

## Topic 5: Conditionals

**David H Smith IV**

**University of Illinois Urbana-Champaign**

**Mon, July 05 2021**

# 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	F
T	F	T	F	T
F	T	T	F	F
F	F	F	F	T

- 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.

is VS ==

# Muddiest Points: `is` vs `==`

- `x is y` translates to `id(x) == id(y)`
  - `==` is for comparing literal values
  - `is` is for comparing to see **two variables are referencing the same object**.
- And there's a caveat...
  - Python pre-instantiate the first few numbers. This isn't true for large numbers.

## Poll Questions: Conditional Syntax Check

# Poll Question: Constructing Conditionals

Which of the following will correctly check if a number (int or float) is between 1 and 10 inclusive.

1)

```
1 <= x <= 10
```

☐ A 1

☐ B 2

☐ C 3

☐ D 1 and 2

☐ E Another  
combo...

2)

```
1 < x < 10
```

3)

```
x >= 1 and x <= 10
```

# Poll Question: Constructing Conditionals

Which of the following evaluates to `True` if a set contains an even number of elements?

1)

```
0 == len(set1) % 2
```

2)

```
len(set1) % 2 == 0
```

3)

```
not (len(set1) % 2)
```

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

# Poll Question: Constructing Conditionals

Which of the following evaluates to `True` if `x` is greater than 10?

1)

```
x not <= 10
```

☐ A 1

☐ B 2

☐ C 3

☐ D 2 and 3

☐ E 1 and 3

2)

```
not x <= 10
```

3)

```
x > 10
```



# Poll Question: Constructing Conditionals

Which of the following will correctly check to make sure a list `x` does not contain a variable `y`.

1)

```
y not in x
```

2)

```
not (x in y)
```

3)

```
x not in y
```

☐ A 1

☐ B 2

☐ C 3

☐ D 1 and 2

☐ E All of the above

# Poll Question: Constructing Conditionals

Which of the following will correctly check to make sure a list `x` does not contain a variable `y`.

1)

```
y not in x
```

☐ A 1

2)

```
not (x in y)
```

☐ B 2

☐ C 3

3)

```
x not in y
```

☐ D 1 and 2

☐ E All of the above

With `in`, position of operands matters. With `is` and `==`, it does not.

# Poll Question: Constructing Conditionals

Which of the following will check if a value  $y$  is in a dictionary  $x$ ?

1)

```
y in x
```

☐ A 1

☐ B 2

☐ C 3

☐ D All of the above

☐ E None of the above

2)

```
y == x
```

3)

```
y is x
```

# Poll Question: Constructing Conditionals

Does this segment of code evaluate to True or False in the end?

```
y = list("hello")  
x = y  
y.append("!")  
print(y is x)
```

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

# Poll Question: Constructing Conditionals

Does this segment of code evaluate to True or False in the end?

```
y = 5  
x = y  
x = x + 1  
print(y is x)
```

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

# Poll Question: Constructing Conditionals

What is the result of the following code?

```
x = 5
y = 5
x += 1
y += 1
print(x is y)
```

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

# Poll Question: Constructing Conditionals

What is the result of the following code?

```
x = 5  
y = 5  
x += 1  
y += 1  
print(x is y)
```

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

What is I replace 5 with 1000?

# Poll Question: Constructing Conditionals

What is the result of the following code?

```
x = None  
y = None  
print(y is x)
```

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



# Poll Question: Constructing Conditionals

What is the result of the following code?

```
x = {"baz" : 2, "bar": 3}
y = {"foo": 1}
z = y
y.update(x)
print(y is z)
```

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

## Poll Questions: Correct Functions

## Poll Question: Conditionals in Functions

Which of the following functions will ask the user for a sentence and return that sentence only if it is of even length?

```
def get_even_sentence():
    x = input("Enter a string
              with an even number of
              letters: ")
    if x % 2 == 0:
        return x
```

```
def get_even_sentence():
    x = input("Enter a string
              with an even number of
              letters: ")
    if len(x) % 2 == 0:
        return x
```

```
def get_even_sentence():
    x = input("Enter a string
              with an even number of
              letters: ")
    if len(x) % 2 is 0:
        return x
```

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

# Poll Question: Conditionals in Functions

Which function(s) return true if both elem1 and elem2 are in some\_list?  
1)

```
def are_in_list(elem1, elem2, some_list):  
    return elem1 and elem2 in some_list
```

2)

```
def are_in_list(elem1, elem2, some_list):  
    return (elem1 in some_list) and (elem2 in some_list)
```

3)

```
def are_in_list(elem1, elem2, some_list):  
    return (elem1 in some_list) or (elem2 in some_list)
```

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

# Poll Question: Conditionals in Functions

Which function will return the kth word in a list of words given any list of strings only if that kth word contains an even number of letters?

1)

```
def get_kth_word_if_even_len(k, words):  
    if len(words) > k:  
        if len(words[k]) % 2 == 0:  
            return words[k]
```

2)

```
def get_kth_word_if_even_len(k, words):  
    if len(words[k]) % 2 == 0 and len(words) > k:  
        return words[k]
```

3)

```
def get_kth_word_if_even_len(k, words):  
    if len(words) > k and len(words[k]) % 2 == 0:  
        return words[k]
```

A 1

B 2

C 3

D 1 and 2

E Another option

# Poll Question: Conditionals in Functions

Which of the functions will correctly execute the task?

1)

```
def is_one_through_five(x):  
    return 1 <= x <=5
```

2)

```
def is_one_through_five(x):  
    return x in [1, 2, 3, 4, 5]
```

3)

```
def is_one_through_five(x):  
    return x == (1 or 2 or 3 or 4 or 5)
```

☐ A 1

☐ B 2

☐ C 3

☐ D 1 and 2

☐ E All of the above

# Poll Question: Nesting

What does the code on the right print?

- ☐ A 2
- ☐ B 5
- ☐ C 8
- ☐ D SyntaxError

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

## Code Blocks



# Code Blocks

- Fancy term for defining a unit of execution.

```
1 model = input('Enter car model: ')
2 year = int(input('Enter year of car manufacture: '))
3
4 antique = False
5 domestic = False
6
7 if year < 1970:
8     antique = True
9
10 if model in ['Ford', 'Chevrolet', 'Dodge']:
11     domestic = True
12
13 if antique:
14     if domestic:
15         print('My own model-T still runs like a charm...')
```

# Conditional Expressions

# Conditional Expressions vs if-else

- Follows this template: `x if <cond> else y`
- Useful for item assignment where a condition must be met in order to avoid errors
- More concise

```
1 x = input()
2 if len(x) > 9:
3     y = x[9]
4 else:
5     y = None
```

```
1 x = input()
2 y = x[9] if len(x) > 9 else
   None
```

# Poll Questions: Conditional Expressions

What is the value of `x` after this code runs and the user attempts to enter the value 10?

```
1 x = "Odd" if int(input("Enter a number")) % 2 != 0 else "Even"
```

- ☐ A SyntaxError
- ☐ B "Odd"
- ☐ C "Even"
- ☐ D This code contains another error

# Poll Questions: Conditional Expressions

What is the the resulting value of `x`?

```
1 y = 1.0  
2 x = "It's a float!" if type(y) == float
```

- ☐ A `SyntaxError`
- ☐ B `"It's a float!"`
- ☐ C `None`
- ☐ D This code contains another error