LEARNING TO CODE

python™

Coordinator: Konstantinos Emmanouilidis

IEEE SB OF THRACE

2017-2018

1. What is Python?
2. Variables
3. String Functions
4. Casting
5. Basic Operators
6. Input & Output
7. Decision Making

WHAT IS PYTHON ?

- object-oriented
- interpreted
- high-level programming language with dynamic semantics.
- Useful modules and packages
- Lot’s of libraries….

1. What is Python?
2. Variables
3. String Functions
4. Casting
5. Basic Operators
6. Input & Output
7. Decision Making

VARIABLES

- memory locations to store values
- No need for type declaration of a variable
- Automatical declaration while assigning values in variables

```
counter = 100          # An integer assignment
miles   = 1000.0       # A float
name    = "John"       # A string
```

VARIABLE TYPES

- Numbers (Αριθμοί)
  - store numeric values
- Strings (Αλφαριθμητικά)
  - set of characters represented in the quotation marks
- Boolean (Λογικές Μεταβλητές)
  - True or False
- Lists (Λίστες)
  - Next time....
- Tuple
  - Coming soon....
- Dictionary

VARIABLE ASSIGNMENT

- The equal sign (=) is used to assign values to variables
- Python allows you to assign a single value to several variables simultaneously

```
a,b,c = 1,2,"john"
```

```
a = b = c = 1
```

STRINGS

- A set of characters in quotes (single or double)
- Use double quotes especially when the string has single quotes itself.
- There is no character type. Character = string of length 1

```
a='Nick'
b="John"
c="It's time to use double quotes"
d='It\'s also ok if you use \ and then the single quote'
```

STRING OPERATORS

		If a="Hello" and b="Python"
+	Concatenation (Ενώνει string μεταξύ τους)	a + b gives "HelloPython"
*	Repetition (Ενώνει τόσα αντίγραφα του αριστερού τελευσταίου όσος είναι ο δεξιά τελεστής)	a*2 gives "HelloHello"
[]	Slice (Δίνει ένα χαρακτήρα του string) (index starts from 0)	a[1] gives "e"
[:]	Range Slice (Δίνει τους χαρακτήρες από ένα εύρος του string)	a[1:4] gives "ell"
in	Membership (Επιστρέφει True αν ένας χαρακτήρας υπάρχει στο string)	(H in a) gives True
not in	Membership (Επιστρέφει True αν ένας χαρακτήρας δεν υπάρχει στο string)	(M not in a) gives True
%	Format (Αλλάζει το format του string)	See at next section

EXAMPLES

```
#!/usr/bin/python3

var1 = 'Hello World!'
var2 = "Python Programming"

print ("var1[0]: ", var1[0])
print ("var2[1:5]: ", var2[1:5])
```

When the above code is executed, it produces the following result –

```
var1[0]:  H
var2[1:5]:  ytho
```

```
#!/usr/bin/python3

str = 'Hello World!'

print (str)          # Prints complete string
print (str[0])       # Prints first character of the string
print (str[2:5])     # Prints characters starting from 3rd to 5th
print (str[2:])      # Prints string starting from 3rd character
print (str * 2)      # Prints string two times
print (str + "TEST") # Prints concatenated string
```

STRING FORMATTING OPERATOR

Format Symbol	Conversion
%c	character
%s	string conversion via str() prior to formatting
%i	signed decimal integer
%d	signed decimal integer
%u	unsigned decimal integer
%o	octal integer
%x	hexadecimal integer (lowercase letters)
%X	hexadecimal integer (UPPERcase letters)
%e	exponential notation (with lowercase 'e')
%E	exponential notation (with UPPERcase 'E')
%f	floating point real number
%g	the shorter of %f and %e
%G	the shorter of %f and %E

EXAMPLE

```
print ("My name is %s and weight is %d kg!" %('Zara', 71))
```

When the above code is executed, it produces the following result:

```
“My name is Zara and weight is 71 kg!”
```

ESCAPE CHARACTERS - BACKSLASH NOTATION

\n	**New line**
\nnn	Octal notation, where n is in the range 0.7
\r	Carriage return
\s	Space
\t	Tab
\v	Vertical tab
\x	Character x
\xnn	Hexadecimal notation, where n is in the range 0.9, a.f, or A.F

\a	Bell or alert
\b	Backspace
\cx	Control-x
\C-x	Control-x
\e	Escape
\f	Formfeed
\M-\C-x	Meta-Control-x

1. What is Python?
2. Variables
3. String Functions
4. Casting
5. Basic Operators
6. Input & Output
7. Decision Making

THE LEN() FUNCTION

- Returns the length of the string
- Syntax : len(string)

```
#!/usr/bin/python3

str = "this is string example....wow!!!"

print ("Length of the string: ", len(str))
```

Result

When we run above program, it produces the following result –

```
Length of the string:  32
```

THE LOWER() FUNCTION

- Function **lower()** returns a copy of the string in which all case-based characters have been lowercased.

- Syntax : string**.lower()**

```
#!/usr/bin/python3

str = "THIS IS STRING EXAMPLE....WOW!!!"

print (str.lower())
```

Result

When we run above program, it produces the following result –

```
this is string example....wow!!!
```

THE UPPER() FUNCTION

- The **upper()** method returns a copy of the string in which all case-based characters have been uppercased.

- Syntax : string.upper()

```
#!/usr/bin/python3

str = "this is string example....wow!!!"
print ("str.upper : ",str.upper())
```

Result

When we run above program, it produces the following result –

```
str.upper :  THIS IS STRING EXAMPLE....WOW!!!
```

THE ISSPACE() FUNCTION

- The isspace() method checks whether the string consists of whitespace.
- Returns true if there are only whitespace characters in the string and there is at least one character,otherwise false.

- Syntax : string.isspace()

```
#!/usr/bin/python3

str = "        "
print (str.isspace())

str = "This is string example....wow!!!"
print (str.isspace())
```

Result

When we run above program, it produces the following result –

```
True
False
```

THE SWAPCASE() FUNCTION

- The **swapcase()** method returns a copy of the string in which all the case-based characters have had their case swapped.

- Syntax : string.swapcase()

```
#!/usr/bin/python3

str = "this is string example....wow!!!"
print (str.swapcase())

str = "This Is String Example....WOW!!!"
print (str.swapcase())
```

Result

When we run above program, it produces the following result –

```
THIS IS STRING EXAMPLE....WOW!!!
tHIS iS sTRING eXAMPLE....wow!!!
```

1. What is Python?
2. Variables
3. String Functions
4. Casting
5. Basic Operators
6. Input & Output
7. Decision Making

CASTING

1. **int(x)**
   Μετατροπή του x σε ακέραιο.
2. **float(x)**
   Μετατροπή του x σε float.
3. **long(x)**
   Μετατροπή του x σε long int.
4. **str(x)**
   Μετατροπή του x σε string.
5. **complex(real[,imag])**
   Μετατροπή του x σε μιγαδικό.
6. **chr(x)** (όπου x ακέραιος)
   Μετατροπή x σε χαρακτήρα.
7. **tuple(x)**
   Μετατροπή του x σε tuple.
8. **list(x)**
   Μετατροπή του x σε λίστα.
9. **set(x)**
   Μετατροπή του x σε set.
10. **frozenset(x)**
    Μετατροπή του x σε frozen set.

1. What is Python?
2. Variables
3. String Functions
4. Casting
5. Basic Operators
6. Input & Output
7. Decision Making

OPERATORS(ΤΕΛΕΣΤΕΣ)

- Arithmetic (Αριθμητικοί)
- Comparison (Σχεσιακοί)
- Logical (Λογικοί)
- Assignment Operators (Ανάθεσης)
- Bitwise Operators
- Membership Operators
- Identity Operators

ARITHMETIC OPERATORS

		If a=2 and b=5
+	Addition	a + b = 7
-	Subtraction	a – b = -3
*	Multiplication	a * b = 10
/	Division	b / a = 2.5
%	Modulus: returns the remainder of the division.	b % a = 1
**	Power(Exponent)	a**b =32
//	Floor Division	b//a = 2 and 5.0//2.0 = 2.0

COMPARISON OPERATORS

		If a=2 and b=5
==	Equals sign	(a == b) is False
!=	Not equals sign	(a!= b) is True
>	Greater than sign	(a > b) is False
<	Less than sign	(a < b) is True
>=	Greater than or equal sign	(a >= b) is False
<=	Less than or equal sign	(a <= b) is True

LOGICAL OPERATORS

		If a=True and b=False
and	Logical AND: If both the operands are true then condition becomes true.	(a and b) is False.
or	Logical OR: If any of the two operands is true then condition becomes true.	(a or b) is True.
not	Logical NOT: Reverses the logical state of its operand.	not(a and b) is True.

ASSIGNMENT OPERATORS

=	Assigns values from right side operands to left side operand	c = a + b assigns (a + b) to c
+=	Assigns the sum of left and right operand to the left operand	c += a equivalent to c = c + a
-=	Assigns the result of the subtraction of right operand from the left operand	c -= a equivalent to c = c - a
*=	Assign the result of multiplication to the left operand	c *= a equivalent to c = c * a
/=	Assign the result of division to the left operand	c /= a equivalent to c = c /a
%=	Assign the result of modulus to the left operand	c %= a equivalent to c = c % a
**=	Assign the result of the power to the left operand	c **= a equivalent to c = c ** a

OTHER TYPES OF OPERATORS

- Bitwise Operators

  Perform bit-by-bit operation

- Membership Operators

  Test for membership in a sequence, such as strings, lists, or tuples.

  - in : returns 1 if left operand belongs to right operand
  - not in : returns 1 if left operand doesn’t belong to right operand

- Identity Operators

  Compare the memory locations of two objects.

1. What is Python?
2. Variables
3. String Functions
4. Casting
5. Basic Operators
6. Input & Output
7. Decision Making

BASIC INPUT AND OUTPUT

- Output

print() function

Prints *objects* to the text stream *file*

- Input

input([prompt])

If the *prompt* argument is present, it is written to standard output without a trailing newline. The function then reads a line from input, converts it to a string and returns it.

!!! Necessary casting: **int(input([prompt]))** in order to compare with integers

EXAMPLES

```
#!/usr/bin/python3

amount = int(input("Enter amount: "))

if amount<1000:
   discount = amount*0.05
   print ("Discount",discount)
else:
   discount = amount*0.10
   print ("Discount",discount)

print ("Net payable:",amount-discount)
```

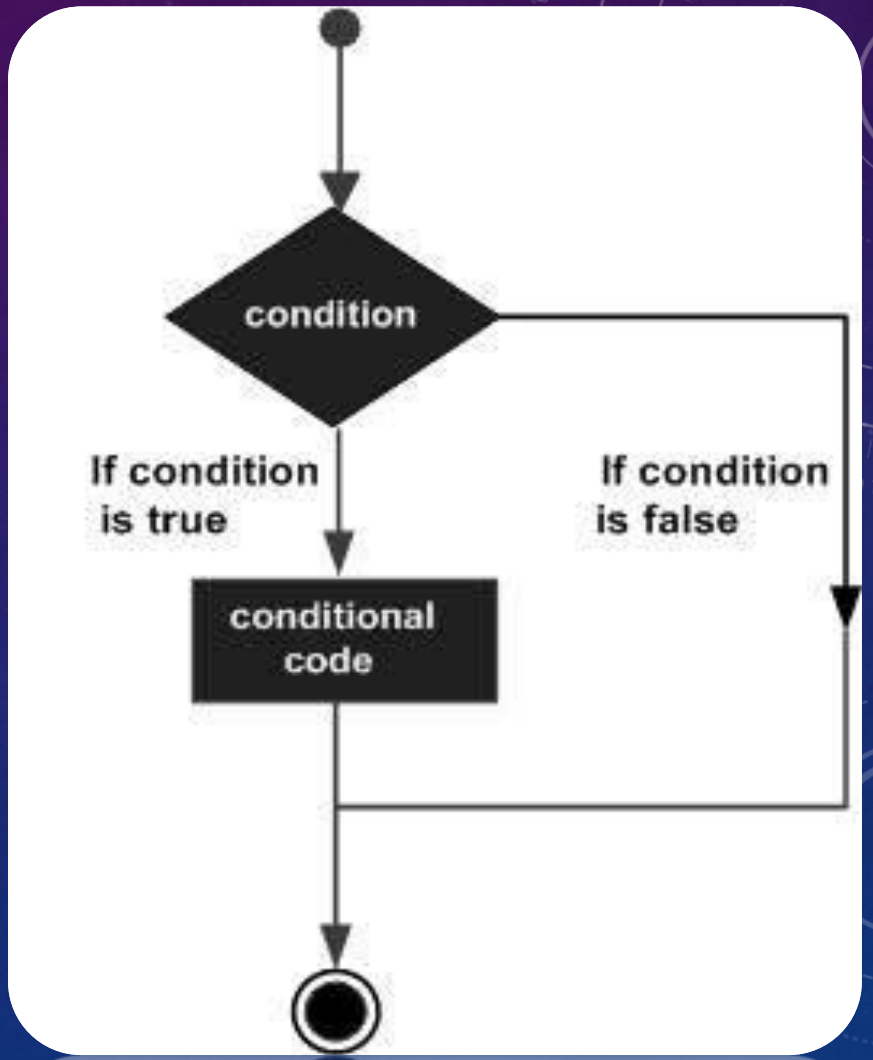
```
#!/usr/bin/python3

var1 = 100
if var1:
   print ("1 - Got a true expression value")
   print (var1)

var2 = 0
if var2:
   print ("2 - Got a true expression value")
   print (var2)
print ("Good bye!")
```

1. What is Python?
2. Variables
3. String Functions
4. Casting
5. Basic Operators
6. Input & Output
7. Decision Making

DECISION MAKING



IF STATEMENT

- If Syntax

```
if condition:
        statement 1
        statement 2
        statement 3
        .
        .
        .
```

- If…Else Syntax

```
if expression:
        statement(s)
else:
        statement(s)
```

- If...Elif...Else Syntax

```
if expression1:
        statement(s)
elif expression2:
        statement(s)
elif expression3:
        statement(s)
else:
        statement(s)
```

Statements in a block are uniformly indented after the : symbol.

```
#!/usr/bin/python3

amount = int(input("Enter amount: "))

if amount<1000:
    discount = amount*0.05
    print ("Discount",discount)
elif amount<5000:
    discount = amount*0.10
    print ("Discount",discount)
else:
    discount = amount*0.15
    print ("Discount",discount)
```