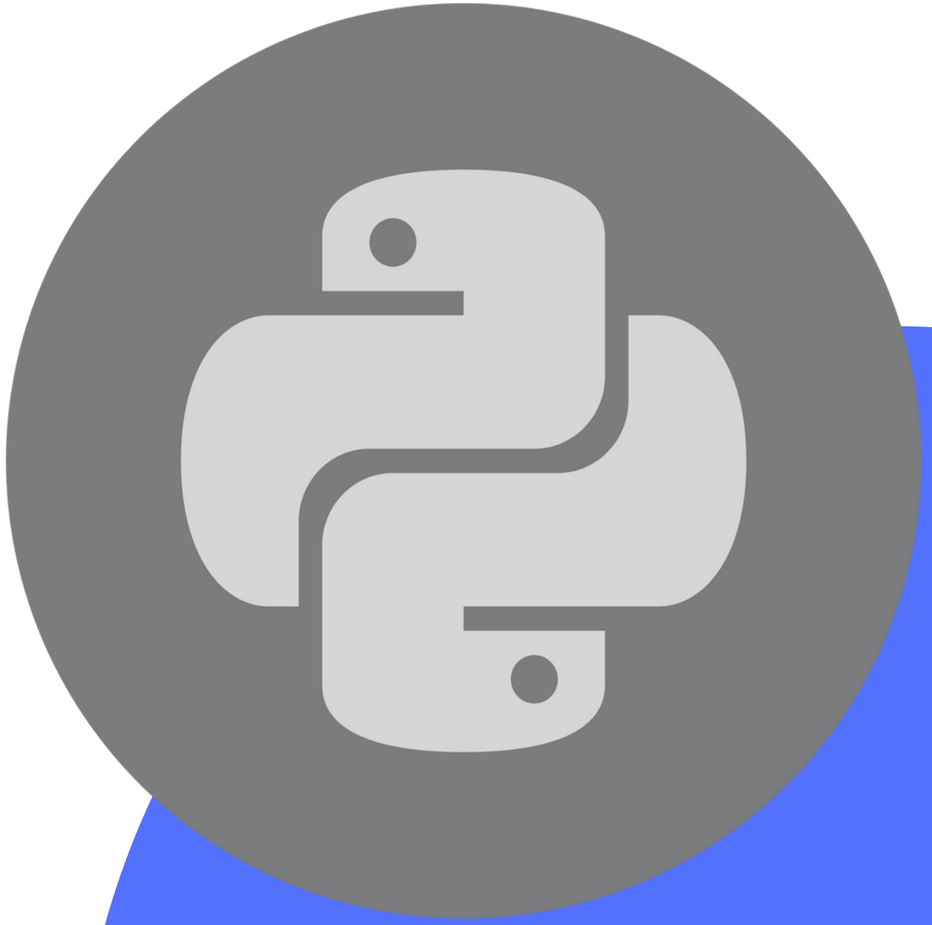
PYTHON COURSE

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

CONTROL FLOW: CONDITIONAL BLOCKS AND LOOPS

PYTHON

CONDITIONAL BLOCKS

conditional blocks are used to change the program flow basing on some conditions:

```
if condition: # condition must be a boolean expression
# all the code indented inside the if block is executed only if condition in True
    print("I'm in the if block")

# after the conditional block, code is executed like always

print(7)
```

PYTHON

CONDITIONAL BLOCKS: INDENTATION

- Indentation is required while building code blocks
- the code to be executed in the conditional block must be indented in it
- an indentation consists in 4 consecutive spaces (or a tab)
- wrong indentation will raise an IndentationError

indentation

```
a = 6

condition = a == 5

if condition:
    print("I'm in the if block")

print(a) # code not indented in the if, so not included in the block
```

PYTHON

CONDITIONAL BLOCKS: SCOPES

a code block has it's own "scope", variables defined in their scopes (all the indented code) only esists within the same scope, they will not be reachable from outside the block

```
a = 6

condition = a == 5

if condition:
    d = 8
    print("I'm in the if block")
    print("last row in the conditional block")

print(d) # will raise a NameError cause d only exists inside the if block
```

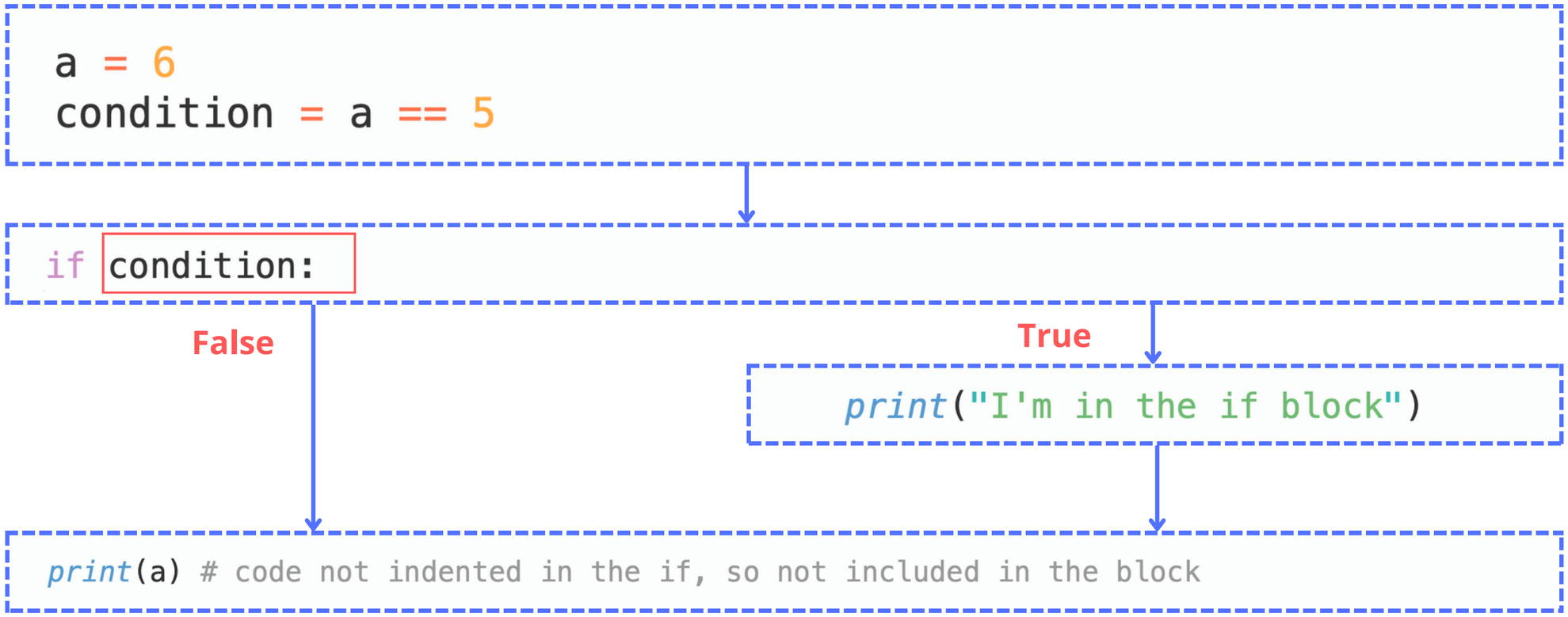
variables defined in the main line, are defined **global variables**
we can access them from everywhere in the program

indented code represent the scope
of the conditional block,
variable d in this case, only exists in it

trying to access variable d outside its scope, will raise a NameError

PYTHON

IF STATEMENT



PYTHON

CONDITIONAL BLOCKS: IF - ELSE

if - else statement will control both False and True output of the verified condition

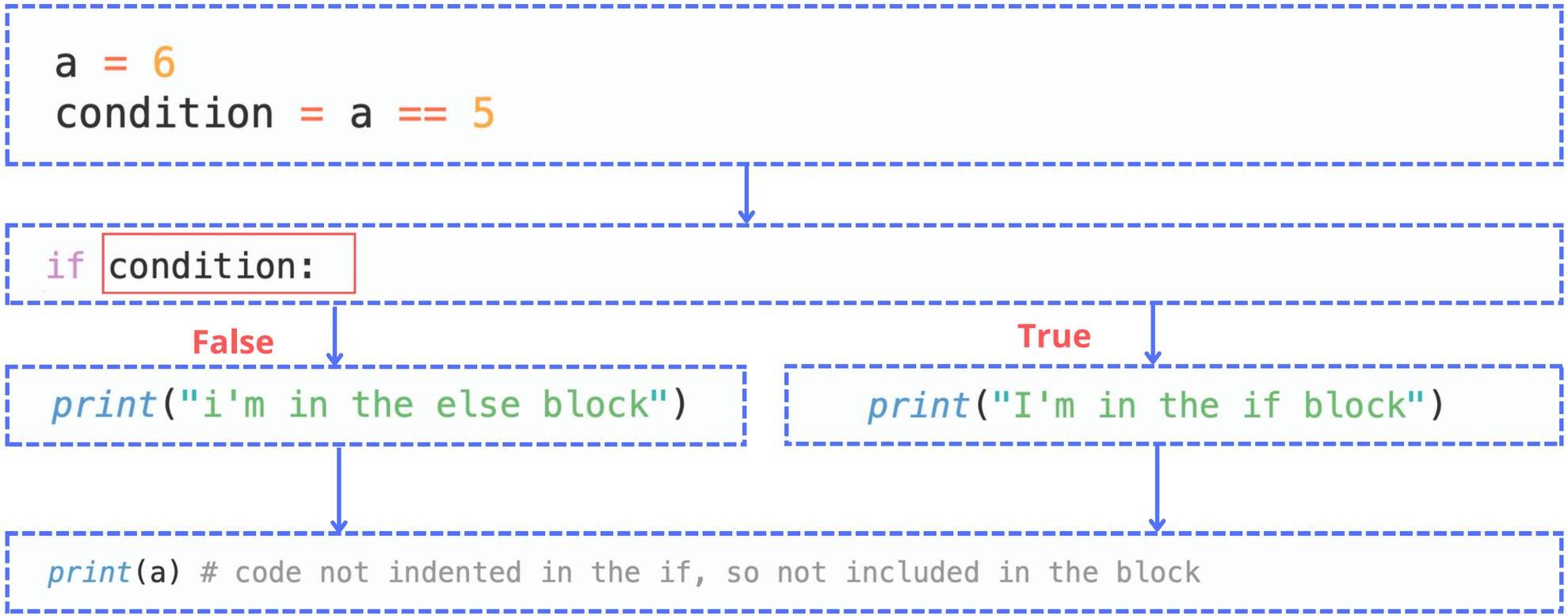
```
a = 6
condition = a == 5

if condition:
    print("I'm in the if block") # executed when condition is True
else:
    print("i'm in the else block") # executed when condition is False

print(a) # code not indented in the if, so not included in the block
```

PYTHON

IF - ELSE STATEMENT



PYTHON

CONDITIONAL BLOCKS: IF - ELIF - ELSE

- Used to check multiple possible conditions
- When executed, conditions are checked sequentially, the one who gives True will make the program execute the correspondent code block.
- When all the conditions are False, the Else block is executed
- The else block is optional

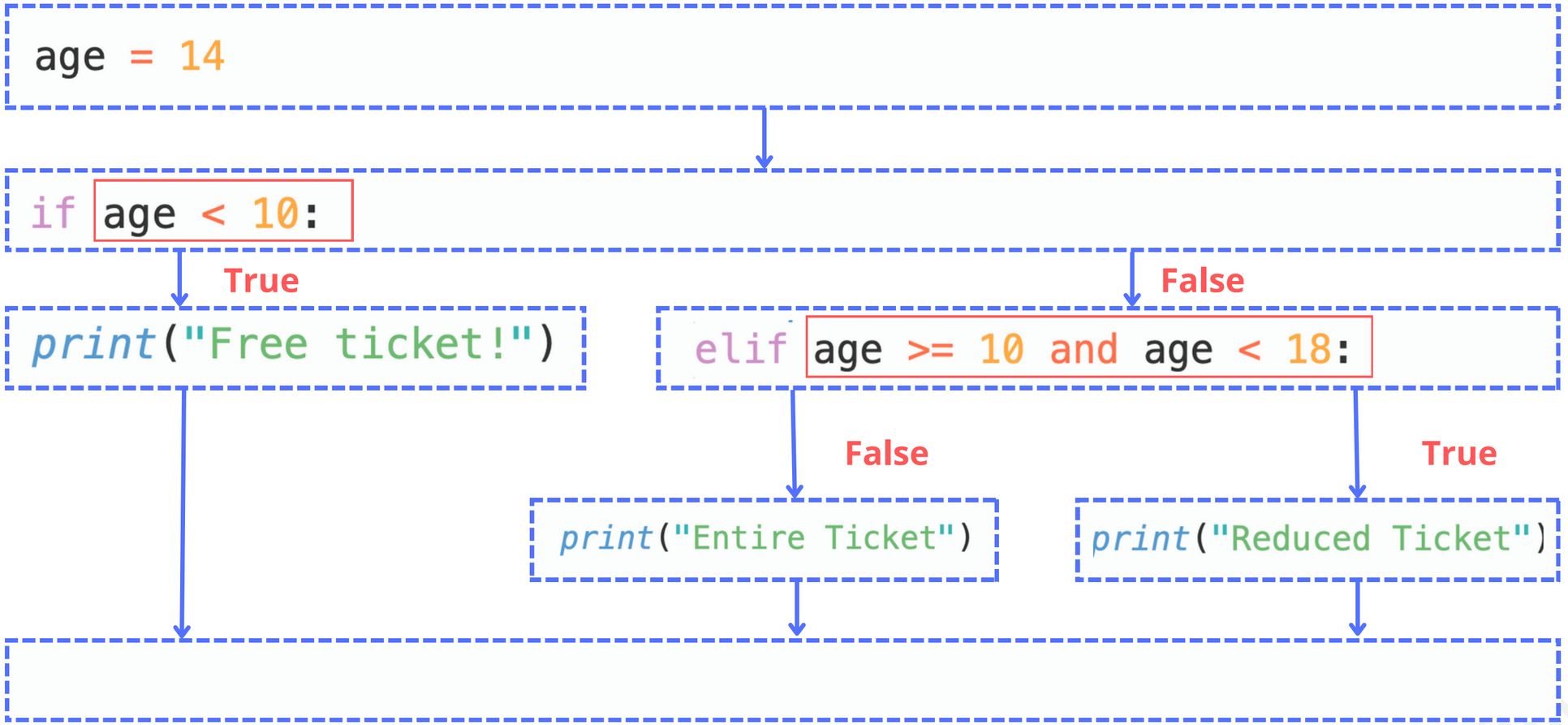
```
age = 14

if age < 10:
    print("Free ticket!") # executed when age < 10 = True
elif age >= 10 and age < 18:
    print("Reduced Ticket") # executed when bot age >= 10 and age < 18 are True
else
    print("Entire Ticket") # executed in the rest of the cases

print(a) # code not indented in the if, so not included in the block
```

PYTHON

IF - ELIF - ELSE STATEMENT



CONDITIONAL BLOCKS: MORE INFO

- In if-elif-else statement you can set how many elif you want but only one else
- in if-elif-else statement, the else block can be omitted
- writing if condition: or if condition == True: its the same thing (so don’t repeat)
- all the code contained in the same block must be indented the same
- every boolean expression can be used as condition for conditional blocks

PYTHON

NESTING CONDITIONAL BLOCKS

Multiple conditional blocks can be nested to create more possible program behaviours

```
a = "Hello"

if a != "":
    if a == "Hello":
        print("Hello to you sir")
    else if a == "Hi":
        print("Hi to you sir")
    else:
        print("This is the else block in the nested loop")
else:
    print("a variable is an empty string")
```

Every conditional block needs a correct indentation or it will cause an error

EXERCISES