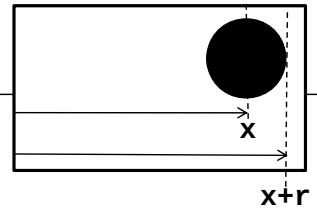
if : จริงถึงทำ




```
if ( นิพจน์ที่ได้ค่า จริง หรือ เท็จ ) {
   กลุ่มคำสั่งที่ทำงานเมื่อเงื่อนไขใน if เป็น จริง
}
```


ลูกบอลเด้งผนัง

```
public static void main(String[] args) {
  DWindow w = new DWindow(250, 200);
  double x = 0, y = 0, dx = 2, dy = 1, r = 25;
  while (true) {
    w.fade(1);
    w.fillEllipse(x, y, 2*r, 2*r);
    x =  x + dx;
    y =  y + dy;
    if (x + r > 250) { dx = -dx; }
    if (x - r < 0  ) { dx = -dx; }
    if (y + r > 200) { dy = -dy; }
    if (y - r < 0  ) { dy = -dy; }
    w.sleep(100);
  }
}
```



ตรวจสอบความถูกต้องก่อนทำงาน

```
public static void main(String[] args) {
  Scanner kb = new Scanner(System.in);
  System.out.print("a = ");
  double a = kb.nextDouble();
  if (a < 0) {
    System.out.println("a ต้องไม่เป็นลบ " );
    System.exit(-1);
  }
  double x = 1.0;
  while (Math.abs(x*x – a) > 1e-5) {
    x = (x + a/x) / 2.0;
  }
  System.out.println("รากที่สอง = " + x );
}
```

ใช้ `return;` ก็ได้

ถ้า **a** เป็นลบ **|x*x – a|** มีค่า **(x*x + |a|)**, วนไม่รู้จบ

ตัวอย่าง : วันใดของสัปดาห์

$$w = \left(d + \left\lfloor \frac{26(m+1)}{10} \right\rfloor + k + \left\lfloor \frac{k}{4} \right\rfloor + \left\lfloor \frac{c}{4} \right\rfloor + 5c \right) \bmod 7$$

$$k = y \bmod 100,\ c = \left\lfloor \frac{y}{100} \right\rfloor$$

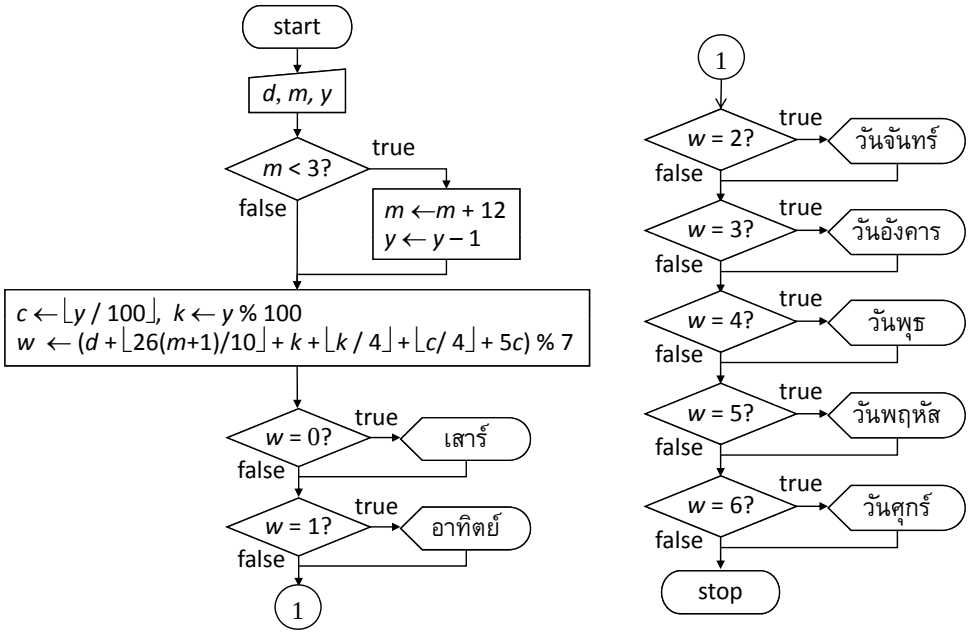
y คือเลขปี ค.ศ ส่วน m คือเลขเดือน = 3, 4, ..., 13, 14
แปลกเล็กน้อย m = 13 แทนมกราคม และ 14 แทนกุมภาพันธ์
ถ้าเป็นสองเดือนพิเศษนี้ให้ลดค่าปีลง 1 ด้วย

ตัวอย่าง : วันที่ 2 มิถุนายน ปี ค.ศ. 2009 เป็นวันอะไร ?

```
d = 2, m = 6, y = 2009
k = 9, c = 20
w = (2 + 18 + 9 + 2 + 5 + 100) % 7
  = 136 % 7
  = 3
```

0 – เสาร์, 1 – อาทิตย์, 2 – จันทร์, ..., 6 – ศุกร์
วันที่ 2 มิถุนายน ปี ค.ศ. 2009 คือ วันอังคาร

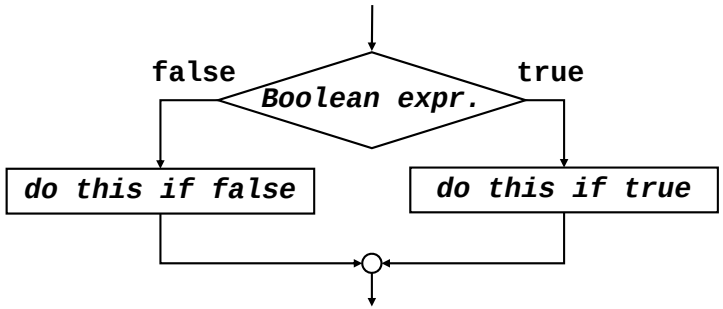
ผังงาน : วันใดของสัปดาห์



โปรแกรม : วันใดของสัปดาห์

```
Scanner kb = new Scanner(System.in);
System.out.print("วัน = ");
int d = kb.nextInt();
System.out.print("เลขเดือน = ");
int m = kb.nextInt();
System.out.print("ปี(ค.ศ.) = ");
int y = kb.nextInt();
if (m < 3) {
  m = m + 12;
  y = y - 1;
}
int c = y / 100, k = y % 100;
int w = (d + 26*(m+1)/10 + k + k/4 + c/4 + 5*c) % 7;
if (w == 0) { System.out.println("เสาร์");     }
if (w == 1) { System.out.println("อาทิตย์");   }
if (w == 2) { System.out.println("จันทร์");    }
if (w == 3) { System.out.println("อังคาร");    }
if (w == 4) { System.out.println("พุธ");       }
if (w == 5) { System.out.println("พฤหัสบดี");  }
if (w == 6) { System.out.println("ศุกร์");     }
```


if-else : จริงทำอย่าง เท็จทำอย่าง

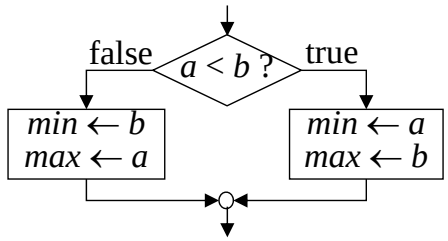



```
if ( นิพจน์ที่ได้ค่า จริง หรือ เท็จ ) {
   กลุ่มคำสั่งที่ทำงานเมื่อเงื่อนไขใน if เป็น จริง
} else {
  กลุ่มคำสั่งที่ทำงานเมื่อเงื่อนไขใน if เป็น เท็จ
}
```


ตัวอย่าง : ตัวน้อย ตัวมาก

- มีข้อมูลในตัวแปร **a** กับ **b**
- ต้องการให้ตัวแปร
  - **min** เก็บตัวน้อยของ **a** กับ **b**
  - **max** เก็บตัวมากของ **a** กับ **b**

```
min = a;
max = b;
if (a > b) {
   min = b;
   max = a;
}
```




```
if (a < b) {
   min = a;
   max = b;
} else {
   min = b;
   max = a;
}
```

ตัวอย่าง : หาค่าประมาณของ π ที่ดีกว่า 22/7

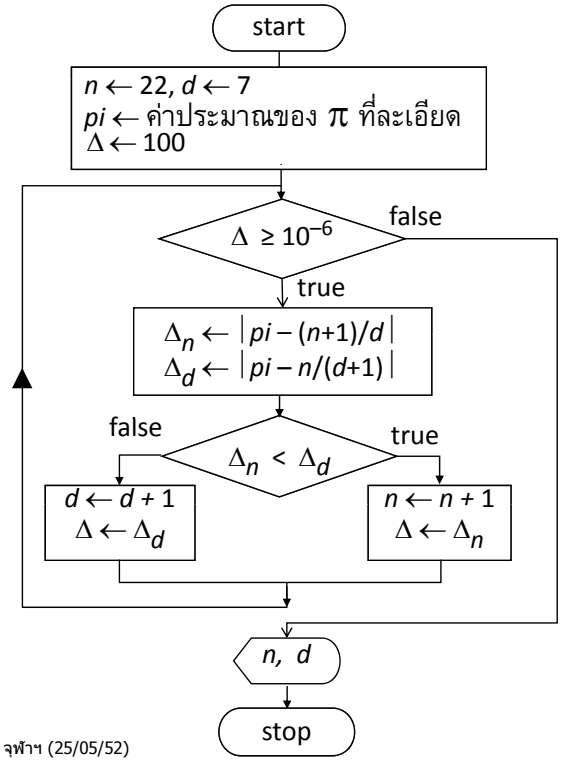
- $|\ \pi - 22/7\ | \approx 0.001264...$
- หาค่า n และ d ที่ทำให้ $|\ \pi - n/d\ | < 10^{-6}$

n/d	$\pi - (n+1)/d$	$\pi - n/(d+1)$
22/7	0.144	0.392
23/7	0.287	
23/8	0.142	
24/8	0.017	0.475
25/8	0.108	0.364
26/8	0.233	0.253
27/8	0.358	0.142
27/9	0.030	0.442
28/9	0.081	0.342
...	...	...
353/112	0.019	0.018
353/113	0.009	0.045
354/113	0.000	0.036

เลือกตัวน้อย

祖沖之
พบ $355/113$
เมื่อปี ค.ศ. 480

355/113



โปรแกรม :หาค่าประมาณของ π ที่ดีกว่า 22/7

```
public static void main(String[] args) {
   double n = 22, d = 7;
   double pi = Math.PI;  // 3.141592653589793;
   double diff = 100, diffN, diffD;
   while (diff >= 1e-6) {
     diffN = Math.abs(pi - (n + 1) / d);
     diffD = Math.abs(pi - n / (d + 1));
     if (diffN < diffD) {
       n = n + 1;
       diff = diffN;
     } else {
       d = d + 1;
       diff = diffD;
     }
   }
   System.out.println(n + "/" + d + " = " + (n/d));
}
```


กลุ่มคำสั่ง (Block)

- Block คือกลุ่มคำสั่ง ครอบด้วย { }
- ถ้ามีคำสั่งเดียวหลัง if, else, while ไม่ต้องใส่ใน block ก็ได้

```
if (diffN < diffD) {
  n = n + 1;
  diff = diffN;
} else {
  d = d + 1;
  diff = diffD;
}
```

```
if (x < 0) {
  x = -x;
}
```

```
if (x < 0)
  x = -x;
```

```
if (x < 0) x = -x;
```

```
if (a < b)
  min = a;
else
  min = b;
```

```
while (Math.abs(x*x – a) > 1e-5)
  x = (x + a/x) / 2.0;
```

ตัด { } หลัง if, else, while ต้องระวัง

```
Scanner kb = new Scanner(System.in);
System.out.print("วัน = ");
int d = kb.nextInt();
System.out.print("เลขเดือน = ");
int m = kb.nextInt();
System.out.print("ปี(ค.ศ.) = ");
int y = kb.nextInt();
if (m < 3) {
  m = m + 12;
  y = y - 1;
}
int c = y / 100, k = y % 100;
int w = (d + 26*(m+1)/10 + k + k/4 + c/4 - 2*c) % 7;
if (w == 0) System.out.println("เสาร์");
if (w == 1) System.out.println("อาทิตย์");
if (w == 2) System.out.println("จันทร์");
if (w == 3) System.out.println("อังคาร");
if (w == 4) System.out.println("พุธ");
if (w == 5) System.out.println("พฤหัสบดี")
if (w == 6) System.out.println("ศุกร์");
```

```
if (m < 3) m = m + 12; y = y - 1;
```