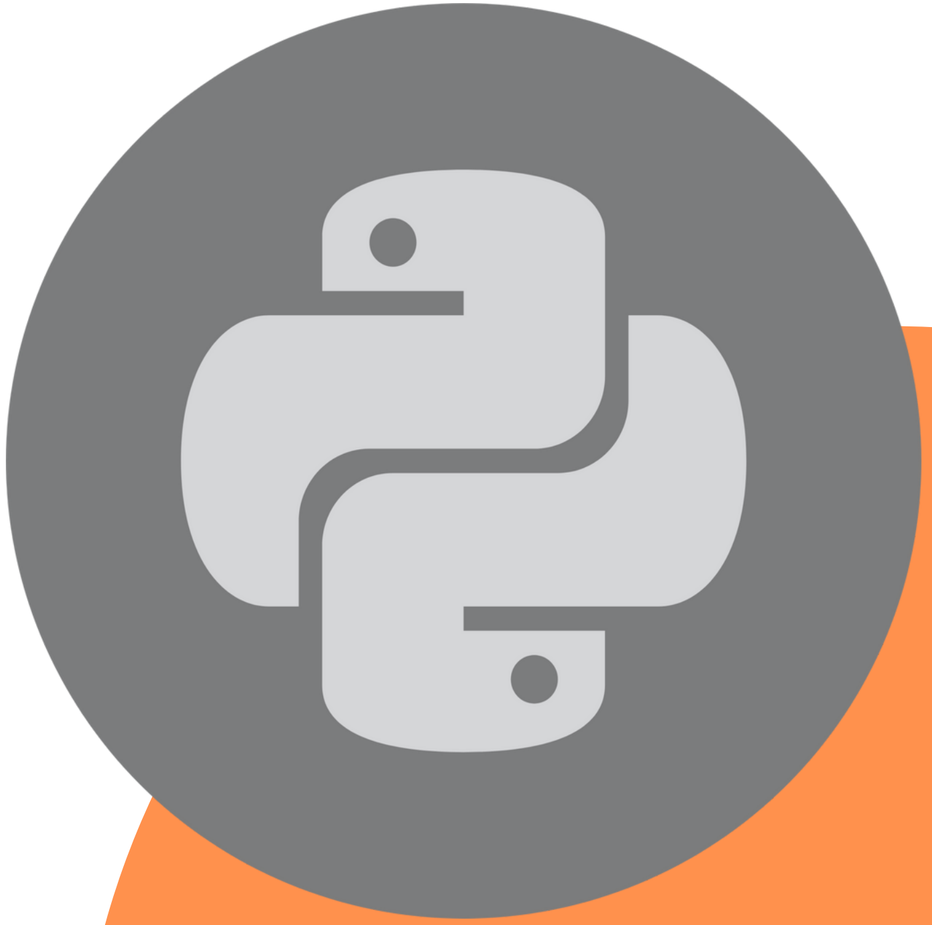
PYTHON COURSE

ENTRY LEVEL

Basics of programming in Python 3.10

This course will cover part of the arguments found in PCEP™ – Certified Entry-Level Python Programmer Certification



OPERATORS AND OPERATIONS

SHORTCUT OPERATORS

Special type of operators. The shortcut operator is used to perform both calculation and assignment operations

```
a = 7

a += 3 # same as a = a + 3

a -= 3 # same as a = a - 3

a /= 2 # same as a = a / 2

a *= 2 # same as a = a * 2
```

BOOLEAN EXPRESSIONS

Boolean expressions are operations that result in True or False

RELATIONAL OPERATORS

equals: **==** operator (returns True if confronted objects have same values, otherwise False)

```
a = 7

b = 7

print(a == b) # prints True
```

PYTHON

BOOLEAN EXPRESSIONS

not equals: != operator (returns True if objects are different in values, otherwise False)

```
a = 7

b = 7

print(a != b) # prints False
```

PYTHON

BOOLEAN EXPRESSIONS

higher then: > operator (returns True if first numeric value is higher than second)

```
a = 7

b = 5

print(a > b) # prints True
```

lower then: < operator (returns True if first numeric value is lower than second)

```
a = 7

b = 5

print(a < b) # prints False
```

PYTHON

BOOLEAN EXPRESSIONS

equals or higher: >= operator (returns True if first numeric value equals or is higher than second)

equals or lower: <= operator (returns True if first numeric value equals or is lower than second)

```
a = 7

b = 5

c = 5

print(a >= b) # prints True
print(b >= a) # prints False
print(b <= c) # prints True
print(c <= a) # prints True
```

BOOLEAN EXPRESSIONS

BOOLEAN OPERATORS

- **not** (negation)

Used to negate an expression (if that expression results in True, negated version will be False). In Python, is used as a keyword.

```
a = 7

bool_expression = not a == 7   # result is False

print(bool_expression)
```

BOOLEAN EXPRESSIONS

- **and**

True if both confronted conditions are True, otherwise False

```
a = 10
b = 5

c = 8

boolean_and = c > b and c > a # False

print(boolean_and)
```

one of the expressions

results in False

BOOLEAN EXPRESSIONS

- **or**

True if at least one of the expressions in True, otherwise False

```
a = 10
b = 5

c = 8

boolean_or = c > b or c > a # True

print(boolean_or)
```

one of the expressions

results in True

PYTHON

BOOLEAN OPERATORS RESULTS TAB

bool_expression_1	boolean_espression_2	and result	or result
True	False	False	True
True	True	True	True
False	True	False	True
False	False	False	False

BOOLEAN OPERATIONS: STRINGS

Some boolean expressions can also be performed on strings

```
a = "Hello"
b = "Hello"

print(a == "Hello") # True

is_greetings = (a == "Hello") or (a == "Hi") # True

print(is_greetings) # True

print(b != a) # False
```

BOOLEAN OPERATIONS: STRINGS

- string membership operators **in, not in**

in operator results True if the first string is contained in the second, otherwise False

```
a = "Hello"

h_in_name = "H" in a

print(h_in_name)
```

BOOLEAN OPERATIONS: STRINGS

not in operator results True if the first string is not contained in the second, otherwise False

```
lecter = "C"

vocals = "aeiouAEIOU"

not_a_vocal = lecter not in vocals # True

print(not_a_vocal)
```

QUESTION TIME

PYTHON

QUESTION 1:

What's the output of the following program:

```
greetings = "Hello sir"

print("S" in greetings)
```

1. True
2. False
3. TypeError
4. S in greetings

QUESTION 2:

What's the output of the following program:

```
b = 8

is_included = (5 <= b) and (b <= 10)

print(is_included)
```

1. True
2. False
3. True and True
4. True and False

QUESTION 3:

What's the output of the following program:

```
x = 8

is_even = x % 2 == 0

print(is_even)
```

1. TypeError at line 3
2. True
3. False
4. SyntaxError

PYTHON

QUESTION 4:

What is the output of the following python program:

```
x = 8

b = "Hello"

print(x < b)
```

1. TypeError at line 5
2. SyntaxError at line 5
3. NameError at line 5
4. None of the above

PYTHON

QUESTION 5:

What is the output of the following python program:

```
starting = "Hello Python"
message = starting + "a" * 3

message = message[:-3]
message + " HELLO"

print(message == starting)
```

1. False
2. True
3. Hello Python
4. Hello Pythonaaa