CSC 111: Intro to Computer Science through Programming

Spring 2017
Prof. Sara Mathieson
Outline: 1/30

- Notecards
- Recap last time
- Creating and using Python files
- New keywords, syntax, concepts
- chaos.py
- Liaisons
Notecards
On a notecard, please fill out:

1) Name (first and last)
2) BannerWeb name/or and Moodle name if different from above
3) Pronoun
4) Previous programming experience
5) Most advanced math course taken or currently taking
6) House
7) Favorite hobby
8) Anything else that would be helpful for me to know!
Recap
Code from last time

```python
>>> def greet(person):
    print("Hello " + person + "!")

>>> greet("Tani")
Hello Tani!

>>> x = 3
```
Informal quiz (discuss with a partner)

1) What is this?

```python
>>> def greet(person):
    print("Hello " + person + "!")
```

2) What are “def” and “print” in Python?

```python
>>> greet("Tani")
Hello Tani!
```

3) Since it is between the (), “person” is a...

4) This entire block of code is a...

5) This line is _________ “greet”

```
>>> x = 3
```

6) x is a _________ and 3 is the ______ of x

7) The colon (:) and tab indentation are examples of Python...
Informal quiz (discuss with a partner)

1) prompt

2) What are “def” and “print” in Python?

>>> def greet(person):
    print("Hello " + person + "!")

3) Since it is between the (), “person” is a...

4) This entire block of code is a...

>>> greet("Tani")
Hello Tani!

5) This line is __________ “greet”

>>> x = 3
6) x is a ________ and 3 is the ______ of x

7) The colon (:) and tab indentation are examples of Python...
Informal quiz (discuss with a partner)

1) prompt

2) “def” is a **keyword** and “print” is a **built-in function** (used to be a keyword in Python 2)

3) Since it is between the (), “person” is a...

4) This entire block of code is a...

5) This line is __________ “greet”

   >>> greet("Tani")

   Hello Tani!

6) x is a ________ and 3 is the ______ of x

   >>> x = 3

7) The colon (:) and tab indentation are examples of Python...
Informal quiz (discuss with a partner)

1) prompt

2) “def” is a **keyword** and “print” is a **built-in function** (used to be a keyword in Python 2)

3) “person” is a **parameter** or **argument**

4) This entire block of code is a...

5) This line is _________ “greet”

Hello Tani!

6) x is a _________ and 3 is the _______ of x

7) The colon (:) and tab indentation are examples of Python...
Informal quiz (discuss with a partner)

1) prompt

2) “def” is a keyword and “print” is a built-in function (used to be a keyword in Python 2)

3) “person” is a parameter or argument

4) This entire block of code is a function

5) This line is _________ “greet”

   >>> greet("Tani")

   Hello Tani!

6) x is a _________ and 3 is the _______ of x

   >>> x = 3

7) The colon (:) and tab indentation are examples of Python...
Informal quiz (discuss with a partner)

1) prompt

2) “def” is a **keyword** and “print” is a **built-in function** (used to be a keyword in Python 2)

3) “person” is a **parameter** or **argument**

4) This entire block of code is a **function**

5) This line is **invoking** or **calling** “greet”

```
>>> greet("Tani")
Hello Tani!
```

6) x is a ________ and 3 is the _______ of x

7) The colon (:) and tab indentation are examples of Python...
Informal quiz (discuss with a partner)

1) prompt

2) “def” is a **keyword** and “print” is a **built-in function** (used to be a keyword in Python 2)

3) “person” is a **parameter** or **argument**

4) This entire block of code is a **function**

```
>>> def greet(person):
    print("Hello " + person + ".")
```

5) This line is **invoking** or **calling** “greet”

```
>>> greet("Tani")
Hello Tani!
```

6) x is a **variable** and 3 is the **value** of x

```
>>> x = 3
```

7) The colon (:) and tab indentation are examples of Python...
Informal quiz (discuss with a partner)

1) prompt

```python
>>> def greet(person):
    print("Hello " + person + "!")

>>> greet("Tani")
Hello Tani!
```

2) "def" is a keyword and "print" is a built-in function (used to be a keyword in Python 2)

3) "person" is a parameter or argument

4) This entire block of code is a function

5) This line is invoking or calling "greet"

6) x is a variable and 3 is the value of x

7) The colon (:) and tab indentation are examples of Python syntax
Python files +
New keywords, vocab, concepts
Why use files?

- Once we close IDLE, our work is lost!
- Files allow us to separate code into tasks that will work together later on
- Python files will have extension .py
- The shell will be able to run the code written in these files

Follow along: https://www.python.org/downloads/
Today’s concepts

+ **Comment**: not executed like a statement (ignored by Python)
  - Extremely important so others can read your code
  - Use a hashtag # to denote a comment
Today’s concepts

+ **Comment**: not executed like a statement (ignored by Python)
  - Extremely important so others can read your code
  - Use a hashtag `#` to denote a comment

+ **Variable**: space in memory reserved for a (changeable) value
Today’s concepts

- **Comment**: not executed like a statement (ignored by Python)
  - Extremely important so others can read your code
  - Use a hashtag `#` to denote a comment

- **Variable**: space in memory reserved for a (changeable) value

- **Assignment**: giving a variable a (computed) value
Today’s concepts

+ **Comment**: not executed like a statement (ignored by Python)
  - Extremely important so others can read your code
  - Use a hashtag # to denote a comment

+ **Variable**: space in memory reserved for a (changeable) value

+ **Assignment**: giving a variable a (computed) value

+ For loops and new keywords: `for`, `in`
Today’s concepts

+ **Comment**: not executed like a statement (ignored by Python)
  - Extremely important so others can read your code
  - Use a hashtag `#` to denote a comment

+ **Variable**: space in memory reserved for a (changeable) value

+ **Assignment**: giving a variable a (computed) value

+ For loops and new keywords: `for`, `in`

+ Built-in functions: `eval`, `input`, `range`
Today’s concepts

- **Comment**: not executed like a statement (ignored by Python)
  - Extremely important so others can read your code
  - Use a hashtag `#` to denote a comment

- **Variable**: space in memory reserved for a (changeable) value

- **Assignment**: giving a variable a (computed) value

- For loops and new keywords: `for, in`

- Built-in functions: `eval, input, range`

- Mathematical functions: `+,-,*,/`
Today’s concepts

- **Comment**: not executed like a statement (ignored by Python)
  - Extremely important so others can read your code
  - Use a hashtag # to denote a comment

- **Variable**: space in memory reserved for a (changeable) value

- **Assignment**: giving a variable a (computed) value

- For loops and new keywords: `for, in`

- Built-in functions: `eval, input, range`

- Mathematical functions: `+, -, *, /`

- Explore: **documentation**
IDLE demo

Integrated Development Environment

To find IDLE:
Mac: Applications
Windows: All Programs
Python modules

+ Module name is the file name (minus the .py)
  + Example: age module
  + Usually will use “Run module”, then do further testing of the functions

+ Functions live inside the module
  + Example: compute_age function
Chaos program

Application: maybe natural systems exhibit chaos behavior.

Small change in inputs creates a large change over time.

Weather patterns are modeled using these types of functions.

```python
def main():
    print("This program illustrates a chaotic function")
    x = eval(input("Enter a number between 0 and 1: "))
    for i in range(10):
        x = 3.9 * x * (1-x)
        print(x)
```
New built-in functions

- **input(…)** – solicit input from the user
  - Parameter: string prompting the user
  - Returns: a string of what the user typed in

- **eval(…)** – evaluate the input
  - Parameter: expression (often a string)
  - Returns: evaluation of the expression

- **range(…)** – produces a list of numbers in a specified range
  - Parameters: start, stop, separation/skip
  - Returns: list of numbers from start (inclusive) to stop (exclusive)
Mathematical functions

- Multiplication: *
- Division: /
- Addition: +
- Subtraction: -
Resources

- Python style guide
  https://www.python.org/dev/peps/pep-0008/

- List of built-in Python functions
  https://docs.python.org/3/library/functions.html
Liaisons