

Business Programming (using Python)

Zhaohu (Jonathan) Fan

Information Technology Management

Scheller College of Business

Georgia Institute of Technology

September 5, 2023

Main topics

- Go over some of the Severance Chapter 3 concept
- HW1 and HW2
 - Control flow
 - Conditional Logic & Conditional Statements
 - Boolean expressions, logical operators, & relational comparison operators
 - Style Guide: Boolean Expressions

Conditional Logic & Conditional Statements

Conditional statement(s)

Basic syntax structure

```
python  
  
if condition:  
    # code to execute if condition is True
```

- The most basic form is the `if` statement, which executes a block of code **if** a specified condition is **true**.

Can I vote if I am 23?

Conditional statement: if statement

Output

```
How old are you? 23  
You're old enough to vote
```

Input

```
How old are you? 
```

if statement

Output

```
How old are you? 23  
You're old enough to vote
```

(Incorrect) code

```
# age = int(input('How old are you?'))  
# if age ≥ 18  
# print('You're old enough to vote')
```

if statement

Output

```
age = int(input('How old are you?'))  
if age >= 18  
print('You're old enough to vote')
```

File "[<ipython-input-3-d3851de01015>](#)", line 2

```
if age >= 18
```

^

SyntaxError: expected ':'

SEARCH STACK OVERFLOW

- It's missing a **colon** `:` at the end of the if statement line.

if statement

Output

```
▶ age = int(input('How old are you?'))  
if age >= 18:  
    print('You're old enough to vote')
```

```
↳ File "<ipython-input-4-c3a0065cb0fc>", line 3  
    print('You're old enough to vote')  
    ^  
IndentationError: expected an indented block after 'if'
```

- Spacing matters!
 - Make sure to **indent** the print statement properly to **maintain the code structure.**

if statement

Output/The corrected code

```
age = int(input('How old are you?'))  
if age >= 18:  
    print('You're old enough to vote')
```

```
How old are you?23  
You're old enough to vote
```

Can I vote if I am under 18?

Conditional statements: if-else

Basic syntax structure

```
python Copy code  
  
if condition:  
    # code to execute if condition is True  
else:  
    # code to execute if condition is False
```

- The `else` statement can be used alongside `if` to execute a block of code if the condition is not true.
 - Sometimes we want to do one thing `if` a logical expression is True and something `else if` the expression is False

Conditional statements: two-way

Output

```
# How old are you? 15  
# You're not allowed to vote
```

The corrected code

```
▶ age = int(input('How old are you?'))  
if age < 18:  
    print('You're not allowed to vote')  
else:  
    print('You can vote!')
```

```
How old are you?15  
You're not allowed to vote
```

Summary

Basic syntax structure

```
python Copy code
if condition_1:
    # This block of code will run if condition_1 is true
    print("Condition 1 is true")
elif condition_2:
    # This block of code will run if condition_1 is false and condition_2 is true
    print("Condition 2 is true")
else:
    # This block of code will run if both condition_1 and condition_2 are false
    print("Both conditions are false")
```

- The most basic form is the `if` statement, which executes a block of code if a specified condition is true.
 - The `else` statement can be used alongside `if` to execute a block of code if the condition is not true.
 - **You can also use `elif` (short for `else if`) to test multiple conditions.**

Others: Nested if-else statement

Output

```
x is greater than 5, but y is not
```

The corrected code

```
python Copy code  
  
x = 10  
y = 5  
if x > 5:  
    if y > 5:  
        print("Both x and y are greater than 5")  
    else:  
        print("x is greater than 5, but y is not")  
else:  
    print("x is not greater than 5")
```

Boolean expressions & relational comparison
operators & Boolean logic

Boolean Expressions (conditions)

More about conditions....

- **Boolean expressions** ask a question and evaluate to either a **Yes or No (True | False)** result which is used to control program flow
- **Relational comparison** operators compare the values held by different variables, or literals (e.g., 18).

Boolean expressions

- If-statement conditions need Boolean expressions
- Examples:
 - $5 > 0$
 - True
 - $10 < 100$
 - True
 - `10 > 5`
 - Error:cannot compare a number (integer) to a string
 - Need data type conversion

Relational comparison operators

Operator	Use	Example
<	Less than	if (a<b)
<=	Less than or equal to	if (a<=b)
>	Greater than	if (a>b)
>=	Greater than or equal to	if (a>=b)
==	Equal	if (a==b)
!=	Not equal	if (a!=b)

Different to the assignment operator '='

Asks if both sides of == are logically equivalent?

```
a = 5
b = 2
a == b
```

False

```
a = 5
b = 5
a == b
```

True

```
c = 5
d = 6
c = 6
c == d
```

True