Study Guide: Conditionals

This study guide provides a quick-reference summary of what you learned in this lesson and serves as a guide for the upcoming practice quiz.

In the Conditionals segment, you learned about the built-in Python operators used for comparing values and the logical operators for making complex comparisons. You also learned how to use operators in if-else-elif blocks.

Knowledge

Comparison operators with numerical values

Comparison expressions return a Boolean result (True or False).

- x **==** y If x is equal to y, return True. Else, return False.
- x **!=** y If x is not equal to y, return True. Else, return False.
- x **<** y If x is less than y, return True. Else, return False.
- x **<=** y If x is less than or equal to y, return True. Else, return False.
- x **>** y If x is greater than y, return True. Else, return False.
- x **>=** y If x is greater or equal to y, return True. Else, return False.

Comparison operators with strings

Comparison expressions with strings also return a Boolean result (True or False).

- "x" **==** "y" If the words are the same, return True. Else, return False.
- "x" **!=** "y" If the words are **not** the same, return True. Else, return False.

When used with strings, the following comparison expressions will alphabetize the strings.

- "x" **<** "y" If string "x" has a smaller Unicode value than string "y", return True. Else, return False.
- "x" **<=** "y" If the Unicode value for string "x" is smaller than or equal to the Unicode value of string "y", return True. Else, return False.
- "x" **>** "y" If string "x" has a larger Unicode value than string "y", return True. Else, return False.
- "x" **>=** "y" If the Unicode value for string "x" is greater than or equal to the Unicode value of string "y", return True. Else, return False.

Unicode values for the alphabet

Uppercase		Uppercase		Lowercase		Lowercase	
Unicode #	Character	Unicode #	Character	Unicode #	Character	Unicode #	Character
65	A	78	N	97	a	110	n
66	B	79	O	98	b	111	o
67	C	80	P	99	c	112	p
68	D	81	Q	100	d	113	q
69	E	82	R	101	e	114	r
70	F	83	S	102	f	115	s
71	G	84	T	103	g	116	t
72	H	85	U	104	h	117	u
73	I	86	V	105	i	118	v
74	J	87	W	106	j	119	w
75	K	88	X	107	k	120	x
76	L	89	Y	108	l	121	y
77	M	90	Z	109	m	122	z

The Unicode numbering for the alphabet starts at 65 for capital letter A and runs to 90 for capital letter Z. Then, the lowercase alphabet values start at 97 for lowercase a and run to 122 for lowercase z. Using these Unicode numbers, capital A's code is less than the codes of all other letters, which Python interprets as the beginning of the alphabet. Lowercase z's code is greater than the codes of all other letters, which Python interprets as the ultimate end of the English alphabet.

Logical operators

Logical operators are used to combine comparison expressions and also return Boolean results (True or False).

- comparison1 **and** comparison2
  - Returns a True result if both comparison1 **and** comparison2 are true.
  - If they are not both true, return False.
- comparison1 **or** comparison2
  - Returns a True result if either comparison1 and/or comparison2 are True.
  - If neither comparison is true, return False.
- **not** comparison1
  - Returns the inverse Boolean value of the comparison.
    - Returns a True result if comparison1 is false.
    - If comparison1 is true, then returns False.

Syntax of an if-elif-else block

```
if condition1:
    action1
elif condition2:
    action2
else:
    action3
```

- If condition1 is True:
  - Then perform action1 and exit if-elif-else block
- If condition2 is True:
  - Then perform action2 and exit if-elif-else block
- If neither condition1 nor condition2 are True:
  - Then perform action3 and exit if-elif-else block

Coding skills

Skill Group 1

- Use a comparison operator with numbers
- Use a comparison operator to alphabetize strings

```
# The value of 10*4 (40) is greater than 14+23 (37), therefore this
# comparison expression will return the Boolean value of True.


print(10*4 > 14+23) # Should print True

# The letter "t" has a Unicode value of 116 and the letter "s" has a
# Unicode value of 115. Since 116 is not less than 115, the
# comparison of "tall" < "short" (or 116 < 115) is False.

print("tall" < "short")  # Should print False
```

Run

Reset

Skill Group 2

- Use a function with the def() keyword
- Pass a parameter to the function
- Use an if-elif-else statement
- Assign strings to variables
- Use conditional operators
- Return a value

```
# This function accepts one variable as a parameter
def translate_error_code(error_code):

# The if-elif-else block assesses the value of the variable
# passed to the function as a parameter. The if statement uses
# the equality operator == to test the value of the variable.
# This test returns a Boolean (True/False) result.
    if error_code == "401 Unauthorized":
# If the comparison above returns True, then the indented
# line(s) inside the if-statement will run. In this case, the
# action is to assign a string to the translation variable.
# The remainder of the if-elif-else block will not run.
# The Python interpreter will skip to the next line outside of
# the if-elif-else block. In this case, the next line is the
# return value statement.
        translation = "Server received an unauthenticated request"

# If the initial if-statement returns a False result, then the
# first elif-statement will run a different test on the value
# of the variable.
    elif error_code == "404 Not Found":
# If the first elif-statement returns a True result, then the
# indented line(s) inside the first elif-statement will run.
# After this line, the remainder of the if-elif-else block will
# not run. The Python interpreter will skip to the next line
# outside of the if-elif-else block.
        translation = "Requested web page not found on server"

# If both the initial if-statement and the first elif-statement
# return a False result, then the second elif-statement will
# run.
    elif error_code == "408 Request Timeout":
# If the second elif-statement returns a True result, then the
# indented line(s) inside the second elif-statement will run.
# After this line, the remainder of the if-elif-else block will
# not run. The Python interpreter will skip to the next line
# outside of the if-elif-else block.
        translation = "Server request to close unused connection"
```

Run