Lecture 8:  
Conditionals & Control Flow  
(Sections 5.1-5.7)  
CS 1110  
Introduction to Computing Using Python

[E. Andersen, A. Bracy, D. Gries, L. Lee, S. Marschner, C. Van Loan, W. White]

http://www.cs.cornell.edu/courses/cs1110/2022sp

Announcements

- A1: a1_first.py & policy_acknowledgement submission
  - Submit whatever you have at 2pm
  - Keep submitting as you make significant changes
  - Final submission due tonight at 11:59pm
- Conditionals—today’s topic—not allowed in A1

What should I wear today?

```python
def what_to_wear(temp):
    print("Today you should wear:")
    # > 60: no jacket required
    if temp > 60:
        print("# > 60: no jacket required")
    # 40-60: jacket
    elif 40 < temp <= 60:
        print("# 40-60: jacket")
    # 20-40: winter coat
    elif 20 < temp <= 40:
        print("# 20-40: winter coat")
    # < 20: all the gear you own
    if temp < 20:
        print("# < 20: all the gear you own")
```

How to we implement this in Python?

What are Boolean expressions?

Expressions that evaluate to a Boolean value.

```python
is_rainy = False
is_windy = True
temp = 12

# Comparison operations
if temp < 30 and is_rainy:
    print("Roads will be icy!")
if temp > 70:
    print("Hallelujah!")

# Boolean variables
if is_rainy:
    print("Bring an umbrella!")
if is_windy and not is_rainy:
    print("Let's fly a kite!")
```
What gets printed, Round 1

<table>
<thead>
<tr>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>print(a)</td>
<td>a = a + 1</td>
<td>if a == 0:</td>
<td>if a == 1:</td>
<td>if a == 0:</td>
</tr>
<tr>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td>a = a + 1</td>
</tr>
<tr>
<td>print(a)</td>
<td>print(a)</td>
<td>print(a)</td>
<td>print(a)</td>
<td>print(a)</td>
</tr>
</tbody>
</table>

(Let's look at these one by one.)

What gets printed? (Question)

a = 0
if a == 0:
    a = a + 1
if a == 0:
    a = a + 2
a = a + 1
print(a)

A: 0
B: 1
C: 2
D: 3
E: I do not know

Conditionals: If-Else-Statements

**Format**

```python
if <boolean-expression>:
    <statement>
else:
    <statement>
```

**Example**

```python
if curr_score > high_score:
    print("New record!")
else:
    print("Nice try.")
```

Execution:

If `boolean-expression` is true, then execute statements indented under `if`; otherwise execute the statements indented under `else`.

What gets printed, Round 2

<table>
<thead>
<tr>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>if a == 0:</td>
<td>if a == 1:</td>
<td>if a == 1:</td>
<td>if a == 1:</td>
<td></td>
</tr>
<tr>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td></td>
</tr>
<tr>
<td>else:</td>
<td>else:</td>
<td>else:</td>
<td>else:</td>
<td></td>
</tr>
<tr>
<td>a = a + 2</td>
<td>a = a + 2</td>
<td>a = a + 2</td>
<td>a = a + 1</td>
<td></td>
</tr>
<tr>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td>a = a + 1</td>
<td></td>
</tr>
<tr>
<td>print(a)</td>
<td>print(a)</td>
<td>print(a)</td>
<td>a = a + 1</td>
<td></td>
</tr>
<tr>
<td>print(a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Let's look at these one by one.)

Program Flow (car locked, 0)

```python
def get_in_car(is_locked):
    car_locked = True
    if is_locked:
        print("Unlock car!")
    else:
        print("Open the door.")
    return car_locked
```

if determines which statement is executed next

1. if is_locked:
2.    print("Unlock car!")
3.    print("Open the door.")

Global Space

Flow Program only takes one path during an execution (something will not be executed!)
def get_in_car(is_locked):
    if is_locked:
        print("Unlock car!")
        print("Open the door.")
car_locked = True
get_in_car(car_locked)

Unlock car!
Open the door.
Program Flow (car not locked, 0)

```python
def get_in_car(is_locked):
    if is_locked:
        print("Unlock car!")
        print("Open the door.")
    car_locked = False
    get_in_car(car_locked)
```

Program Flow (car not locked, 1)

```python
def get_in_car(is_locked):
    if is_locked:
        print("Unlock car!")
        print("Open the door.")
    car_locked = False
    get_in_car(car_locked)
```

Program Flow (car not locked, 2)

```python
def get_in_car(is_locked):
    if is_locked:
        print("Unlock car!")
        print("Open the door.")
    car_locked = False
    get_in_car(car_locked)
```

Program Flow (car not locked, 3)

```python
def get_in_car(is_locked):
    if is_locked:
        print("Unlock car!")
        print("Open the door.")
    car_locked = False
    get_in_car(car_locked)
```

Program Flow (car not locked, 4)

```python
def get_in_car(is_locked):
    if is_locked:
        print("Unlock car!")
        print("Open the door.")
    car_locked = False
    get_in_car(car_locked)
```

Program Flow (car not locked, 5)

```python
def get_in_car(is_locked):
    if is_locked:
        print("Unlock car!")
        print("Open the door.")
    car_locked = False
    get_in_car(car_locked)
```
What does the call frame look like next? (Q)

```python
def max(x, y):
    if x > y:
        return x
    return y
```

Current call frame:

```
max(0, 3)
```

What does the call frame look like next? (Q)

```python
def max(x, y):
    if x > y:
        return x
    return y
```

Current call frame:

```
max(0, 3)
```

---

Program Flow and Variables

Variables created inside `if` continue to exist past `if`:

```python
a = 0
if a == 0:
    b = a + 1
print(b)
```

…but are only created if the program actually executes that line of code

---

Control Flow and Variables (Q1)

```python
def max(x, y):
    # Returns: max of x, y
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger
```

```
maximum = max(3, 0)
```

Value of `maximum`?

- A: 3
- B: 0
- C: Error!
- D: I do not know

---

Control Flow and Variables (Q2)

```python
def max(x, y):
    # Returns: max of x, y
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger
```

```
maximum = max(0, 3)
```

Value of `maximum`?

- A: 3
- B: 0
- C: Error!
- D: I do not know
Program Flow and Variables

```python
def zero_or_one(a):
    if a == 1:
        b = 1
    else:
        b = 0
    print(b)
```

Conditionals: If-Elif-Else-Statements (1)

**Format**
```
if <Boolean expression>:
    <statement>
elif <Boolean expression>:
    <statement>
...
elif <Boolean expression>:
    <statement>
else:
    <statement>
```

**Example**
```
# Find the winner
if score1 > score2:
    winner = "Player 1"
elif score2 > score1:
    winner = "Player 2"
else:
    winner = "Players 1 and 2"
```

Conditionals: If-Elif-Else-Statements (2)

**Format**
```
if <Boolean expression>:
    <statement>
elif <Boolean expression>:
    <statement>
...
elif <Boolean expression>:
    <statement>
else:
    <statement>
```

**Notes on Use**
- No limit on number of **elif**
- Must be between **if**, **else**
- **else** is optional
- **if-elif** by itself is fine
- Booleans checked in order
- Once Python finds a true **<Boolean-expression>**-**else** skips over all the others
- **else** means all **<Boolean-expression>** are false

If-Elif-Else (Question)

```python
a = 2
if a == 2:
    a = 3
elif a == 3:
    a = 4
print(a)
```

What gets printed?

A: 2
B: 3
C: 4
D: I do not know

What gets printed, Round 4

<table>
<thead>
<tr>
<th>a = 2</th>
<th>a = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>if a == 2:</td>
<td>if a == 2:</td>
</tr>
<tr>
<td>a = 3</td>
<td>a = 3</td>
</tr>
<tr>
<td>elif a == 3:</td>
<td>if a == 3:</td>
</tr>
<tr>
<td>a = 4</td>
<td>a = 4</td>
</tr>
<tr>
<td>print(a)</td>
<td>print(a)</td>
</tr>
</tbody>
</table>

The logic can get a little dizzying...

```python
def what_to_wear(raining, freezing):
    if raining and freezing:
        print("Wear a waterproof coat.")
    elif raining and not freezing:
        print("Bring an umbrella.")
    elif not raining and freezing:
        print("Wear a warm coat!")
    else:
        print("A sweater will suffice.")
```
**Nested Conditionals to the rescue!**

```python
def what_to_wear(raining, freezing):
    if raining:
        if freezing:
            print("Wear a waterproof coat.")
        else:
            print("Bring an umbrella.")
    else:
        if freezing:
            print("Wear a warm coat!")
        else:
            print("A sweater will suffice.")
```

**Program Flow and Testing**

```
# determine winner
print('before the if')
if x_score > y_score:
    print('inside the if')
    winner = "x"
else:
    print('inside the else')
    winner = "y"
print('after the if')
```

**Traces (control) and Watches (data)**

```
# determine winner
print('before the if')
if x_score > y_score:
    print('inside the if')
    winner = 'x'
    print('winner = ' + winner)
else:
    print('inside the else')
    winner = 'y'
    print('winner = ' + winner)
print('after the if')
```