

## Objectives

- Advanced conditionals
- Boolean operators
- **sys** module

Feb 8, 2016

Sprengle - CSCI111

1

## Animations

Feb 8, 2016

Sprengle - CSCI111

2

## Review

- How can we make Python code execute only under certain circumstances?
- How do we say “otherwise” in Python?

Feb 8, 2016

Sprengle - CSCI111

3

## Review: Syntax of **if** statement: Simple Decision

```
if condition :  
    statement1  
    statement2  
    ...  
    statementn
```

keyword

“then” Body

- Note indentation

English Examples:

```
if it is raining :  
    I will wear a raincoat  
if the PB is new :  
    Remove the seal
```

Feb 8, 2016

Sprengle - CSCI111

4

## Syntax of **if** statement: Two-Way Decision

```
if condition :  
    statement1  
    statement2  
    ...  
    statementn  
else :  
    statement1  
    statement2  
    ...  
    statementn
```

keywords

English Example:

```
if it is Saturday or Sunday :  
    I wake up at 10 a.m.  
else :  
    I wake up at 7 a.m.
```

“then” Body

“else” Body

Feb 8, 2016

Sprengle - CSCI111

5

## Review: Relational Operators

- Syntax:  
    > <expr> <relational\_operator> <expr>

	Relational Operator	Meaning
Low precedence	<	Less than?
	<=	Less than or equal to?
	>	Greater than?
	>=	Greater than or equal to?
	==	Equals?
	!=	Not equals?

Feb 8, 2016

Sprengle - CSCI111

6

## Examples: Using Conditionals

- Determine if a number is even or odd

```
x = eval(input("Enter a number: "))
remainder = x%2
if remainder == 0 :
    print(x, "is even")
if remainder == 1:
    print(x, "is odd")
```

Feb 8, 2016

Sprengle - CSCI111

evenorodd.py

7

## Examples: Using Conditionals

- Determine if a number is even or odd

```
x = eval(input("Enter a number: "))
remainder = x%2
if remainder == 0 :
    print(x, "is even")
if remainder == 1:
    print(x, "is odd")
```

```
x = eval(input("Enter a number: "))
remainder = x % 2
if remainder == 0:
    print(x, "is even")
else:
    print(x, "is odd")
```

Why is this a more efficient implementation?

Feb 8, 2016

## Common Mistake:

### Assignment Operator vs. Equality Operator

- Assignment operator: =
- Equality operator: ==

```
x = eval(input("Enter a number: "))
remainder = x%2
if remainder = 0 :
    print(x, "is even.")
```

Syntax error

Feb 8, 2016

Sprengle - CSCI111

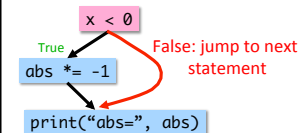
9

## If-Else statements (absolute values)

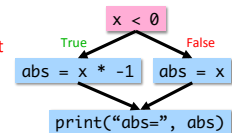
```
abs=x
if x < 0 :
    abs *= -1
print("abs=", abs)
```

```
if x < 0 :
    abs = x * -1
else :
    abs = x
print("abs=", abs)
```

### If statement



### If-else statement



Feb 8, 2016

Sprengle - CSCI111

10

## Practice: Numeric to Letter Grade

- Determine a numeric grade's letter grade (A, B, C, D, or F)

Numeric Grade	Letter Grade
90 and above	A
80 to below 90	B
70 to below 80	C
60 to below 70	D
Below 60	F

Feb 8, 2016

Sprengle - CSCI111

11

## Using the building blocks:

### Nesting if-else statements

```
if condition :
    if condition :
        statements
    else:
        statements
else:
    statements
```

if-else statement is nested inside the if

Feb 8, 2016

Sprengle - CSCI111

12

## Syntax of if statement: Multi-Way Decision

```

if condition :
    <then-body>
elif condition :
    <then-body2>
elif condition :
    <then-body3>
    ...
else:
    <default-body>
    
```

keywords

### English Example:

```

if it is Saturday:
    I wake up at 10 a.m.
elif it is Sunday:
    I wake up at 9 a.m.
else:
    I wake up at 7 a.m.
    
```

Feb 8, 2016

Sprengle - CSCI111

13

## Using the building blocks: Nesting if-else statements

```

if condition :
    statements
else:
    if condition :
        statements
    else :
        statements
    
```

if-else statement is nested inside the else

This structure can be rewritten as an if-elif-else statement

Feb 8, 2016

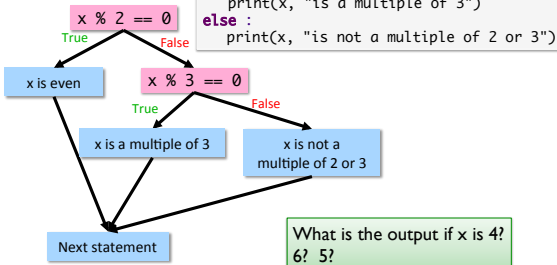
Sprengle - CSCI111

14

## If-Else-If statements

```

if x % 2 == 0 :
    print(x, "is a multiple of 2")
elif x % 3 == 0 :
    print(x, "is a multiple of 3")
else :
    print(x, "is not a multiple of 2 or 3")
    
```



Feb 8, 2016

Sprengle - CSCI111

15

## Modify to use elif

- Determine if a numeric grade is a letter grade (A, B, C, D, or F)

Numeric Grade	Letter Grade
90 and above	A
80 to below 90	B
70 to below 80	C
60 to below 70	D
Below 60	F

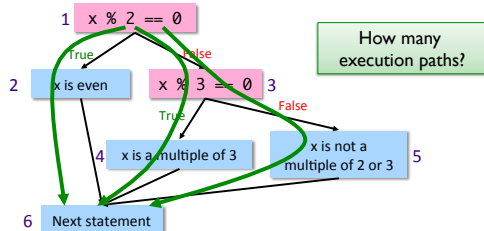
Feb 8, 2016

Sprengle - CSCI111

16

## Testing with If Statements

- Make sure have test cases that execute each branch in control flow diagram
  - i.e., Each execution path is "covered"



Feb 8, 2016

Sprengle - CSCI111

17

## Review: Speeding Ticket Fines

- Any speed clocked over the limit results in a fine of at least \$50, plus \$5 for each mph over the limit, plus a penalty of \$200 for any speed over 90mph.
- Our program
  - Input: speed limit and the clocked speed
  - Output: either (a) that the clocked speed was under the limit or (b) the appropriate fine

What should our test cases be?

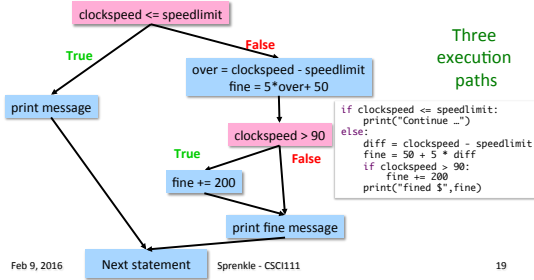
Feb 8, 2016

Sprengle - CSCI111

speedingticket.py 18

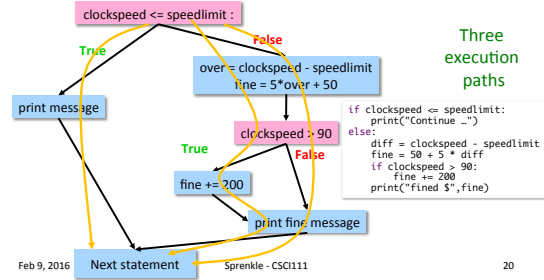
## Review: Testing with `if` Statements

- Make sure *at least* have test cases that execute each branch in control flow diagram
  - i.e., Each execution path is "covered"



## Review: Testing with `if` Statements

- Make sure have test cases that execute each branch in control flow diagram
  - i.e., Each execution path is "covered"



## More Complex Conditions

- Boolean
  - Two logical values: True and False
- Combine conditions with Boolean operators
  - and** – True only if **both** operands are True
  - or** – True if **at least one** operand is True
  - not** – True if the operand is not True
- English examples
  - If it is raining **and** it is cold
  - If it is Saturday **or** it is Sunday
  - If the shirt is on sale **or** the shirt is purple

## What is the output?

```
x = 2
y = 3
z = 4
```

```
b = x==2
c = not b
d = (y<4) and (z<3)
print("d=", d)
d = (y<4) or (z<3)
print("d=", d)
```

Because of precedence, we don't need parentheses

```
d = not d
print(b, c, d)
```

## Truth Tables

operands

A	B	A and B	A or B	not A	not B	not A and B	A or not B
T	T						
T	F						
F	T						
F	F						

## Truth Tables

operands

A	B	A and B	A or B	not A	not B	not A and B	A or not B
T	T	T	T	F	F	F	T
T	F	F	T	F	T	F	T
F	T	F	T	T	F	T	F
F	F	F	F	T	T	F	T

## Practice: Numeric Grade Input Range

- Enforce that user must input a numeric grade between 0 and 100
  - In Python, we can't (always) write a condition like `0 <= num_grade <= 100`, so we need to break it into two conditions
- Write an appropriate condition for this check on the numeric grade
  - Using **and**
  - Using **or**

Feb 8, 2016

Sprengle - CSCI111

25

## Practice: Numeric Grade Input Range

- Enforce that user must input a numeric grade between 0 and 100

➢ Using **and**

```
if num_grade >= 0 and num_grade <= 100:  
    computation  
else:  
    print error message
```

➢ Using **or**

```
if num_grade < 0 or num_grade > 100:  
    print error message  
else:  
    computation
```

Feb 8, 2016

Sprengle - CSCI111

26

## Short-circuit Evaluation

- Don't necessarily need to evaluate all expressions in a compound expression
- A **and** B
  - If A is **False**, compound expression is **False**
- A **or** B
  - If A is **True**, compound expression is **True**
- No need to evaluate B
  - Put more important/limiting expression first
  - Example: 

```
if count != 0 and sum/count > 10:  
    do something
```

Feb 8, 2016

Sprengle - CSCI111

27

## Looking Ahead

- Lab 4 tomorrow
- No Broader Issue this week

Feb 8, 2016

Sprengle - CSCI111

28