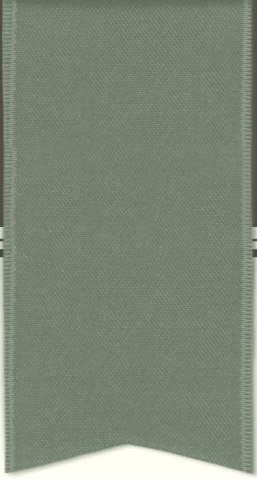


ITD62-123 COMPUTER PROGRAMMING

IMI62-122 FUNDAMENTAL OF COMPUTER PROGRAMMING

Theerat Saichoo & Yunyong Punsawad
School of Informatics, Walailak University



Chapter 1

Introduction to computer programming

Theerat Saichoo
School of Informatics, Walailak University





Part 3: Operators and Formatted

Topics

- Operators
- User input
- Python String Formatting

Operators

- An operator is a symbol that tells the compiler to perform specific mathematical or logical functions.
- Python divides the operators in the following groups:
 - Arithmetic operators
 - Assignment operators
 - Comparison operators
 - Logical operators
 - Identity operators
 - Membership operators
 - Bitwise operators

OPERATORS

Operators

■ Arithmetic Operators

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

Operators

■ Assignment operators

Operator	Example	Same As
=	<code>x = 5</code>	<code>x = 5</code>
<code>+=</code>	<code>x += 3</code>	<code>x = x + 3</code>
<code>-=</code>	<code>x -= 3</code>	<code>x = x - 3</code>
<code>*=</code>	<code>x *= 3</code>	<code>x = x * 3</code>
<code>/=</code>	<code>x /= 3</code>	<code>x = x / 3</code>
<code>%=</code>	<code>x %= 3</code>	<code>x = x % 3</code>

Operators

■ Assignment operators (Cont.) ****binary**

Operator	Example	Same As
<code>//=</code>	<code>x //= 3</code>	<code>x = x // 3</code>
<code>**=</code>	<code>x **= 3</code>	<code>x = x ** 3</code>
<code>&=</code>	<code>x &= 3</code>	<code>x = x & 3</code>
<code> =</code>	<code>x = 3</code>	<code>x = x 3</code>
<code>^=</code>	<code>x ^= 3</code>	<code>x = x ^ 3</code>
<code>>>=</code>	<code>x >>= 3</code>	<code>x = x >> 3</code>
<code><<=</code>	<code>x <<= 3</code>	<code>x = x << 3</code>

Operators

■ Comparison operators

Operator	Name	Example
<code>==</code>	Equal	<code>x == y</code>
<code>!=</code>	Not equal	<code>x != y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

Operators

■ Logical operators

Operator	Description	Example
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>

Operators

■ Identity operators

Operator	Description	Example
is	Returns True if both variables are the same object	x is y
is not	Returns True if both variables are not the same object	x is not y

Operators

■ Membership operators

Operator	Description	Example
<code>in</code>	Returns True if a sequence with the specified value is present in the object	<code>x in y</code>
<code>not in</code>	Returns True if a sequence with the specified value is not present in the object	<code>x not in y</code>

Operators

■ Bitwise operators

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

User input

- Python allows for user input. That means we are able to ask the user for input.
- Python 3.6 uses the `input()` method.
- Syntax

```
variable = input(request text)
```

- Examples

```
username = input("Enter username:")
```

```
gpax = input("Enter GPAX:")
```

Python String Formatting

- To make sure a string will display as expected, The **format()** method allows you to format selected parts of a string.
- To control such values, add placeholders (curly brackets `{}`) in the text, and run the values through the `format()` method:

Python String Formatting

- **Method #1 Single Value**

- Add a placeholder where you want to display the price:

- Ex.

```
price = 49  
txt = "The price is { } dollars."  
print(txt.format(price))
```

- Output

The price is 49 dollars.

Python String Formatting

- **Method #1.1 Single Value – two decimals**
- Add a placeholder where you want to display the price:
- Ex.

```
price = 49  
txt = "The price is {:.2f} dollars."  
print(txt.format(price))
```

Output

The price is 49.00 dollars.

Python String Formatting

- **Method #2 Multiple Values**

- And add more placeholders:

- Ex. `quantity = 3`

`itemno = 567`

`price = 49`

`myorder = "I want {} pieces of item number {} for {:.2f} dollars."`

`print(myorder.format(quantity, itemno, price))`

- Output

I want 3 pieces of item number 567 for 49.00 dollars.

Python String Formatting

- **Method #3 Index Numbers**

- Use index numbers (a number inside the curly brackets `{0}`) to be sure the values are placed in the correct placeholders:

- Ex. `quantity = 3`
`itemno = 567`
`price = 49`
`myorder = "I want {0} pieces of item number {1} for {2:.2f} dollars."`
`print(myorder.format(quantity, itemno, price))`

- **Output**

I want 3 pieces of item number 567 for 49.00 dollars.

Python String Formatting

- **Method #4 Named Indexes**

- Use named indexes by entering a name inside the curly brackets {name} but then you must use names when you pass the parameter values `txt.format(name = "XXXX")`:

- Ex. `myorder = "I have a {carname}, it is a {model}."`

`print(myorder.format(carname = "Ford", model = "Ranger"))`

- **Output**

I have a **Ford**, it is a **Ranger**.

Class Activity 2: Operators and Formatted

Direction: Write the Python commands to action following.

- Print the text “***School of Informatics***” on a screen.
- using string formatting method#1
 - Print a value from variable ***iPhonePrice*** on a screen.
 - Print a value from variable ***gpax*** that show only 2 decimals.
- using string formatting method#2.
 - Print a value from variable ***student_name*** and ***student_id*** in the same line.
- using string formatting method#3.
 - Get 2 user input to 2 variable ***university_name*** and ***faculty_name***.
 - Print a value from variable ***university_name*** and ***faculty_name*** in the same line.

THE END