

Lecture 7 Condition , Selection Statement

9 กันยายน 2557 15:10

สเตทเมนต์ Statement เงื่อนไข หรือ ทางเลือก Decisions , Decisions

การเขียนโปรแกรมในบทก่อนหน้านี้ เราสามารถสร้าง การรับค่าอินพุต - input การคำนวณ processing การแสดงผล เอาต์พุต-output เราสามารถเก็บและแสดงผลผลลัพธ์ตัวแปร - variable และคำนวณทางคณิตศาสตร์เบื้องต้น

คำถามที่เกิดขึ้น โปรแกรมจำเป็นหรือไม่ที่จะทำงานเป็นลำดับเหมือนเดิมทุกครั้ง ซึ่งคงเป็นอะไรที่น่าเบื่อและไม่สามารถทำให้เกิดประโยชน์อะไรได้มาก ดังนั้นโปรแกรมจึงต้องการสิ่งที่เรียกว่าการตัดสินใจ หรือการเลือกทำอีกทางเลือก เราจะได้ศึกษาในบทนี้

Testing, Testing : การตรวจสอบ

โปรแกรมจำเป็นจะต้องทำงานได้หลากหลายรูปแบบขึ้นอยู่กับอินพุตที่ได้รับเข้ามา ดังตัวอย่างดังนี้

- If Tim got the right answer, add 1 point to his score
ถ้า Tim ตอบถูก จะเพิ่มหนึ่งคะแนนให้ Tim
- If Jane hit the alien, make an explosion sound.
ถ้า Jane ทำลายเอเลี่ยนได้ ให้เล่นเสียงระเบิด
- If the file isn't there, display an error message
ถ้าไฟล์ไม่มีอยู่ ให้แสดงผลลัพท์ข้อความ error

การตรวจสอบหรือตัดสินใจ จะทำหรือไม่ทำบางสิ่งบางอย่าง เราเรียกว่า condition หรือเงื่อนไข ที่มีค่าเป็นไปได้อาจจริงหรือเท็จ (true or false) ใช่หรือไม่ (yes or no) ซึ่ง Python มีรูปแบบการตรวจสอบข้อมูลเพื่อให้ทราบว่าเป็น True หรือ False ดังตัวอย่างแสดงต่อไปนี้

- ตรวจสอบว่าสองตัวแปรหรือค่าเท่ากันหรือไม่?
- ตรวจสอบสิ่งหนึ่งน้อยกว่าอีกสิ่งหรือไม่?
- ตรวจสอบสิ่งหนึ่งมากกว่าอีกสิ่งหรือไม่?

เราจำเป็นจะต้องแปลงค่าพูลของเราให้คอมพิวเตอร์เข้าใจมากยิ่งขึ้น ถ้าเราบอกว่า Tim ตอบถูกหรือไม่ เราจะเปลี่ยนเป็น Tim ตอบตรงคำตอบหรือไม่ เช่น

If Tim got the right answer

เป็น If Tim's Answer is equal to Correct Answer

ถ้าเขาตอบถูกแสดงว่าคำตอบของเขาตรงกับคำตอบที่ต้องการ

อีกคำสั่งหนึ่งที่ใช้กับ condition คือ เมื่อเราตัดสินใจได้ว่าจริงหรือไม่ หลังจากนั้นทิศทางการเดินของโปรแกรมจะต้องไปสองทางไม่ซ้ำหรือขวา ไม่จริงก็เท็จ เหมือนกับกิ่งไม้ที่แยกจากกัน ดังนั้นคำศัพท์ที่ใช้ในเรื่องนี้อีกคำคือคำว่า branching การแตกกิ่ง



ตัวอย่างต่อไปนี้เป็นการใช้เงื่อนไขในภาษา Python โดยใช้คำสั่ง if

```
if timesAnswer == correctAnswer:  
→ print("You got it right!")  
    score = score + 1  
print("Thanks for playing.")
```

สองบรรทัดนี้จะต้องกดแท็บเข้ามาหนึ่งครั้ง
ระวังถ้าหากไม่ใช้แท็บ tab จริงโปรแกรมอาจจะ error
การแท็บเข้ามานี้เรียกว่า indented

ส่วนที่ถูกแท็บเข้ามานี้เป็นเนื้อหาส่วนที่อยู่ภายใต้ if เราจะเรียกว่า block คือกลุ่มของคำสั่งตั้งแต่หนึ่งคำสั่งขึ้นไป
ที่ทำงานพร้อมกันเป็นกลุ่ม ซึ่งในภาษา Python คือ ส่วนที่อยู่ในระดับแถบเดียวกันคือ Block เดียวกัน
เราจะสังเกตเห็นเครื่องหมายโคลอน (: colon) ตัวอักษรสุดท้ายของบรรทัด if โคลอนแสดงถึงการเริ่มของ คำ
สั่ง block เริ่มต้น ซึ่งบรรทัดที่อยู่ภายใต้ block ด้วยการแถบเข้าไป และสิ้นสุดหรือจบ block เมื่อ บรรทัดไม่มีการ

แทนเข้าไป ดังตัวอย่างสิ้นสุดที่คำสั่ง `print("Thanks for... ซึ่งไม่ได้อยู่ภายใต้ block`

การทำงานของ `if` ถ้าเงื่อนไขเป็นจริง (คำตอบของ Tim เท่ากับคำตอบที่ถูกต้อง) คำสั่งที่อยู่ใน Block ภายใต้ `if` จะเริ่มทำงาน

ภาษาโปรแกรมมิ่งโดยส่วนใหญ่ จะใช้เครื่องหมายปีกกา `{}` แสดงพื้นที่ของ block และการแทนเพื่อให้โค้ดสวยงาม (coding style) เช่น ภาษา `c/c++` `java`

```
if (timeAnswer == correctAnswer){
    score = score + 1;
}
```

แต่ Python จะใช้ `(:)` และการแทน indenting เป็นการแสดง พื้นที่ภายใน Block แทน

การเปรียบเทียบโดย Python จะใช้เครื่องหมายเท่ากับสองตัวติดกัน `(==)`

เนื่องจากเราได้ใช้เครื่องหมาย `(=)` เพื่อแสดงการให้ค่า (setting) หรือภาษาโปรแกรมเรียกว่า assigning การให้ค่าตัวอย่างเช่น

```
correctAnswer = 5 + 3
temperature = 35
name = "Sarayut"
```

ซึ่งหมายถึงการให้ค่าแก่ตัวแปร -variable ต่างๆ

แต่เราต้องตรวจสอบ หรือ เปรียบเทียบ ซึ่งอาจจะเหมือนการถามคำถาม ว่าเท่ากันหรือไม่เราจะใช้ `(==)`

```
if myAnswer == correctAnswer:
if temperature == 40:
if name == "Sarayut":
```

โดยส่วนใหญ่แล้วคนเขียนโปรแกรมจะซับซ้อนกับการใช้ จะชอบใช้ `=` ในการเปรียบเทียบแทนที่จะเป็น `==` และภาษาการเขียนโปรแกรมส่วนใหญ่ก็ใช้การเปรียบเทียบเหมือนกับภาษา Python

การเปรียบเทียบเราจะเรียกว่า comparing เครื่องหมาย `==` ก็เป็นส่วนหนึ่งของเครื่องหมายเปรียบเทียบ - comparison operator เครื่องหมาย - operator

การเปรียบเทียบแบบอื่นๆ

- Less than `<` น้อยกว่า
- Greater than `>` มากกว่า
- Not equal `!=` ไม่เท่ากับ
- Less than or equal `<=` น้อยกว่า
- Greater than or equal `>=` มากกว่า

เราสามารถเปรียบเทียบระหว่างค่าเช่น อายุมากกว่า 8 น้อยกว่า 12 เราสามารถเขียนดังนี้

```
if 8 < age < 12:
```

อายุเท่าไรที่เป็นจริงบ้าง 9 , 10 และ 11 เนื่องจากเราไม่ได้รวม 8 , และ 12 ถ้าต้องการรวมด้วยจะต้องใช้เท่ากับมาช่วย

```
if 8 <= age <= 12:
```

Comparison - การเปรียบเทียบ

Comparison operator อาจจะเรียกว่า Relational Operators Relational แปลว่า เชิงสัมพันธ์ ดูระหว่างสองค่าที่เปรียบเทียบมีความสัมพันธ์กันแบบไหน

Comparison test การตรวจสอบด้วยการเทียบค่า หรืออาจจะเป็นการเปรียบเทียบทางตรรกะ logical test ตรรกะ logical คือสิ่งที่จะให้คำตอบได้เพียงจริงหรือเท็จอย่างใดอย่างหนึ่งเท่านั้น

จงทดสอบเขียนโปรแกรมเพื่อเปรียบเทียบค่าตัวเลขสองตัว โดยการรับค่าตัวแปรมาสองตัว
Comparing.py

```

File Edit Format Run Options Windows Help
num1 = float(input("Enter num1:"))
num2 = float(input("Enter num2:"))
if num1 < num2:
    print(num1, "is less than", num2)
if num1 > num2:
    print(num1, "is greater than", num2)
if num1 == num2:
    print(num1, "is equal", num2)
if num1 != num2:
    print(num1, "is not equal", num2)

```

ทดสอบและแสดงผลลัพท์

```

Enter num1:5
Enter num2:3
5.0 is greater than 3.0
5.0 is not equal 3.0

```

เกิดอะไรขึ้นถ้าการตรวจสอบพบว่ามีค่าเป็น False

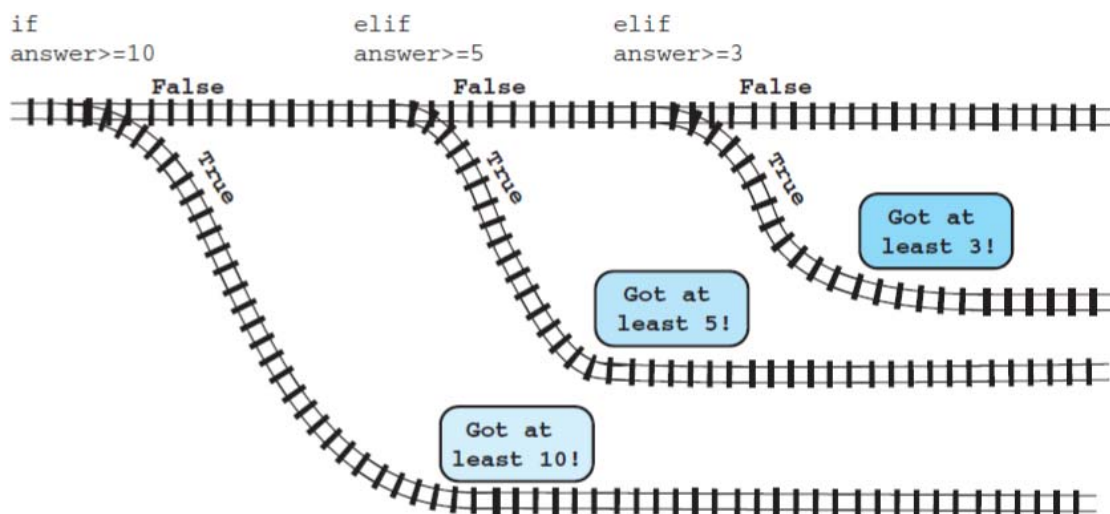
- ทำการทดสอบคำสั่งอื่น ถ้าเจอว่าเป็นเท็จ python จะทดสอบสิ่งอื่น (else) ที่มีคำว่า elif (ย่อมาจาก else if) อย่างเช่น

```

if answer >= 10:
    print("You got at least 10!!")
elif answer >= 5:
    print("You got at least 5!!")
elif answer >= 3:
    print("You got at least 3!!")

```

เราจะใช้ elif หลังจากใช้ if

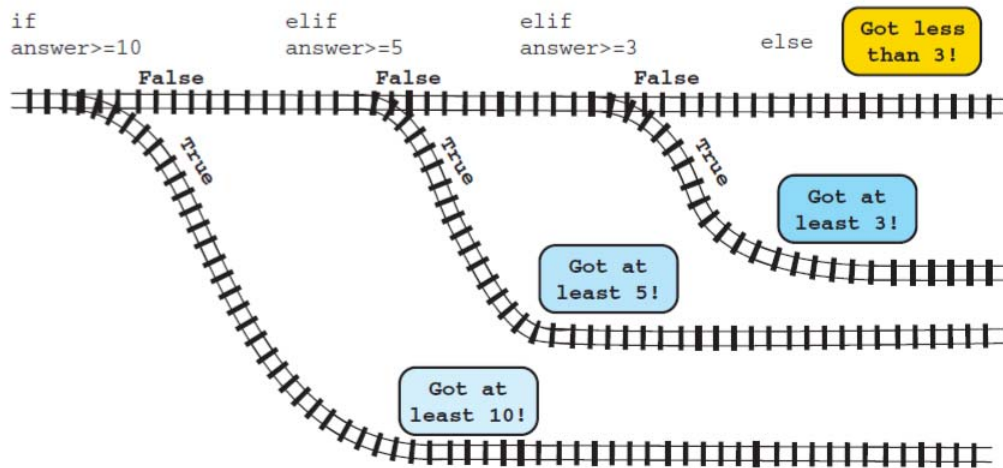


- ทำคำสั่งอื่น คำสั่งก่อนหน้านี้ที่ if หรือ elif เป็นเท็จ เราจะทำที่คำสั่งอื่น (else) คือ ใช้คำสั่ง else:

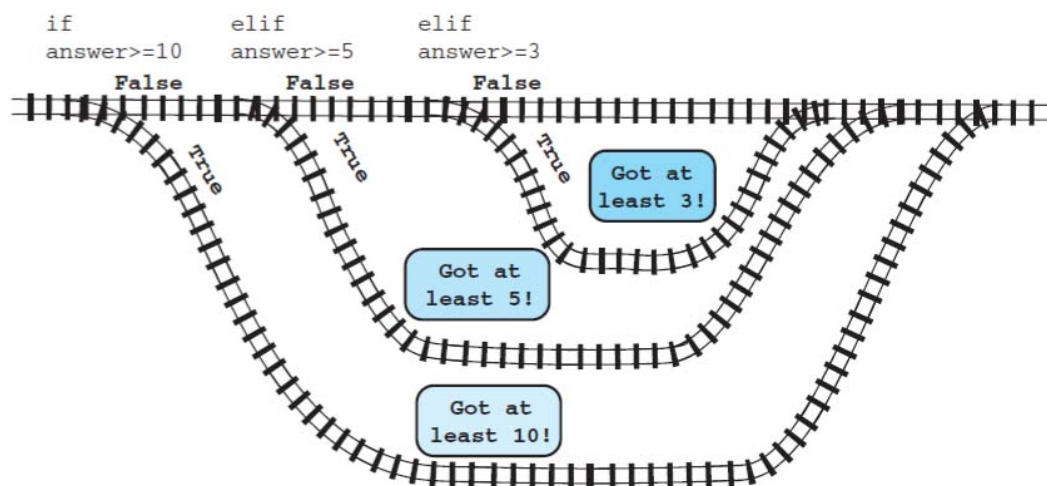
```

if answer >= 10:
    print("You got at least 10!!")
elif answer >= 5:
    print("You got at least 5!!")
elif answer >= 3:
    print("You got at least 3!!")
else:
    print("You got least than 3!!")

```



- ไปต่อที่คำสั่งอื่น ถ้าเราไม่ได้ใส่คำสั่ง else หลัง if หรือ elif เมื่อทดสอบแล้วเป็นเท็จทุกกรณี โปรแกรมจะเลื่อนไปยังโค้ดที่ต่อจากโค้ด if



ลองเขียนโปรแกรมทดสอบโค้ดด้านบนใส่คำสั่ง อ่านค่าอินพุต answer ไว้ก่อนใช้ if

```
answer = float(input("Enter answer:"))
```

Testing for more than one condition : ตรวจสอบมากกว่า หนึ่งเงื่อนไข

เราต้องการเขียนโปรแกรมกำหนดอายุ (age) และ ผลการเรียนของเด็ก (grade) คนที่มีอายุมากกว่าหรือเท่ากับ 8 และผลการเรียนมากกว่าหรือเท่ากับเกรด 3 ถึงจะให้เล่นเกมส์

```
age = float(input("Enter your age: "))
grade = float(input("Enter your grade: "))
if age >= 8:
    if grade >= 3:
        print("You can play this game.")
else:
    print("Sorry , you can't play the game");
```

สังเกต else จะอยู่ระดับเดียวกันกับ if , block ของ if ข้อมูลจะโดนแทน เช่นเดียวกับ else แสดงผลลัพธ์จากโค้ด

```

Enter your age: 12
Enter your grade: 3
You can play this game.
>>> =====
>>>
Enter your age: 5
Enter your grade: 3
Sorry , you can't play the game
>>> =====
>>>
Enter your age: 12
Enter your grade: 2

```

Using and : การใช้เงื่อนไข and - และ

จากตัวอย่างที่แล้วเราสามารถรวมคำสั่ง if เข้าด้วยกัน ด้วยการใช้ and

```

age = float(input("Enter your age: "))
grade = float(input("Enter your grade: "))
if age >= 8 and grade >= 3:
    print("You can play this game.")
else:
    print("Sorry , you can't play the game");

```

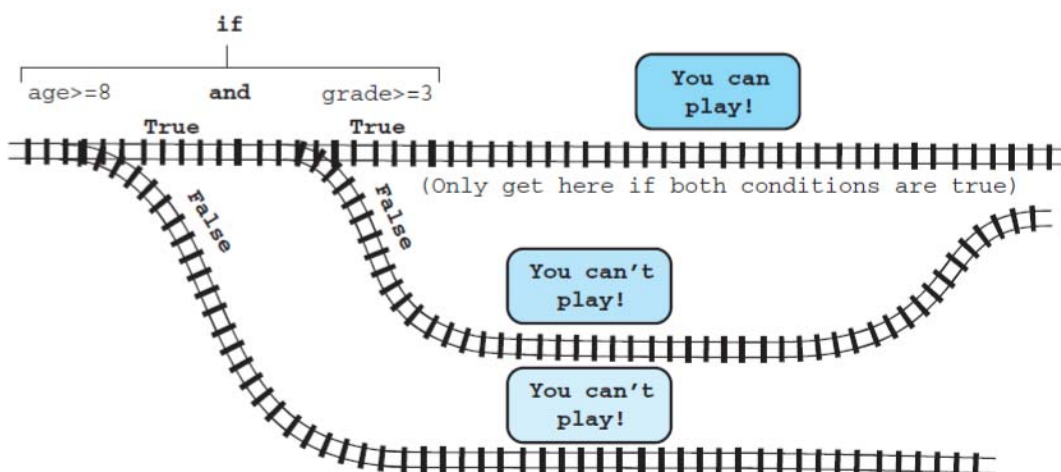
แสดงผลลัพธ์ของโปรแกรม

```

Enter your age: 12
Enter your grade: 2
>>> =====
>>>
Enter your age: 12
Enter your grade: 3
You can play this game.
>>> =====
>>>
Enter your age: 5
Enter your grade: 3
Sorry , you can't play the game
>>> =====
>>>
Enter your age: 12
Enter your grade: 2
Sorry , you can't play the game

```

การหาคำตอบ



เราสามารถใส่ and มากกว่า 1 ใช้เชื่อมเงื่อนไขมากกว่าสอง เช่นเราถาม สีด้วย

```

age = float(input("Enter your age: "))
grade = float(input("Enter your grade: "))
color = input("Enter your color: ");
if age >= 8 and grade >= 3 and color == "green":
    print("You can play this game.")
else:
    print("Sorry , you can't play the game");

```

แสดงผลการลัพท์

```

Enter your age: 12
Enter your grade: 3
Enter your color: green
You can play this game.
>>> =====
>>>
Enter your age: 12
Enter your grade: 3
Enter your color: red
Sorry , you can't play the game

```

ถ้าเรามีเงื่อนไขตั้งแต่ 2 ขึ้นไปเงื่อนไขจะเป็นจริงได้ ทุกเงื่อนไขต้องเป็นจริง true นอกจากนี้ and ยังมีการเชื่อมแบบอื่นอีก

Using Or :การใช้ or หรือ

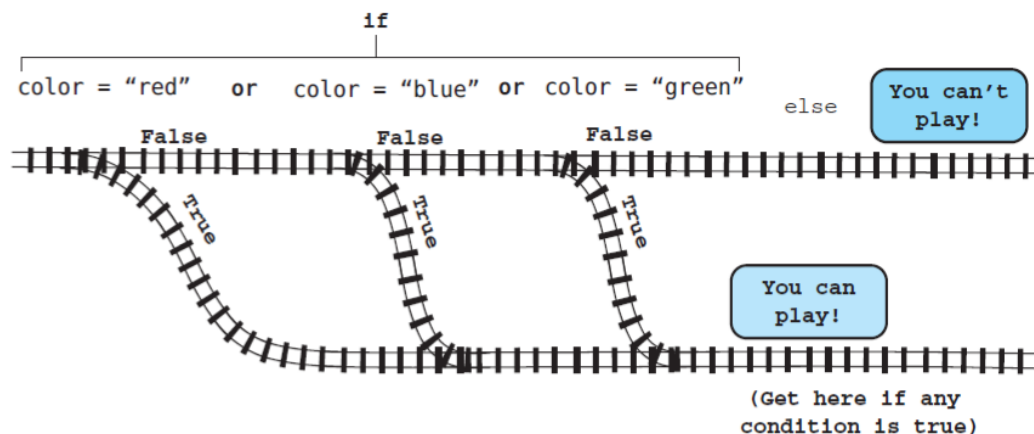
การใช้หรือเราไม่จำเป็นจะต้องเป็นจริงทั้งสองอย่าง เราสามารถทำได้เมื่อมีอย่างใดอย่างหนึ่งเป็นจริง

```

color = input("Enter your color: ");
if color == "red" or color == "blue" or color == "green":
    print("You can play this game.")
else:
    print("Sorry , you can't play the game");

```

การหาคำตอบ



Using Not: การใช้ not (การกลับค่า)

```

age = float(input("Enter your age: "))
if not (age < 8) :
    print("You can play this game.")
else:
    print("Sorry , you can't play the game");

```

จากโค้ดด้านบนเป็นการไม่รับเด็กอายุต่ำกว่า 8

บรรทัดนี้ if not (age < 8):

มีค่าเท่ากับ if age >= 8:

สรุปตั้งแบบบทที่ 4 เครื่องหมายต่างๆ ที่เราใช้

เครื่องหมายในการคำนวณ (Math Operators)

=	Assignment	การให้ค่า
+	Addition	การบวกค่าระหว่างสองตัวเลข

=	Assignment	
+	Addition	การบวกค่าระหว่างสองตัวเลข
-	Subtraction	การลบค่าระหว่างสองตัวเลข
+=	Increment	การเพิ่มค่าตัวเลข
-=	Decrement	การลดค่าตัวเลข
*	Multiplication	การคูณ
/	Division	การหาร
%	Modulus	การหารเอาเศษ
**	Exponentiation	ยกกำลัง

เครื่องหมายเปรียบเทียบ (Comparison Operators)

==	Equality	เปรียบเทียบสองสิ่งเท่ากันหรือไม่
<	Less than	เปรียบเทียบค่าแรกน้อยกว่าค่าสองหรือไม่
>	Greater than	เปรียบเทียบค่าแรกมากกว่าค่าสองหรือไม่
<=	Less than or equal to	เปรียบเทียบค่าแรกน้อยกว่าหรือเท่ากับค่าสองหรือไม่
>=	Greater than or equal to	เปรียบเทียบค่าแรกมากกว่าหรือเท่ากับค่าสองหรือไม่
!=	Not equal to	เปรียบเทียบสองสิ่งไม่เท่ากันหรือไม่
<>		

อาจจะเก็บหน้านี้ไว้ใช้ในการอ้างอิงเวลาเขียนโปรแกรม

What did you learn?

บททวน

- การทดสอบเปรียบเทียบและการใช้เครื่องหมายเปรียบเทียบ
- Block และ การแทน indenting
- การใช้ and และ or เชื่อมเพื่อการทดสอบ
- การกลับโดยใช้เครื่องหมาย not

Test your knowledge

1. ผลลัพธ์ที่ได้จากโค้ดด้านล่างนี้คือ

```
my_number = 7
if my_number < 20:
    print("Under 20")
else:
    print("20 or Over")
```
2. จากข้อหนึ่งถ้า เปลี่ยนตัวเลข my_number เป็น 25 จะได้ผลลัพธ์อะไร
3. เราจะใช้ if อย่างไรในการตรวจสอบตัวเลขว่ามีค่ามากกว่า 30 แต่ น้อยกว่า 40?
4. จะตรวจสอบอย่างไร ผู้ใช้งานป้อนตัวอักษร Q ใหญ่หรือ q เล็ก

Try it Out จงเขียนโปรแกรมต่อไปนี้

1. เราจะเขียนโปรแกรมให้ร้านค้าคิดเงินอย่าง ถ้าเราคาสินค้าไม่เกิน 1000 บาท ลดให้ 10% แต่ถ้ามากกว่าลดให้ 20% เขียนโปรแกรมถามราคาซื้อ (purchase price) แสดงส่วนลด (10 % หรือ 20%) และแสดงราคาสุดท้ายที่จ่ายจริง

```
Enter purchase price: 500
Discount 10%
Discount amount = 50.0
Final price = 450.0
```

```
>>> =====
>>>
```

```
Enter purchase price: 2000
Discount 20%
Discount amount = 400.0
Final price = 1600.0
```

2. จงเขียนโปรแกรมเพื่อคัดเลือักนักเขียนรหัสเตอร์อายุ 18 - 22 ปี ที่เป็นผู้หญิง ให้เขียนโปรแกรมถามเพศว่าเป็นชายหรือหญิง ให้ตอบเป็นตัวอักษร m ผู้ชาย(male) f ผู้หญิง(female) ถามอายุ แล้วตอบว่ามีสิทธิ์เข้าร่วมหรือไม่ (Bonus: ถ้าถามเพศแล้วเป็นผู้ชาย แล้วไม่ต้องถามอายุต่อ)

```
Enter your gender m (male) or f(female):m
You are not eligible.
```

```
>>> ===== RESTART ==
>>>
```

```
Enter your gender m (male) or f(female):f
Enter your old:12
You are not eligible.
```

```
>>> ===== RESTART ==
>>>
```

```
Enter your gender m (male) or f(female):f
Enter your old:25
You are not eligible.
```

```
>>> ===== RESTART ==
>>>
```

```
Enter your gender m (male) or f(female):f
Enter your old:20
You are eligible.
```

3. จงเขียนโปรแกรมให้คำนวณการเติมน้ำมัน ถ้าปั้มน้ำมันข้างหน้าห่าง 200 km (next gas station) โดยให้ป้อนข้อมูลต่อไปนี้

- a. ถังน้ำมันมีความจุกี่ลิตร (Size of tank)
- b. เปอร์เซ็นต์น้ำมันที่เหลืออยู่ในถังตอนนี้ (percent full)
- c. อัตราการวิ่งของรถได้กิโลต่อลิตร (km per liter)

ให้โปรแกรมแสดงผลว่ารถวิ่งได้อีกกี่กิโล (you can go another)
และควรเติมน้ำมันที่ปั้มน้ำหรือไม

```
Size of tank: 60
percent full: 40
km per liter: 10
You can go another 240 km
The next gas station is 200 km away
You can wait for the next station.
```

or

```
Size of tank: 60
percent full: 30
km per liter: 8
You can go another 144 km
The next gas station is 200 km away
Get gas now!
```