



PYTHON

การคำนวณทาง

คณิตศาสตร์

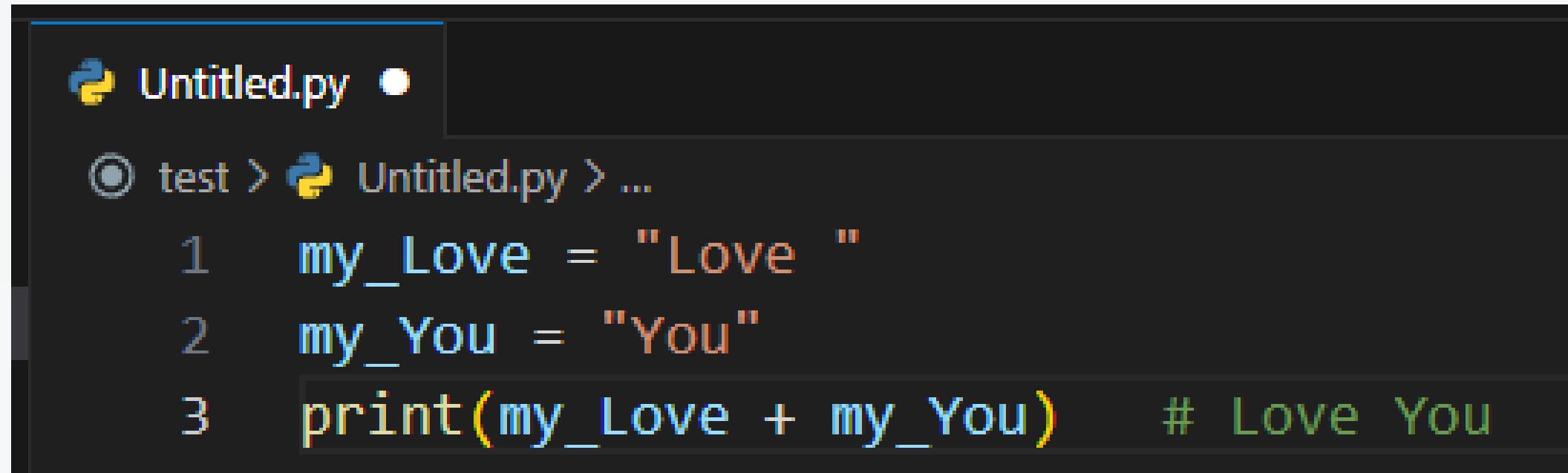
PART 1

- 01** รู้จักกับ PYTHON
- 02** ติดตั้ง PYTHON และการใช้งาน PYTHON SHELL
- 03** ตัวแปรและชนิดข้อมูล
- 04** การคำนวณค่าทางคณิตศาสตร์
- 05** พังก์ชัน
- 06** การตัดสินใจ
- 07** การวนซ้ำ

การคำนวณค่าทางคณิตศาสตร์

นิพจน์ (Expression)

- คือ การนำข้อมูลมากรอกำร่วมกัน โดยใช้เครื่องหมายหรือคำสั่งพิเศษรูปแบบต่าง ๆ
 - เช่น
 - $8 + 10$ คือ นิพจน์ สำหรับนำ **8** ไปรวมกับ **10** มีผลลัพธ์นิพจน์มีค่าเป็น **18**
 - “**Love**” + “**you**” คือ นิพจน์คือ **“Love”** ต่อกับ **“you”** คือ **“Love you”**
 - my_Number() * 2** คือ การเรียกฟังก์ชัน **my_Number()** มาคูณด้วย **2**



```
Untitled.py •
◎ test > Python Untitled.py > ...
1 my_Love = "Love "
2 my_You = "You"
3 print(my_Love + my_You) # Love You
```

บทที่ 7 ประมวลไปด้วย 2 ส่วน

โอเปอเรเตอร์ (Operator)

- โอเปอเรเตอร์ (Operator : ตัวดำเนินการ) คือ เครื่องหมาย สัญลักษณ์ หรือคำสั่ง ที่ใช้กำหนดค่า แก้ไข หรือกระทำกับค่าด้วยรูปแบบต่าง ๆ ดังนี้
 - Assignment Operators ใช้ในการกำหนดค่าให้กับตัวแปร เช่น
 - = , += , -= , *= , /=
 - Arithmetic Operators ใช้ในการคำนวณค่าทางคณิตศาสตร์ เช่น
 - + , - , * , /
 - Comparison Operators ใช้ในการเปรียบเทียบค่า เช่น
 - == , != , < , > , <= , >=
 - Logical Operators ใช้ในการหาค่าความจริงทางตรรกะ เช่น
 - or , and , not

บทที่ ๘ ประมวลไปด้วย 2 ส่วน

โอเปอเรเตอร์ (Operator)

- โอเปอเรเตอร์ (Operator : ตัวดำเนินการ) คือ เครื่องหมาย สัญลักษณ์ หรือคำสั่ง ที่ใช้กำหนดค่า แก้ไข หรือกระทำกับค่าด้วยรูปแบบต่าง ๆ ดังนี้
 - Bitwise Operators ใช้ในการแก้ไขหรือเปลี่ยนแปลงค่าในระดับบิต เช่น
 - | , & , ^ , >> , <<
 - Membership Operators ใช้ในการตรวจสอบตัวแปรหรือที่สนใจ ว่าเป็นชุดข้อมูลที่ น่าใจหรือไม่ เช่น
 - in , not in
 - Identity Operators ใช้ในการตรวจสอบว่าสองเจกต์หรือที่สนใจ ว่าเป็นชุดข้อมูลที่ น่าใจหรือไม่ เช่น
 - is , is not

尼พจน์ประกอบໄປດ້ວຍ 2 ส່ວນ

ໂອເປົອແຣນດໍ (Operand)

- ໂອເປົອແຣນດໍ (Operand : ຕັວຄູກດຳເນີນການ) ຄື່ອ ຂ້ອມູລທີ່ຄູກໂອເປົອເຮັດວຽກ (Operator) ນຳຄ່າໄປໃຊ້ງານ ເຊັ່ນ
 - $7 + 10$
 - ນີ້ພຈນີ້ກີ່ອກາຮາພລບວກສໍາຮັບຕົວເລຂ 7 ແລະ 10
 - ໃນກຮນີ້ ເຄຣື່ອງໝາຍ + ຄື່ອ ໂອເປົອເຮັດວຽກ (Operator)
 - 7 ແລະ 10 ຄື່ອ ໂອເປົອແຣນດໍ (Operand)

```
Untitled.py ×
test > Untitled.py > ...
1 my_num1 = 7
2 my_num2 = 10
3 print(my_num1 + my_num2) # 7 + 10 = 17
```

ตัวดำเนินการทางคณิตฯ (Arithmetic Operators)

Arithmetic Operators

- คือ โอเปอเรเตอร์ ที่ใช้หาผลลัพธ์จากการคำนวณทางคณิตศาสตร์ ซึ่ง Python จะมีตัวดำเนินการทางคณิตฯ อยู่หลายชนิด ดังนี้



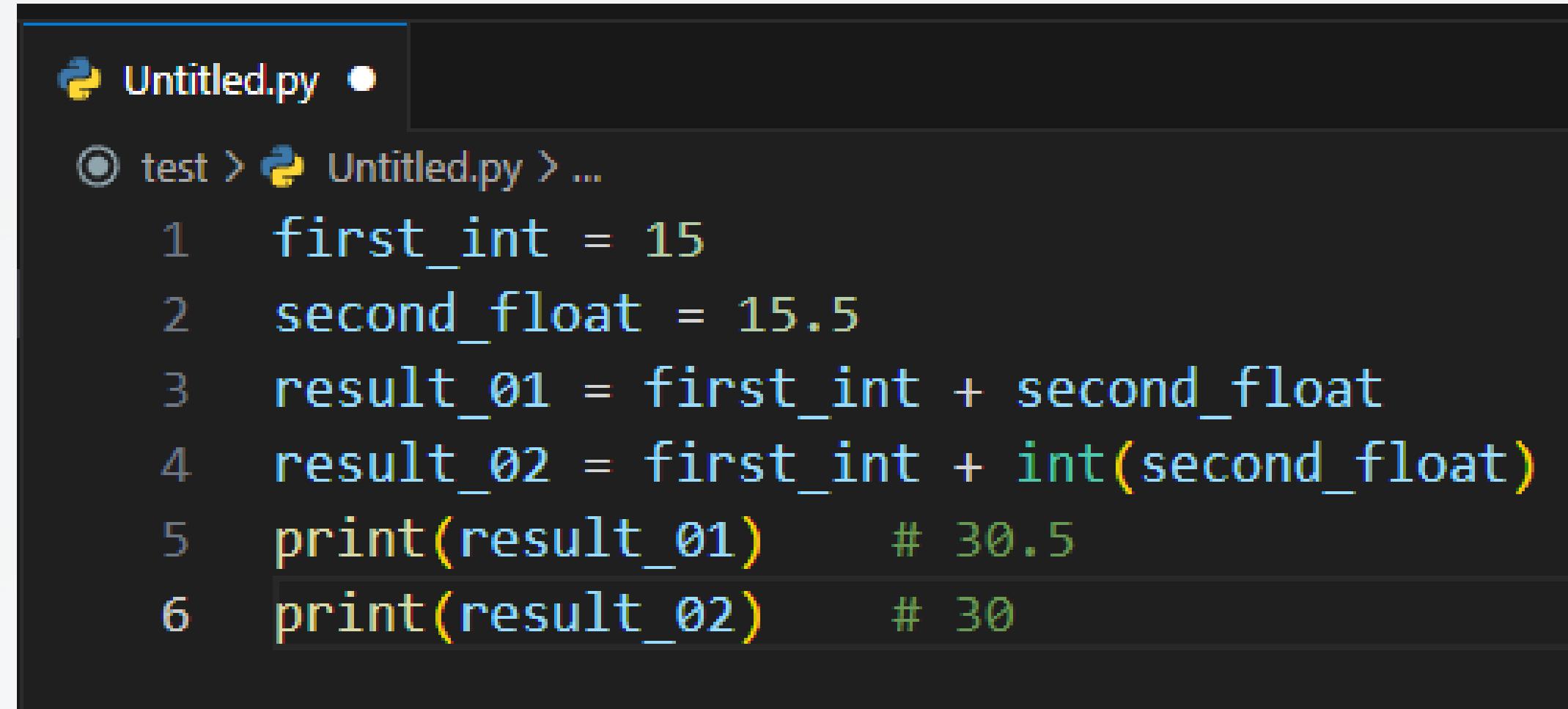
โอเปอเรเตอร์	จุดประสงค์	ตัวอย่าง
+	บวก	$10 + 9$
-	ลบ	$20 - 5$
*	คูณ	$30 * 2$
/	หาร	$9 / 3$
**	ยกกำลัง	$5 ** 2$
//	หารไม่เอาเศษ	$10 // 4$
%	หารเอาเศษ	$10 \% 4$

การใช้เครื่องหมายต่าง ๆ

เครื่องหมายทางคณิตฯ

- เครื่องหมายบวก (+) เรียกว่า **Additional Operator**
 - เครื่องหมายลบ (-) เรียกว่า **Subtraction Operator**
 - เครื่องหมายคูณ (*) เรียกว่า **Multiplication Operator**
 - เครื่องหมายหาร (/) เรียกว่า **Division Operator**
-
- หากนำตัวเลขหรือตัวแปรมา **kratgarwm** กับกี่มีชนิดข้อมูลเป็นจำนวนเต็ม (int) จะได้ผลลัพธ์เป็นจำนวนเต็มเสมอ (int)
 - แต่ถ้านำตัวเลขจำนวนเต็ม (int) **kratgarwm** กับตัวเลขทศนิยม (float) จะได้ผลลัพธ์เป็นจำนวนบทศนิยม (float)

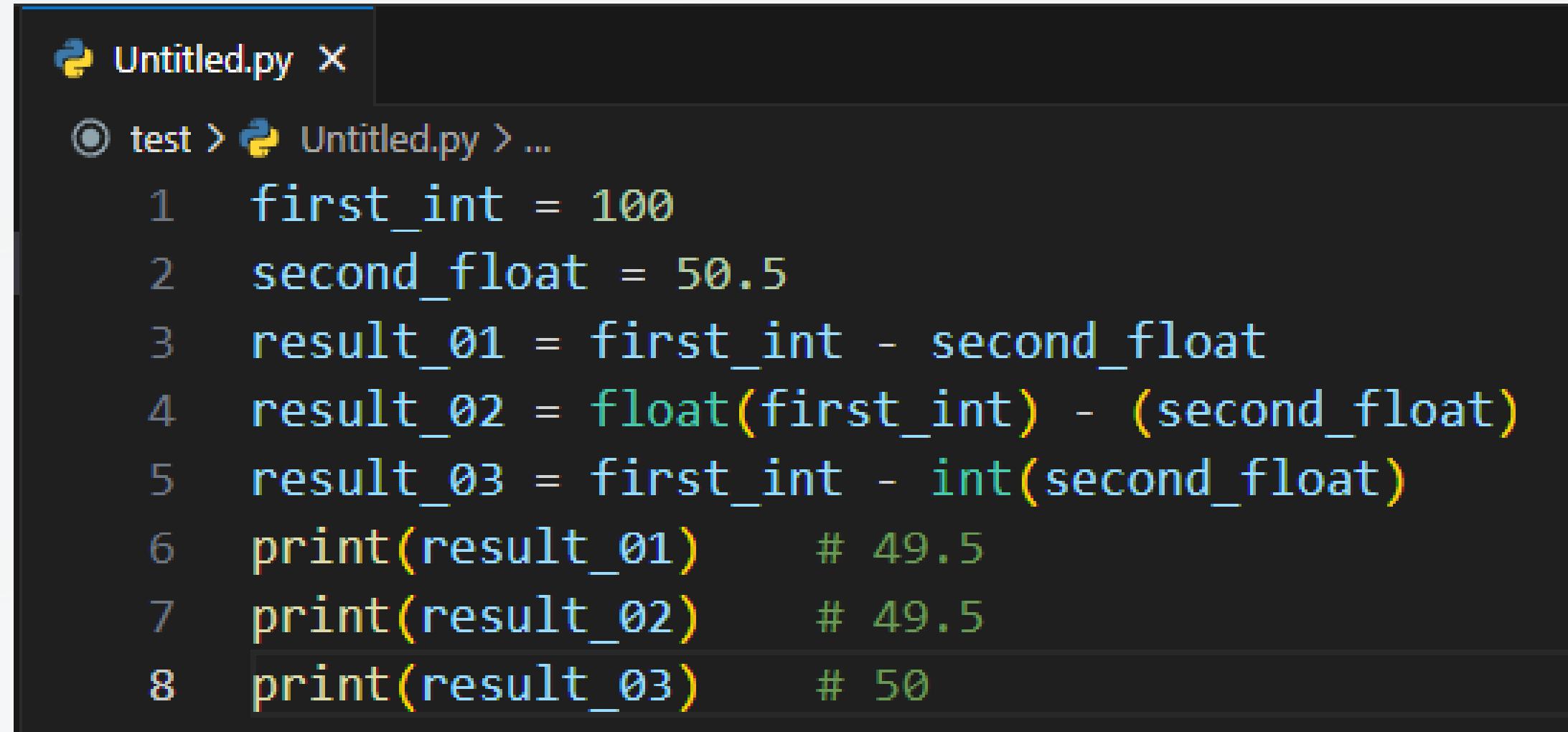
จากตัวอย่าง การบวก (+) จะดูตัวแปรต่างกัน



```
Untitled.py
test > Untitled.py > ...
1 first_int = 15
2 second_float = 15.5
3 result_01 = first_int + second_float
4 result_02 = first_int + int(second_float)
5 print(result_01)      # 30.5
6 print(result_02)      # 30
```

- **บรรทัดที่ 1-2** ประกาศตัวแปรจำนวนเต็ม (int) และ ตัวเลขทศนิยม (float)
- **บรรทัดที่ 3-4** ประกาศตัวแปรเพื่อเก็บผลลัพธ์ของการบวก (+)
- **บรรทัดที่ 5** print แสดงค่า result_01 ผลลัพธ์คือ **int + float = float**
- **บรรทัดที่ 6** print แสดงค่า result_02 ผลลัพธ์คือ **int + int = int**

ຈາກຕັວອຍ່າງ ກາຣສບ (-) ຂົດຕັວແປຣຕ່າງກັນ



```
Untitled.py X
test > Untitled.py > ...
1 first_int = 100
2 second_float = 50.5
3 result_01 = first_int - second_float
4 result_02 = float(first_int) - (second_float)
5 result_03 = first_int - int(second_float)
6 print(result_01)      # 49.5
7 print(result_02)      # 49.5
8 print(result_03)      # 50
```

- **ចັດກຳ 1-2** ປະກາສຕັວແປຣຈຳນວນເຕີມ (int) ແລະ ຕັວເລຂທຄນິຍມ (float)
- **ចັດກຳ 3-5** ປະກາສຕັວແປຣເພື່ອເກີບຜລັພຣຂອງກາຣສບ (-)
- **ចັດກຳ 6** print ແສດງຄ່າ result_01 ຜລັພຣດີ້ວ່າ **int - float = float**
- **ចັດກຳ 7** print ແສດງຄ່າ result_02 ຜລັພຣດີ້ວ່າ **float - float = float**
- **ចັດກຳ 8** print ແສດງຄ່າ result_03 ຜລັພຣດີ້ວ່າ **int - int = int**

จากตัวอย่าง การคูณ (*) ชนิดตัวแปรต่างกัน

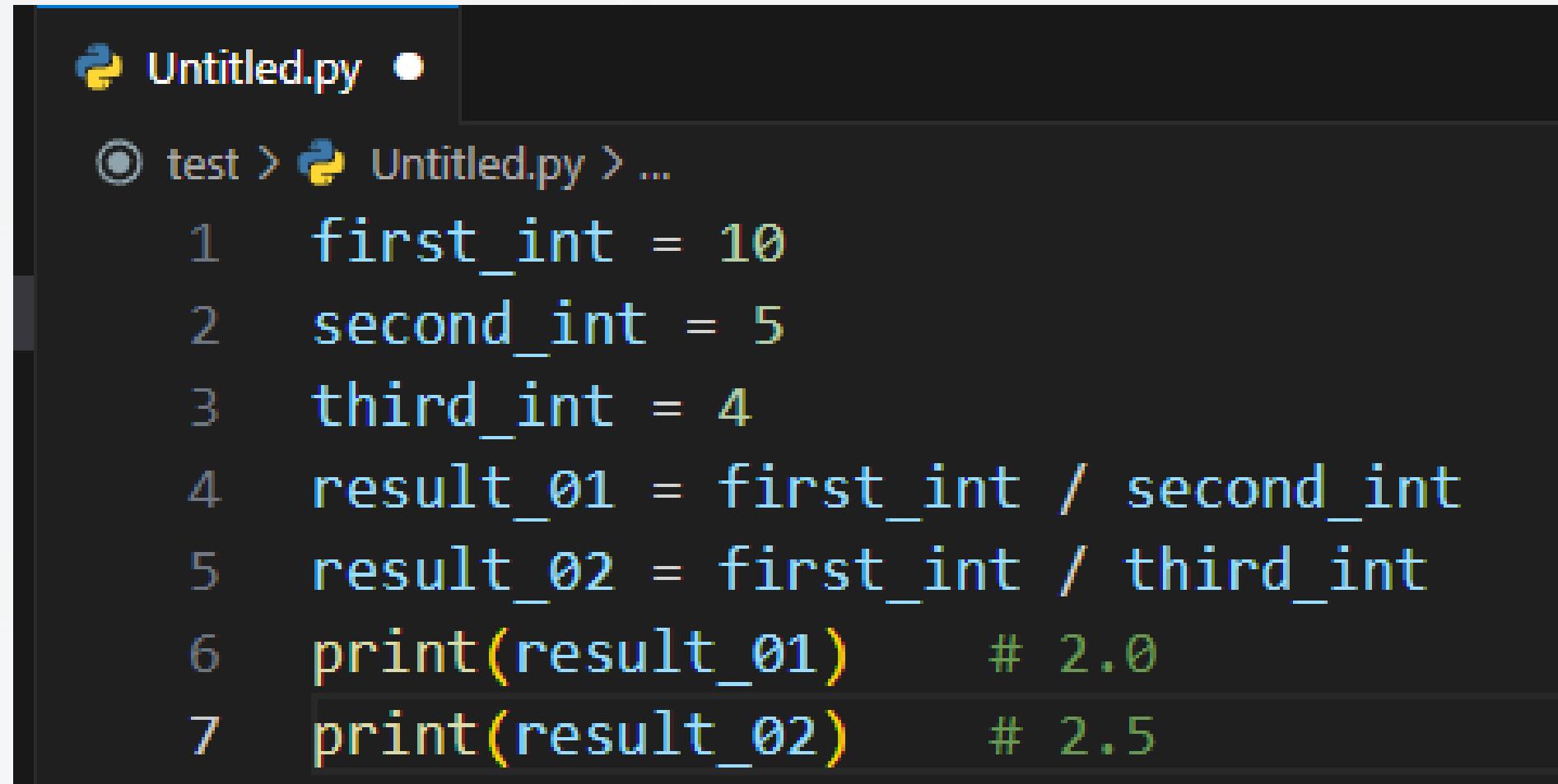
```
Untitled.py ×

test > Untitled.py > ...

1 first_int = 10
2 second_float = 5.5
3 result_01 = first_int * second_float
4 result_02 = float(first_int) * (second_float)
5 result_03 = first_int * int(second_float)
6 print(result_01)      # 55.0
7 print(result_02)      # 55.0
8 print(result_03)      # 50
```

- **บรรทัดที่ 1-2** ประกาศตัวแปรจำนวนเต็ม (int) และ ตัวเลขทศนิยม (float)
- **บรรทัดที่ 3-5** ประกาศตัวแปรเพื่อเก็บผลลัพธ์ของการคูณ (*)
- **บรรทัดที่ 6** print แสดงค่า result_01 ผลลัพธ์คือ **int * float = float**
- **บรรทัดที่ 7** print แสดงค่า result_02 ผลลัพธ์คือ **float * float = float**
- **บรรทัดที่ 8** print แสดงค่า result_03 ผลลัพธ์คือ **int * int = int**

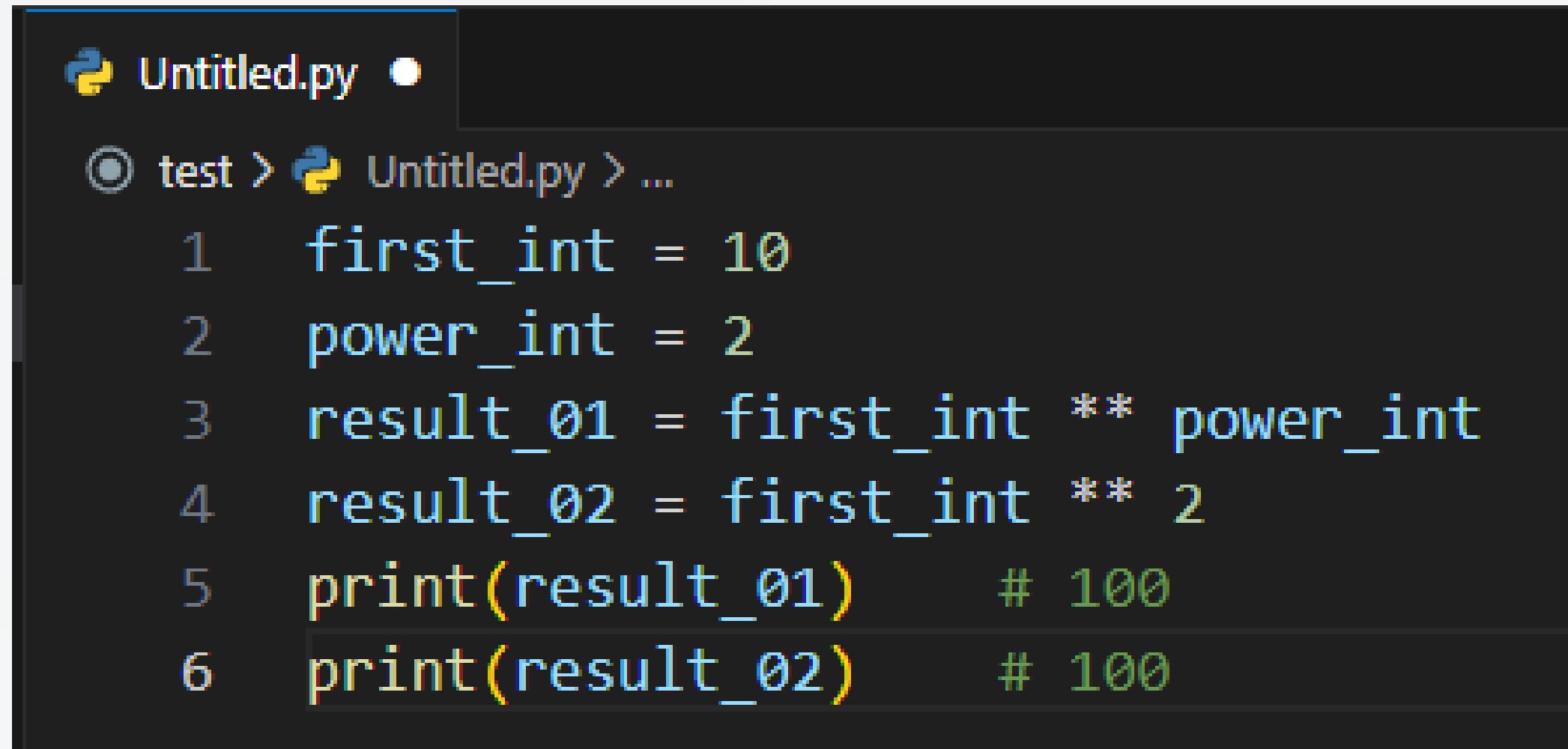
จากตัวอย่าง การหาร (/) จะได้ตัวแปรต่างกัน



```
Untitled.py
test > Untitled.py > ...
1 first_int = 10
2 second_int = 5
3 third_int = 4
4 result_01 = first_int / second_int
5 result_02 = first_int / third_int
6 print(result_01)      # 2.0
7 print(result_02)      # 2.5
```

- **บรรทัดที่ 1-3** ประกาศตัวแปรจำนวนเต็ม (int)
- **บรรทัดที่ 4-5** ประกาศตัวแปรเพื่อเก็บผลลัพธ์ของการหาร (/)
- **บรรทัดที่ 6** print แสดงค่า result_01 ผลลัพธ์คือ เป็นตัวเลขทศนิยมเสมอ float
- **บรรทัดที่ 7** print แสดงค่า result_02 ผลลัพธ์คือ เป็นตัวเลขทศนิยมเสมอ float

จากตัวอย่าง การยกกำลัง (**) ชนิดตัวแปรต่างกัน



```
Untitled.py •  
● test > Untitled.py > ...  
1 first_int = 10  
2 power_int = 2  
3 result_01 = first_int ** power_int  
4 result_02 = first_int ** 2  
5 print(result_01)      # 100  
6 print(result_02)      # 100
```

- **บรรทัดที่ 1-2** ประกาศตัวแปรจำนวนเต็ม (int)
- **บรรทัดที่ 3-4** ประกาศตัวแปรเพื่อเก็บผลลัพธ์ของการยกกำลัง (**)
- **บรรทัดที่ 5** print แสดงค่า result_01 ผลลัพธ์คือ 100
- **บรรทัดที่ 6** print แสดงค่า result_02 ผลลัพธ์คือ 100

จากตัวอย่าง หารเอาเศษและไม่เอาเศษ (//) , (%)

```
Untitled.py x
① test > Untitled.py > ...
1 first_int = 10
2 second_int = 2
3 third_int = 3.0
4 result_01 = first_int // second_int      # ผลลัพธ์การหาร
5 result_02 = first_int % third_int        # แสดงผลลัพธ์ที่เหลือจากการหาร
6 print(result_01)    # 5
7 print(result_02)    # 1.0
```

- **บรรทัดที่ 1-3** ประกาศตัวแปรจำนวนเต็ม (int) , ทศนิยม (float)
- **บรรทัดที่ 4-5** ประกาศตัวแปรเพื่อเก็บผลลัพธ์ของการหารเอาเศษและไม่เอาเศษ (//) , (%)
- **บรรทัดที่ 6** print แสดงค่า result_01 ผลลัพธ์คือ 5
- **บรรทัดที่ 7** print แสดงค่า result_02 ผลลัพธ์คือ 1.0

ลำดับในการคำนวณ

ลำดับการคณิตฯ

- คำนวณค่าต่างๆ ภายในเครื่องหมายวงเล็บ () ก่อนเสมอ
- คำนวณค่าในส่วนที่เป็นเลขยกกำลัง (**)
- คำนวณค่า **คูณ** (*) หรือ **หาร** (/) โดยมีค่าลำดับความสำคัญเท่ากัน
- คำนวณค่า **บวก** (+) หรือ **ลบ** (-) โดยมีค่าลำดับความสำคัญลำดับสุดท้าย
 - โดยจะคำนวณ **จากซ้าย ไปขวา** เสมอ
 - **จากตัวอย่าง** $(1*10) ** 2 - 8 + 4 / 5$

ลำดับในการคำนวณ

ลำดับทางคณิตฯ

- จากตัวอย่าง $(1*10) ** 2 - 8 + 4 / 5$

- บรรทัดที่ 1 คำนวณในวงเล็บก่อน $(1*10)$
- บรรทัดที่ 2 คำนวณตัวเลขยกกำลัง $(10)**2$
- บรรทัดที่ 3 คำนวณการหาร $(4/5)$
- บรรทัดที่ 4 เมื่อเจอเครื่องบวกหรือลบ จะคำนวณจากซ้ายไปขวา $(100-8)$
- บรรทัดที่ 5 คำนวณการบวก $(92+0.8)$
- บรรทัดที่ 6 ผลลัพธ์คือ (92.8)

1	$(1*10) ** 2 - 8 + 4 / 5$
2	$10 ** 2 - 8 + 4 / 5$
3	$100 - 8 + 4 / 5$
4	$100 - 8 + 0.8$
5	$92 + 0.8$
6	92.8

WORKSHOP 2

- ให้ใช้ฟังก์ชัน input เพื่อรับค่าตัวเลขชุดที่ 1 และ 2
- โดยใช้ฟังก์ชัน int() เพื่อแปลงข้อความให้เป็นตัวเลข
- จากนั้นทำการ บวก ลบ คูณ หาร ของตัวเลขทั้ง 2 ชุด
- แสดงผลลัพธ์ที่หน้าจอทั้ง 4 แบบ

1. แสดงข้อความบอกให้ผู้ใช้กรอกตัวเลข
2. แสดงผลลัพธ์บอกตัวเลขที่กรอก
3. แสดงผลลัพธ์ทั้ง 4 แบบ

