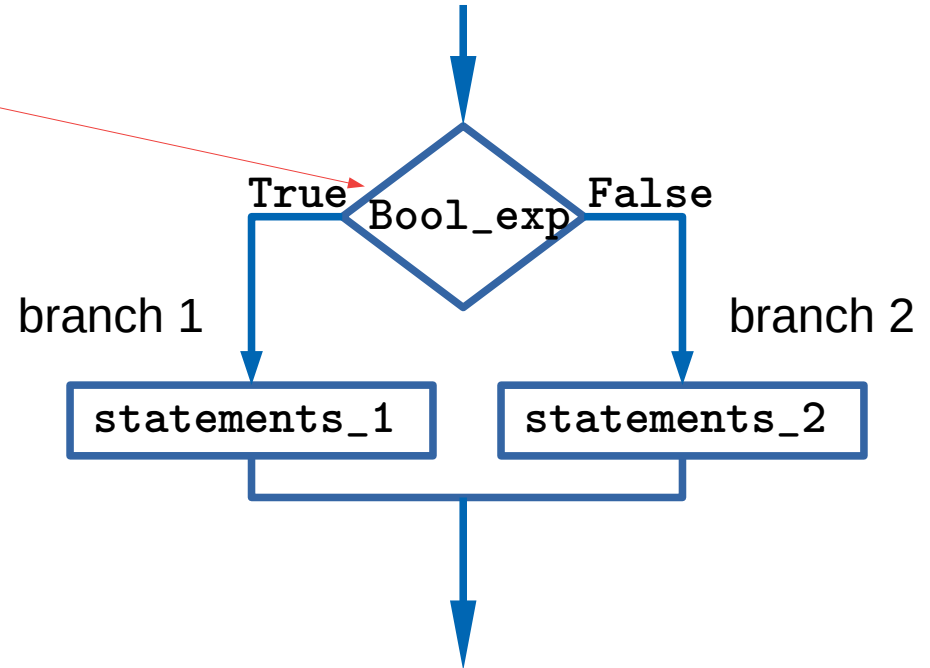


Branching

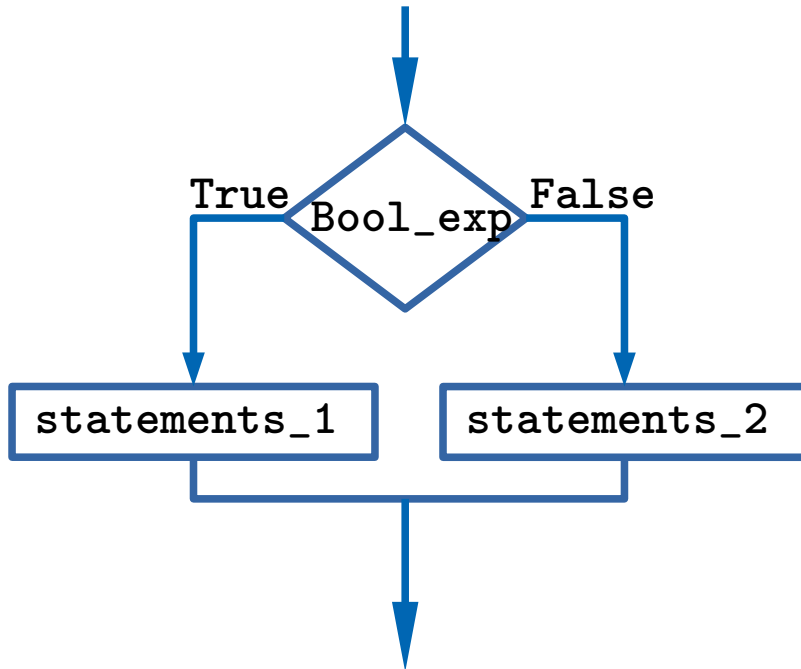
Conditional (if) statement

checks a condition and modifies the program flow accordingly
simple case: 2 branches

evaluates a Boolean expression



Conditional (if) statement



2 compound statements:
beginning with a keyword: `if`, `else`

ending with a colon

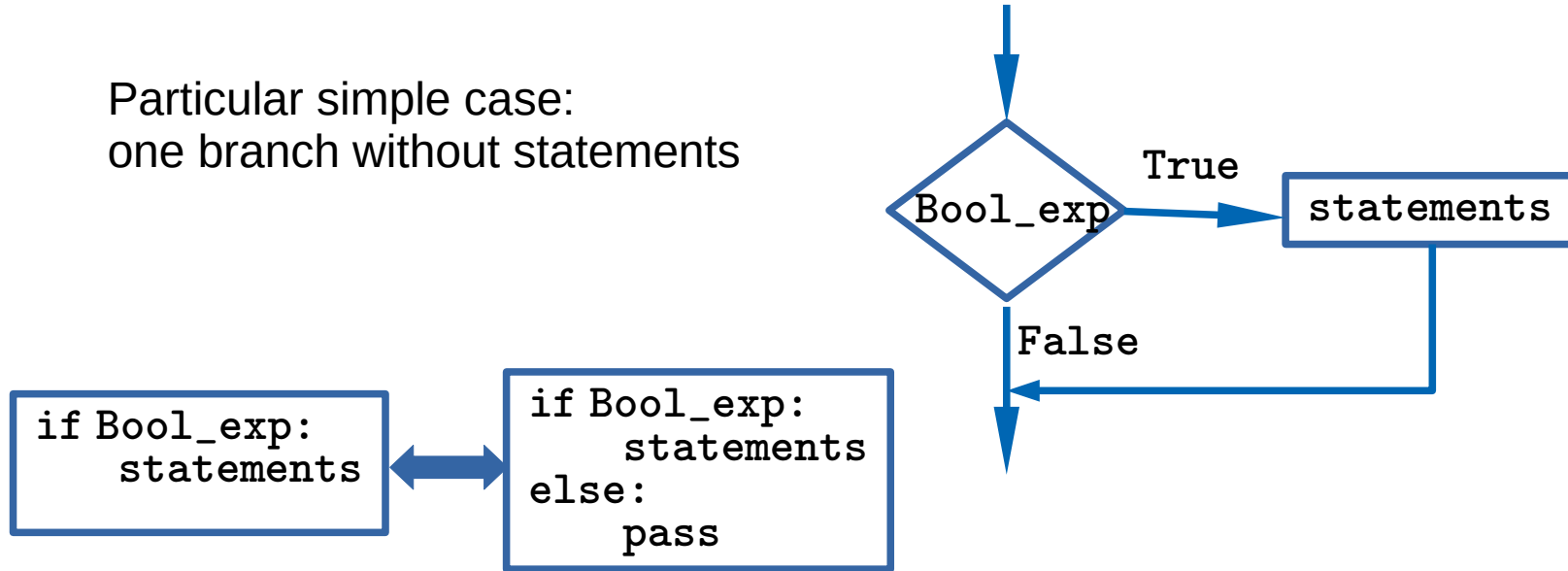
2 branches:
indented statements

```
if Bool_exp:  
    statements_1  
else:  
    statements_2
```

Two red arrows point from the text above to the 'if' and 'else' keywords in the code block.

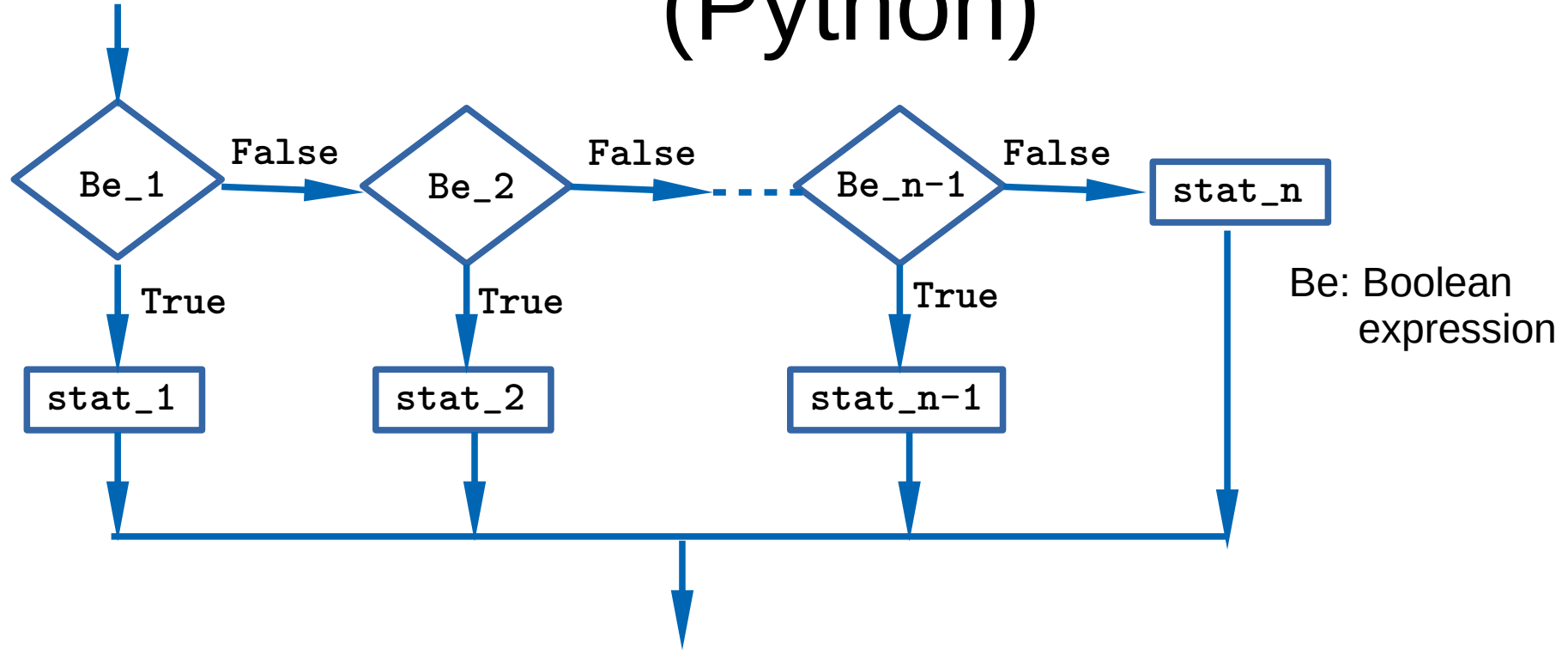
Conditional (if) statement

Particular simple case:
one branch without statements



keyword: pass

Conditional statement: general case (Python)



Condition for $stat_i$ to be executed: Be_i and not Be_1 and not Be_2 and ... and not Be_{i-1}

Conditional statement: general case

```
if Be_1:  
    stat_1  
elif Be_2:  
    stat_2  
...  
elif Be_n-1:  
    stat_n-1  
else:  
    stat_n
```

Be: Boolean
expression

keyword: `elif` (contraction for else if)

Condition for `stat_i` to be executed:
`Be_i and not Be_1 and not Be_2 and ... and not Be_i-1`

Exercise 1

Problem definition: Write function `century` that takes a year (integer between 1900 and 2099) and returns a string showing the corresponding century ('XX' or 'XXI'). Save this function in file `century.py`. You can download this file with tests: [test-century.txt](#)

```
>>> from century import century
```

```
>>> century(1919)
'XX'
```

```
>>> century(1964)
'XX'
```

```
>>> century(2007)
'XXI'
```

```
>>> century(2020)
'XXI'
```

```
>>> century(1999)
'XX'
```

```
>>> century(2001)
'XXI'
```

Exercise 1

Problem definition: Write function `century` that takes a year (integer between 1900 and 2099) and returns a string showing the corresponding century ('XX' or 'XXI').

- Conditional simple: 2 branches
- Two level indentation: function (1st level), if (2nd level)

```
def century(year):  
    if year <= 2000:  
        cent = 'XX'  
    else:  
        cent = 'XXI'  
    return cent
```

Design with a single return statement:

- local variable `cent`
- 1 return statement

```
def century(year):  
    if year <= 2000:  
        return 'XX'  
    else:  
        return 'XXI'
```

Design with several return statements:

- no local variables
- 2 return statements: only one is executed