

Lecture 3: Conditionals

Eleanor Birrell

January 27, 2026

Review: Programming in Python

- Values

- 47
- `"hello, world!\n"`

- Types

- `int`
- `float`
- `str`

- Variables

- `dist_in_miles = 3.1`
- `a_string = "hello"`

- Operations

- `1 * 2 * 3`
- `a_string + " world"`

- Functions

- ```
def example(x):
 y = 2*x
 return y
```
- `z = example(25)`
- `print("hello, world!")`
- `int("50")`

# What about?

```
num1 = 20
ans1 = fav_num(num1)
```

```
My favorite number is 47, not 20.
```

```
ans2 = fav_num(47)
```

```
47 is my favorite number too!
What a coincidence!
```

# if-statements

- syntax

condition

```
if x == 47 :
 print("47 is my favorite number too!")
 print("What a coincidence!")
```

whitespace matters!

- condition must be an expression that evaluates to True or False (type bool)

# bool datatype - values

- Values: True and False

# Boolean Expressions

- Examples of operations that return a bool:
  - relational operators: ==, !=, >, <, >=, <=

```
10 < 0
11 >= 11
11 > 11.0
10 == 10.1
```

```
x != 3
```

```
y <= x
```

```
x = 5
```

# bool datatype - operators

- Values: True and False
- Operators: and, or, not
- <bool expression> **and** <bool expression>
  - True if both expressions True, False otherwise
- <bool expression> **or** <bool expression>
  - True if at least one expression True, False otherwise
- **not** <bool expression>
  - returns True if expression is False, False otherwise

# Example

Assume the variable  $n$  currently stores the value 25.  
What will the following expressions evaluate to?

$2 < n \text{ and } n < 30$

$n > 2 \text{ and } < 20$

$(2 > n) \text{ or } (n == 25)$

$\text{not } (n == n)$

$\text{not } (n != 3)$

# Exercise 1: Boolean Expressions

After executing each snippet of code, the variable `val` stores a value of type `bool`. What value does it store?

A) `val = (5 > 0 and 5 < 10)`

B) `n = 25`  
`val = (n % 2 == 0 or n % 3 == 0)`

C) `s = "string"`  
`val = ("String" != s)`

D) `val = not True and False`

# if-statements

- syntax

condition

```
if x == 47 :
 print("47 is my favorite number too!")
 print("What a coincidence!")
```

- condition must be an expression that evaluates to True or False (type bool)
  - Booleans: True, False
  - relational operators: ==, !=, >, <, >=, <=
  - logical operators: and, or, not
  - functions that evaluate to type bool

## Exercise 2: If statements

Define a function `double_or_half` that takes one parameter (an int). If the input is odd, it returns a value twice as big as the input value. If the input is even, it returns half the input value. It should always return an int.

Example behavior:

```
x = double_or_half(13)
print(x)
print(double_or_half(50))
```

should print

```
26
25
```

# if-else statements

- syntax

```
if x == 47:
 print("47 is my favorite number too!")
 print("What a coincidence!")
else:
 print("My favorite number is 47 not "+str(x))
```

# Exercise 3: if-else statements

```
def exercise3(num)
 if num % 4 == 0:
 print(4*num)

 if num % 2 == 0:
 print(2*num)
 else:
 print(num)
```

What does `exercise3(2)` print?

What about `exercise3(3)`?

What about `exercise3(4)`?

What about `exercise3(5.0)`?

# What about?

```
fav_num(47)
```

```
47 is my favorite number too!
What a coincidence!
```

```
fav_num(20)
```

```
My favorite number is bigger than 20
```

```
fav_num(105)
```

```
My favorite number is less than 105
```

# Attempt #1: If statements

```
def fav_num(x):
 if x == 47:
 print("47 is my favorite number too!")
 print("What a coincidence!")
 if x > 47:
 print("my favorite number is less than "+str(x))
 if x < 47:
 print("my favorite number is more than "+str(x))
```

# Attempt #2: nested if-statements

```
def fav_num(x):
 if x == 47:
 print("47 is my favorite number too!")
 print("What a coincidence!")
 else:
 if x > 47:
 print("my favorite number is less than "+str(x))
 else:
 print("my favorite number is more than "+str(x))
```

# Attempt #3: using elif

```
def fav_num(x):
 if x == 47:
 print("47 is my favorite number too!")
 print("What a coincidence!")
 elif x > 47:
 print("my favorite number is less than "+str(x))
 else:
 print("my favorite number is more than "+str(x))
```

# Exercise 4: elif statements

Convert the following function to a function with the same behavior that doesn't use nested if-statements

```
def exercise4(x):
 if x == 47:
 print("mine too!")
 else:
 if x > 100 or x < 0:
 print("mine is 47")
 else:
 if x == 13:
 print("13!")
 else:
 print("?")
 print(":-)")
 print("!")
```