

Topic 5: Conditionals

David H Smith IV

University of Illinois Urbana-Champaign

Mon, July 05 2021

Weekly Reminders



Truth Tables

The Basic Operations: OR, AND, NOT

A	B	A OR B	A	B	A AND B	A	NOT A
T	T	T	T	T	T	T	T
T	F	F	T	F	F	F	T
F	T	T	F	T	T	T	F
F	F	F	F	F	F	F	F

Group Work: Combining Operations

A	B	(A OR B) AND (NOT B)
T	T	?
T	F	?
F	T	?
F	F	?

Group Work: Combining Operations

A	B	(A OR B) AND (NOT B)
T	T	T
T	F	F
F	T	T
F	F	F

De Morgan's Laws

- $\text{NOT}(A \text{ AND } B) = (\text{NOT}(A) \text{ OR } \text{NOT}(B))$
- $\text{NOT}(A \text{ OR } B) = (\text{NOT}(A) \text{ AND } \text{NOT}(B))$

Boolean Expressions

Truth Table to Expressions

The As and Bs in the truth tables correspond to *the result of boolean expressions*.

```
1 # Get some variables
2 x = int(input())
3 y = int(input())
4
5 # Construct the expr
6 # and assign to A or B
7 A = (x == 3)
8 B = (y > 5)
```

A	B	A OR B	A AND B	NOT A
T	T	T	T	T
T	F	F	F	T
F	T	T	T	F
F	F	F	F	F

Truth Table to Expressions

The As and Bs in the truth tables correspond to *the result of boolean expressions*.

```
1 # Get some variables
2 x = int(input())
3 y = int(input())
4
5 # Construct the expr
6 # and assign to A or B
7 A = (x == 3)
8 B = (y > 5)
```

A	B	A OR B	A AND B	NOT A
T	T	T	T	T
T	F	F	F	T
F	T	T	T	F
F	F	F	F	F

- You can have **as many operands (e.g., A, B, C, ...)** as you like.
- The truth tables get BIG as you have to consider more permutations.

Poll Questions

Poll Question: Boolean Expressions

Expressions that evaluate to True or False.

```
1 (1 + 6) < (2 + 5)
```

- ☐ A True
- ☐ B False
- ☐ C TypeError
- ☐ D SyntaxError

Poll Question: Boolean Expressions

Expressions that evaluate to True or False.

```
1 "cat" < "Dog"
```

- ☐ A True
- ☐ B False
- ☐ C TypeError
- ☐ D SyntaxError

Relational Ops on Non-numbers

A `ord("c") → 99`

Relational Ops on Non-numbers

A `ord("c")` \rightarrow 99

B `ord("D")` \rightarrow 68

Relational Ops on Non-numbers

- Ⓐ `ord("c")` \rightarrow 99
- Ⓑ `ord("D")` \rightarrow 68
- Ⓒ Strings are compared based on the ASCII values of their characters.

Relational Ops on Non-numbers

- Ⓐ `ord("c") → 99`
- Ⓑ `ord("D") → 68`
- Ⓒ Strings are compared based on the ASCII values of their characters.
- Ⓓ People often normalize strings before comparisons:
`thing1.lower() < thing2.lower()`

Conditional Branching

Poll Question: If Statements

What does this code print?

```
1 x = 1
2 if x < 7:
3     print(x)
4 print(7)
```

- ☐ A 1
- ☐ B 7
- ☒ C

1
7
- ☐ D SyntaxError

Poll Question:

What does this code print?

```
1 age = 17
2 young = age < 30
3 if young == true:
4     print(age)
```

- ☐ A Nothing
- ☐ B 17
- ☐ C 30
- ☐ D SyntaxError

Poll Question: If-Else Statements

What does this code print?

```
1 x = 2
2 if x > 8:
3     x = x - 2
4     print(x)
5 else:
6     print(8)
```

- ☐ A 0
- ☐ B 8
- ☒ C

0
8
- ☐ D SyntaxError

More Poll Questions

Poll Question:

What does `test(7)` return?

```
1 def test(num):  
2     if num > 0:  
3         return True  
4     return False
```

- ☐ A True
- ☐ B False
- ☐ C SyntaxError
- ☒ D **Always** True

Poll Question: Constructing Conditionals

Which of the following will correctly report whether a student got an A, B, or something else?

```
def print_grade(percent):  
    if grade >= 90:  
        print("You got an A!")  
    elif grade >= 80:  
        print("You got a B!")  
    else:  
        print("Other")
```

```
def print_grade(percent):  
    if grade >= 90:  
        print("You got an A!")  
    if grade >= 80:  
        print("You got a B!")  
    else:  
        print("Other")
```

```
def print_grade(percent):  
    if grade >= 90:  
        print("You got an A!")  
    if grade >= 80:  
        print("You got a B!")  
    if grade < 80:  
        print("Other")
```

- ☐ A 1
- ☐ B 2
- ☐ C 3
- ☐ D 1 and 2
- ☐ E All of the above

Poll Question: Multi-way Branches

If you were choosing between 6 possibilities, what is the fewest `elif` statements you could have?

- ☐ A 1
- ☐ B 2
- ☐ C 3
- ☐ D 4
- ☐ E 5

Poll Question: Multi-way Branches

If you were choosing between 6 possibilities, what is the fewest `elif` statements you could have?

- A** 1
- B** 2
- C** 3
- D** 4
- E** 5

```
1 if <cond>:  
2     ...  
3 elif <cond>:  
4     ...  
5 elif <cond>:  
6     ...  
7 elif <cond>:  
8     ...  
9 elif <cond>:  
10    ...  
11 else:  
12    ...  
13
```

Poll Question: If Statements

What's the result of running the following code?

```
1 x = 5  
2 y = x == 3 or 4
```

- ☐ A True
- ☐ B False
- ☐ C SyntaxError

Poll Question: If Statements

What's the result of running the following code?

```
1 x = 5
2 y = x == 3 or 4
```

- ☐ A True
- ☐ B False
- ☐ C SyntaxError

Will it ever be false?

Boolean Operators

- Ⓐ Why is `x == 3 or 4` always True?
- Ⓑ Alternatives:
 - ① `x == 3 or x == 4`
 - ② `x in [3, 4]`
- Ⓒ Types of operators:
 - ① **Binary operators:** `and`, `or`
 - ② **Unary Operators:** `not`

Truthy and Falsy

- Ⓐ Python will convert non-Boolean types to Booleans.
`if "hello":`
- Ⓑ Accomplished via the use of the `bool()` function.
`bool("hello")`
- Ⓒ All values are truthy (convert to `True`) except those displayed to the right:

- `None`
- `False`
- `0`
- `0.0`
- `0j`
- `Decimal(0)`
- `Fraction(0, 1)`
- `[]`
- `{}`
- `()`
- `,`
- `b''`
- `set()`
- `range(0)`

Poll Question: Printing with Bools

What does the following segment of code produce?

```
1 print("George") and print("Boole")
```

- ☐ A George
- ☐ B Boole
- ☐ C George
Boole
- ☐ D SyntaxError

Short Circuit

Short Circuiting

- Python is **lazy** (for efficiency reasons)
- It won't evaluate Boolean expressions it doesn't need to

```
1 True or anything() # This is True
2 False and anything() # This is False
```

- Python won't evaluate the `anything()` part.
- You can use this to prevent errors from occurring in your code or having to next if statements:

```
1 if (len(my_str) > 10) and (my_str[10] == 'a'):
2     print("the tenth character of my string is ", my_str[10])
```