

Objectives

- Conditional statements

Feb 7, 2018

Sprenkle - CSCI111

1

Review

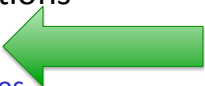
- What makes a function “good”?
- How do we typically use a function that returns something?

Feb 7, 2018

Sprenkle - CSCI111

2

Parts of an Algorithm

- Input, Output
 - Primitive operations
 - What data you have, what you can do to the data
 - Naming
 - Identify things we're using
 - Sequence of operations
 - Conditionals
 - Handle special cases
 - Repetition/Loops
 - Subroutines
 - Call, reuse similar techniques
- 

Feb 7, 2018

Sprenkle - CSCI111

3

Making Decisions

- Sometimes, we do things only if some condition holds (i.e., “is true”)
- Examples
 - If the PB is new (has a safety seal)
 - Then, I will take off the safety seal
 - If it is raining and it is cold
 - Then, I will wear a raincoat
 - If it is Saturday or it is Sunday
 - Then, I will wake up at 9 a.m.
 - Otherwise, I wake up at 7 a.m.
 - If the shirt is purple or the shirt is on sale and blue
 - Then, I will buy the shirt

Feb 7, 2018

Sprenkle - CSCI111

4

Conditionals

- Sometimes, we only want to execute a statement in certain cases
 - Example: Finding the absolute value of a number
 - $|4| = 4$
 - $|-10| = 10$
 - To get the answer, we multiply the number by -1 *only if it's a negative number*
 - Code:

```
if x < 0 :  
    abs = x*-1
```

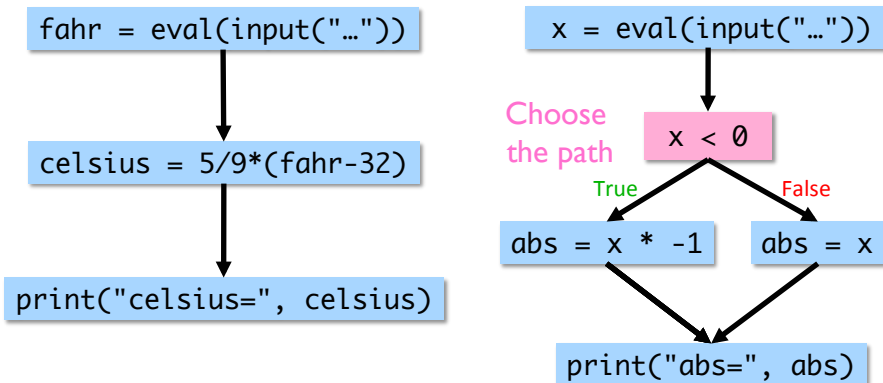
Feb 7, 2018

Sprenkle - CSCI111

5

if Statements

- Change the *control flow* of the program



Feb 7, 2018

Sprenkle - CSCI111

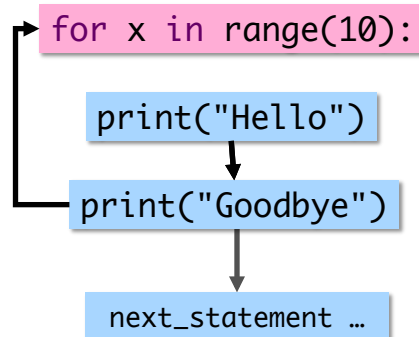
6

Other Constructs That Change Control Flow

- **for** loops

- Repeats a loop body a fixed number of times before going to the next statement after the **for** loop

```
for x in range(10):  
    print("Hello")  
    print("Goodbye")  
next_statement ...
```



Feb 7, 2018

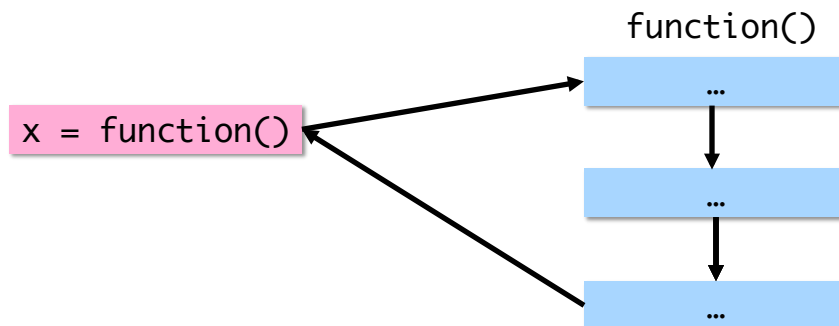
Sprenkle - CSCI111

7

Other Constructs That Change Control Flow

- Function calls

- “Go execute some other code and then come back with the result”



Feb 7, 2018

Sprenkle - CSCI111

8

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 7, 2018

Sprenkle - CSCI111

9

Conditions

- Syntax (typical, others later):
 - `<expr> <relational_operator> <expr>`
- Evaluates to either **True** or **False**
 - Boolean type

Feb 7, 2018

Sprenkle - CSCI111

10

Relational Operators

- Syntax:

➤ `<expr> <relational_operator> <expr>`

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

Feb 7, 2018

Sprenkle - CSCI111

Use Python interpreter

11

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 7, 2018

Sprenkle - CSCI111

evenorodd.py

12

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 (pointing to the single equals sign in the if statement)

Feb 7, 2018

Sprenkle - CSCI111

13

Syntax of `if` statement:

Two-Way Decision

English Example:

if condition :

statement1
statement2
...
statementn

else :

statement1
statement2
...
statementn

keywords (pointing to **if** and **else**)

"then" Body (bracketed next to the first block of statements)

"else" Body (bracketed next to the second block of statements)

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

Feb 7, 2018

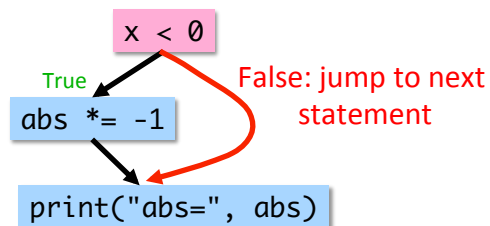
Sprenkle - CSCI111

14

If-Else statements (absolute values)

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

If statement

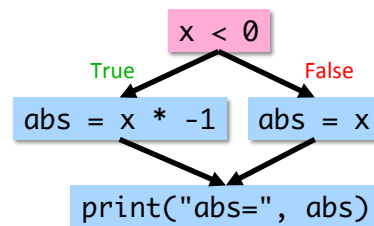


Feb 7, 2018

Sprenkle - CSCI111

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

If-else statement



15

Examples: Using Conditionals

- Determine if a number is even or odd
- More efficient implementation
 - Don't need to check if remainder is 1 because if it's not 0, it *must* be 1

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

Feb 7, 2018

Sprenkle - CSCI111

16

Practice: Draw the Flow Chart

```
print("This program determines your birth year")
print("given your age and current year")
print()
age = eval(input("Enter your age: "))

if age > 120:
    print("Don't be ridiculous, you can't be that old.")
else:
    currentYear = eval(input("Enter the current year: "))
    birthyear = currentYear - age
    print()
    print("You were either born in", birthyear, end='')
    print("or", birthyear-1)
print("Thank you. Come again.")
```

What does this code do?

Feb 7, 2018

Sprenkle - CSCI111

17

Flow of Control: Using **return**

Is this implementation of
the function correct?

```
def max(num1, num2):
    if num1 >= num2:
        return num1
    else:
        return num2
```

Feb 7, 2018

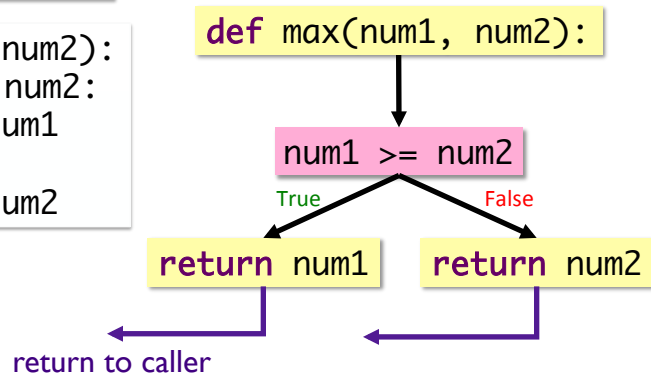
Sprenkle - CSCI111

18

Flow of Control: Using **return**

Is this implementation of the function correct?

```
def max(num1, num2):  
    if num1 >= num2:  
        return num1  
    else:  
        return num2
```



Feb 7, 2018

Sprenkle - CSCI111

19

Flow of Control: Using **return**

Is this implementation of the function correct?

```
def max(num1, num2):  
    if num1 >= num2:  
        return num1  
    return num2
```

Feb 7, 2018

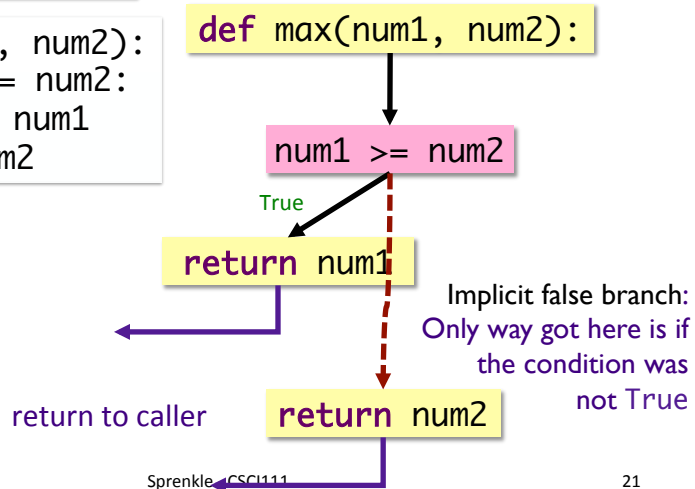
Sprenkle - CSCI111

20

Flow of Control: Using **return**

Is this implementation of the function correct?

```
def max(num1, num2):  
    if num1 >= num2:  
        return num1  
    return num2
```



Feb 7, 2018

Sprenkle - CSCI111

21

Nondeterministic Decisions

- Sometimes, we don't want to necessarily know that a specific decision is always made
- For example, games often use randomness to make decisions
 - Roll dice
 - Coin flips
 - Location and behavior of baddies

How can we simulate coin flips?

Feb 7, 2018

Sprenkle - CSCI111

22

Flipping Coins

- Simulate by randomly selecting between 0 (heads) and 1 (tails)
- Program: coinFlip.py

```
from random import randint

HEADS=0
TAILS=1

# flip the coin
if randint(0,1) == HEADS:
    print("heads")
else:
    print("tails")
```

Feb 7, 2018

Sprenkle - CSCI111

23

Practice: 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

speedingticket.py

Feb 7, 2018

Sprenkle - CSCI111

24

Our Test Cases So Far

Speed limit	Clocked speed	Expected
25	26	\$55
30	32	\$60
50	65	\$125
70	95	\$375
20	15	☹
90	91	\$255
91	91	☺

Feb 7, 2018

Sprenkle - CSCI111

25

Looking Ahead

- Exam
 - Your Questions

Feb 7, 2018

Sprenkle - CSCI111

26