Logic and Multiway Decisions

Announcements

- Quiz 1 on Friday
- Lab 2 due Saturday
  - Run `update21` again to get questionnaire file
  - Use pencil and paper
- Ninja sessions tonight and Friday
- My office hours: tomorrow 2-4pm

Today's plan

- Quick quiz review
- Review Monday's lecture
- Logic with `and`, `or`, `not`
- Multiway decision making
- Examples

Quiz 1

- Compute expressions, identify types
  - Types: `int`, `float`, `string`, `list`
  - Operators: `+`, `-`, `*`, `/`, `%`
  - Conversion functions: `int()`, `float()`, `str()`
  - `range` function
  - Integer division, promotion
  - String concatenation, replication

Quiz 1

- Understand and write basic programs that use:
  - `raw_input()` to get input
  - Conversion functions
  - Assignment statements
  - String concatenation
  - `print()` to display output
  - Basic for loops

```
for i in range(n):
```

Review

- We can “ask” yes-no “questions” with the comparison operators
  - `==`, `!=`, `<`, `>`, `>=`, `<=`
- We can represent the answer to a yes-no question with a *Boole*an
  - `True`, `False`
- We can perform instructions based on the answer to a yes-no question with `if` and `if-else` statements

Logical Operators

- Before: ‘If I’m hungry, then I’ll eat’
- Now: ‘If I’m hungry **and** it’s morning, then I’ll eat eggs. If I’m hungry **and** it’s evening, then I’ll eat pizza.’
- Now: ‘If I chop onions **or** watch *Titanic*, then I’ll cry.’
- Now: ‘If there are **not** cars coming, then I’ll cross the street.’

Logical operators: syntax

- `<A> and <B>`
  - `True` if `<A>` and `<B>` evaluate to `True`, `False` otherwise
- `<A> or <B>`
  - `True` if `<A>` or `<B>` evaluates to `True`, `False` if both evaluate to `False`
- `not <A>`
  - `True` if `<A>` evaluates to `False`, `False` if it evaluates to `True`

Truth Tables

<A>	<B>	<A> and <B>	<A> or <B>
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Multiway decision making

- Syntax:

```
if <condition 1>:
    <block 1>
elif <condition 2>:
    <block 2>
else:
    <block 3>
```

Multiway decision making

- Semantics: Test the conditions in order. When the first condition evaluates to True, perform the corresponding block of code, then skip past the rest of the if-elif-else statement. If none of the conditions is True, perform the code in the else block.

- Can have many elif's

- The else is optional

Multiway decision making

- Can nest if's, if-else's and if-elif-else's inside of each other.

```
if <condition>:
    if <sub-condition>:

    else:

else:
```

- There are always multiple ways to express the same logic.

Good luck on the quiz!