CS 1340 Introduction to Computing Concepts

Instructor: Xinyi Ding

Sep 6 2019, Lecture 5

Agenda

- Agenda:
  - Quick review of concepts from last lecture
  - Control flow, If statements
  - For/while loops

Dictionaries

- Dictionaries store a mapping between a set of keys and a set of values
  - Keys can be any immutable type (usually string), must be unique
  - Values can be any type
    - values can be another dictionary or list, etc.
- Dictionaries are mutable, you can define/add/modify/delete key-value pairs of a dictionary

Dictionaries

- Creating and accessing dictionaries
  - Use {} to create a dictionary (we use [] for list)
  - Specify the key-value pair if not empty
  - Use [] and key to access elements (we use [] and index for list)

```
>>> emp_dict = {}
>>> user_info = {'name': 'ethan', 'age': 23, 'address': {'state': 'TX', 'zip': 75206}}
>>> print(user_info['name'])
ethan
>>> print(user_info['address'])
{'state': 'TX', 'zip': 75206}
>>> print(user_info['address']['zip'])
75206
>>>
```

Conditional test (Boolean expression)

- Programming often involves examining a set of conditions (conditional test) and deciding which action to take based on those conditions.
- Return Boolean data type: True or False
- If statement allows you to examine the current state of a program and respond appropriately to that state

Control flow

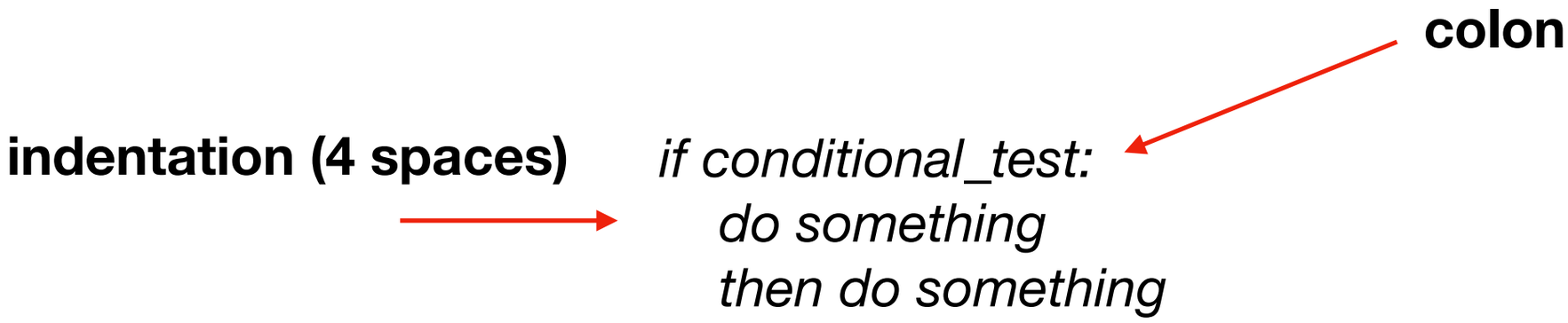
- Python execute code line by line from top to bottom.
- if statements allow you to take different actions based on different situations

- if statements
  - Use of indentation for blocks
  - Colon(:) after boolean expression.
  - For example:

```
if conditional_test:
    do something
```

if statements

- if statements syntax



- Be careful when using whitespace for indentation
  - use the same number of spaces for indentation. PEP-8 recommends 4 whitespaces. (2 spaces, tab are also valid)
  - you can use any number of spaces in other cases, though valid, but not recommended. (say, x = 5)

if statements

- Simple if statements
  - if the conditional test is True, then execute the following statements, otherwise ignore.

```
1    age = 19
2    if age >= 18:
3        print("You are old enough to vote!")
4        print("Have you registered to vote yet?")
5
6
```

```
1    age = 19
2    if age >= 18:
3    print("You are old enough to vote!")
4    print("Have you registered to vote yet?")
5
6
```

if_statements

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
  File "/Users/xinyi/Courses/cs1340/week2/if_statements.py", line 3
    print("You are old enough to vote!")
        ^
IndentationError: expected an indented block

Process finished with exit code 1
```

if statements

- if-else statements
  - Often, you will want to take one action when a conditional test passes and a different action in all other cases

```
1    age = 17
2    if age >= 18:
3        print("You are old enough to vote!")
4        print("Have you registered to vote yet?")
5    else:
6        print("Sorry, you are too young to vote")
7        print("Please register to vote as soon as you turn 18!")
8
```

else

if_statements ×

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Sorry, you are too young to vote
Please register to vote as soon as you turn 18!

Process finished with exit code 0
```

if statements

- if-elif-else chain
  - When you need to test more than two possible situations.

```python
age = 12

if age < 4:
    print("Your admission cost is $0")
elif age < 18:
    print("Your admission cost is $5")
else:
    print("Your admission cost is $10")
```

```
if_statements
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Your admission cost is $5
```

```python
age = 12

if age < 4:
    price = 0
elif age < 18:
    price = 5
else:
    price = 10

print("Your admission cost is $" + str(price))
```

elif age < 18

```
if_statements
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Your admission cost is $5
```

if statements

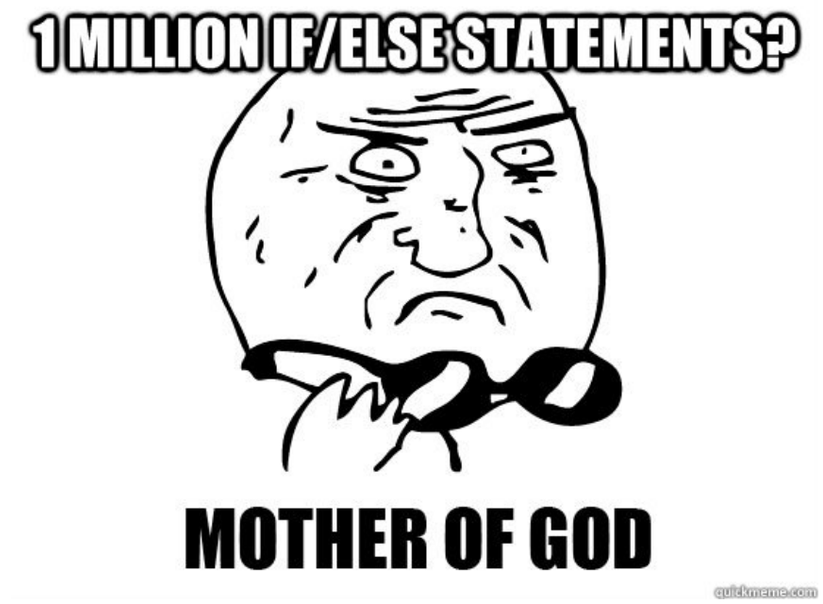
- Using multiple *elif* blocks
  - You can use as many *elif* blocks in your code as you like

```
1   age = 12
2
3   if age < 4:
4       price = 0
5   elif age < 18:
6       price = 10
7   elif age < 65:
8       price = 15
9   else:
10      price = 5
11
12  print("Your admission cost is $" + str(price))
```

else

if_statements

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Your admission cost is $10
```



if statements

- Omitting the *else* block
  - Python does not require an *else* block at the end of an *if-elif* chain.
  - Sometimes, an else block is useful, sometimes it is clearer to use an additional elif statement that catches the specific condition of interest

```
1   age = 12
2
3   if age < 4:
4       price = 0
5   elif age < 18:
6       price = 10
7   elif age < 65:
8       price = 15
9   elif age >= 65:
10      price = 5
11
12  print("Your admission cost is $" + str(price))
```

elif age >= 65

if_statements

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Your admission cost is $10
```

if statements

- Testing multiple conditions
  - The *if-elif-else* chain is powerful, but it's only appropriate to use when you just need one test to pass.

```
1   age = 12
2
3   if age < 4:
4       price = 0
5   elif age < 18:
6       price = 10
7   elif age < 65:
8       price = 15
9   else:
10      price = 5
11
12  print("Your admission cost is $" + str(price))
```

else

if_statements ×

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Your admission cost is $10
```

Once the conditional test age < 18 is True, it will stop and ignore others

if statements

- Testing multiple conditions
  - Use a series of simple *if* statements with no *elif* or *else* blocks

```
requested_toppings = ["mushrooms", "extra cheese"]

if "mushrooms" in requested_toppings:
    print("Adding mushrooms.")

if "pepperoni" in requested_toppings:
    print("Adding pepperoni.")

if "extra cheese" in requested_toppings:
    print("Adding extra cheese.")

print("Finished making your pizza!")
```

```
if_statements
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Adding mushrooms.
Adding extra cheese.
Finished making your pizza!
```

if statements

- Testing multiple conditions
  - if we use if-elif-else block

```
1   requested_toppings = ["mushrooms", "extra cheese"]
2
3   if "mushrooms" in requested_toppings:
4       print("Adding mushrooms.")
5
6   elif "pepperoni" in requested_toppings:
7       print("Adding pepperoni.")
8
9   elif "extra cheese" in requested_toppings:
10      print("Adding extra cheese.")
11
12  print("Finished making your pizza!")
13
```

elif "pepperoni" in requested_t...

if_statements ×

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
Adding mushrooms.
Finished making your pizza!
```

if statements

- Other values are treated as equivalent to either True or False when used in conditionals:
  - False: zero, None, empty containers (empty list [])
  - True: non-zero numbers, non-empty objects

```
1   this_is_a_list = []
2
3   if this_is_a_list:
4       print("this is not an empty list")
5   else:
6       print("this is an empty list")
```

if this_is_a_list

if_statements

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week2/if_statements.py
this is an empty list
```

Demo