LECTURE 2: VARIABLES, EXPRESSIONS, STATEMENTS

Hung-Yu Wei

Assignment statement

- Syntax: assignment

  =

- Example

  message = 'I like programming'

  n=60

  pi=3.1415

```
1  message = 'I like computer programming :)'
2  n=17
3  pi=3.1415
```

```
1  message = 'I like computer programming :)'
2  n=17
3  pi=3.1415
4  print (message)
5  print (n)
6  print (pi)
```

```
I like computer programming :)
17
3.1415
```

Rules of Naming Variables

- Variables names must start with a letter or an underscore _

  myVariable_ _my_variable

- The remainder of your variable name may consist of letters, numbers and underscores.
- Names are case sensitive

  xyz XYZ xYz XYz

- Avoid keywords

```
>>> 76trombones = 'big parade'
SyntaxError: invalid syntax
>>> more@ = 1000000
SyntaxError: invalid syntax
>>> class = 'Advanced Theoretical Zymurgy'
SyntaxError: invalid syntax
```

Keywords in Python

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Expression and Statement

- Expression
  - *A combination of values, variable, operations*

    42

    n

    n+25
- Assign a value to a variable
  - *Assignment* =
  - *Value* *21*
  - *Variable* *n*
- Statement to display the value of a variable on screen

```
1  print (5)
2  n = 21
3  print (n)
4  print (n*2)
```

```
5
21
42
```

Syntax: print

- Display text on screen
- print

```
print ('Hello World')

N=3
print(N)
```

- Syntax

```
print(' text to be displayed')
print(variableName)
```

Example: using variables for arithmetic operations

```
1  totalMoney = 300
2  priceCake = 60
3  totalMoney=totalMoney-priceCake
4  print(totalMoney)
5  print(totalMoney-priceCake)
```

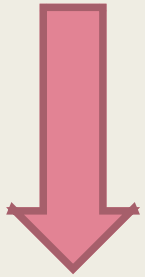
```
240
180
```

Order of Operation

- You have learned this in math class (e.g. 先乘除後加減)
- Operators of the same precedence are evaluated "**from left to right**"

- Order of operations

  Parentheses ()
  *Exponential ***
  ** /*
  + -



- My suggestion: Use ( ) to avoid confusion

String

- double-quotes (") and single-quotes (') are both ok
- opened and closed by the same type of quote mark

- Example

  String1 = “This is a cool string @#$%^^& “

  x = ‘ x is a string’

String Operation

- String concatenation **+**

```
1  first_string = 'Today is'
2  second_string = ' Tuesday!'
3  first_string + second_string
```

```
'Today is Tuesday!'
```

- String repetition with ***

```
1  myString = 'Ha! '
2  myString * 3
```

```
'Ha! Ha! Ha! '
```

Comments

- Start with #
- Syntax

  # write your comments here

- Where to write your comments
  - *Beginning of a file (e.g. your name/NTU ID/homework #/how to run it)*
  - *Before a statement or at the end of a statement*

```
# compute the percentage of the hour that has elapsed
percentage = (minute * 100) / 60
```

```
percentage = (minute * 100) / 60      # percentage of an hour
```

Comments

- Make your codes clear
  - *What the code does*
  - *Explain why*
    - Meaning of the variables/statements
- Good practices
  - *Write meaningful comments*
  - *Naming your variables carefully*

This comment is redundant with the code and useless:

```
v = 5      # assign 5 to v
```

This comment contains useful information that is not in the code:

```
v = 5      # velocity in meters/second.
```

Debugging and Errors

- Syntax error
- Runtime error
- Semantic error

Reading

- Chapter 2 in textbook “Think Python”

- Write the codes and run them
  - *E.g. Exercise 2.1*