Topic 2: Vars and Expressions

David H Smith IV

University of Illinois Urbana-Champaign

Thur, Aug 26 2021

Announcements

1. Homework 2 (Part 2) and Post-Reading for Topic 3 (Part 1) are posted and due Friday.
2. Participation Topic 3 (Part 1) is due Friday
3. Challenge Activities for Topic 2 are due Sunday.

Review Poll Questions

Poll Question:

How many of the following characters are **visible** on the screen?

```
1 print("\t\\n\\\t")
```

A. 1
B. 2
C. 3
D. 4
E. 5

Math Operators

Poll Question: Multiplication

What is the result of the following?

```
1 x = 2(10 + 2)
2 print(x)
```

A. 24
B. SyntaxError
C. TypeError
D. ValueError

Poll Question: Math

What is the value of `y` after this code executes?

```
1 x = 2
2 y = x + 3
3 x = 3
```

- A 2
- B 3
- C 5
- D 8
- E 10

Poll Question: Math

What is the value of `y` after this code executes?

```
1 x = 7
2 y = x
3 x = x + 2
```

1. 2
2. 5
3. 7
4. 9

Poll Question: Math

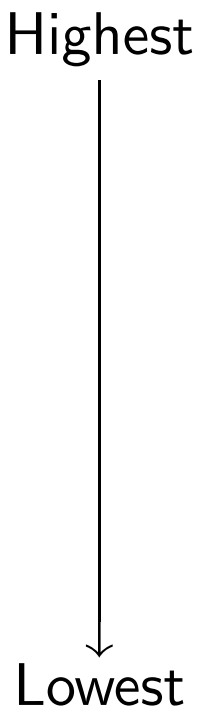
What is the value of this expression?

```
1 -3 ** 2
```

A. -9
B. -8
C. 8
D. 9

Order of Operations in Python

- Parentheses
- Exponentiation
- Positive and negative
- Multiplication, Division, Modulo
- Addition, Subtraction



Note: Python evaluates from left to right within a precedence level

Poll Question: More Math Operators

Which computes how many (whole) apples I can give to each friend?

1. `num_apples / num_friends`
2. `num_friends // num_apples`
3. `num_apples // num_friends`
4. `num_friends % num_appples`
5. `num_apples % num_friends`

Poll Question: More Math Operators

Which computes how many (whole) apples you have left over if you give `num_apples` to `num_friends`?

1. `num_apples / num_friends`
2. `num_friends // num_apples`
3. `num_apples // num_friends`
4. `num_friends % num_appples`
5. `num_apples % num_friends`

Division, Floor Division, and Modulo

1. Division operator (/) gives best approximation to true result and *always return a float*.
2. Floor division (//) rounds down the closet whole number. The type of the result will follow the normal rules.
3. Modulo operator(%) performs a division and returns the remainder. The type of the result will always be the same.
4. For any numbers x and y, the following equality holds:

```
(y == (y // x) * x + (y % x))
```

Math Module

Poll Question: More Math Operators

Which of the following will print the value of π?

A. `print(math.pi)`

B. `print(pi)`

C.
```
import math.pi
print(math.pi)
```

D.
```
import math
print(math.pi)
```

Poll Question: Rounding

What is the result of this code if the user types in `4.51` and `5.9`?

```
1 x = math.ceil(float(input()))
2 y = math.floor(float(input()))
3 print(x + y)
4
```

A. SyntaxError
B. NameError
C. 10
D. 10.0

Poll Question: Rounding

What is the result of this code if the user types in `4.1` and `5.9`?

```
1 import math
2 x = math.ceil(float(input()))
3 y = math.floor(float(input()))
4 print(x + y)
5
```

A. 10

B. 10.0

Math Module

- Should I memorize (operators, function, modules, module functions, etc.)?

Math Module

A. Should I memorize (operators, function, modules, module functions, etc.)?

B. Yes and no.

Math Module

A. Should I memorize (operators, function, modules, module functions, etc.)?

B. Yes and no.

C. Google + `help()` function are your friends

Math Module

A. Should I memorize (operators, function, modules, module functions, etc.)?

B. Yes and no.

C. Google + `help()` function are your friends

D. Modules vs Scripts: Modules are just (for the purposes of what we've discussed so far) scripts that someone else wrote that you can use in your own scripts.

Math Module

A. Should I memorize (operators, function, modules, module functions, etc.)?

B. Yes and no.

C. Google + `help()` function are your friends

D. Modules vs Scripts: Modules are just (for the purposes of what we've discussed so far) scripts that someone else wrote that you can use in your own scripts.

Math Module

A. Should I memorize (operators, function, modules, module functions, etc.)?

B. Yes and no.

C. Google + `help()` function are your friends

D. Modules vs Scripts: Modules are just (for the purposes of what we've discussed so far) scripts that someone else wrote that you can use in your own scripts.

To get information on a module:

```
1 import math
2 help(math)
```

`__name__` and `"__main__"`

1. When you run a script in python it gets a few "environment variates".
2. For the script you run (e.g, `test.py`) the `__name__` variable will always be `"__main__"`.
3. For any scripts/modules you import `__name__` variable will always be the name of that script/module.

Lab

Lab 1

Python + TextEditor

1. Everyone should be able to open a zsh terminal (Mac), powershell cmd (Windows), or Git Bash (Either) and type: `python`.
2. You should be greeted by a Python command line (e.g., >>>).

Git Setup

1. `git config`: Sets up your local
2. All of these local setting will be represented in your local "commit history".
3. When you push that commit history to GitHub (a remote repository) they will also be reflected there.

GitHub Setup

1. **Secure Socket Shell (SSH) Key:** A "password" for remote access to a repository.
2. Follow the instructions listed on the link in the lab instructions.
3. The key generation command it will give you a path to a file that ends in `.pub`
4. Open the aformentioned file and copy-paste it's contents into the key section of GitHub.

Cloning A Repository

1. Homework 2 (Part 2) and Post-Reading for Topic 3 (Part 1) are posted and due Friday.
2. Participation Topic 3 (Part 1) is due Friday
3. Challenge Activities for Topic 2 are due Sunday.
4. Lab 1 (100pts) will be due Sunday, Sept 4th:
   - Setup (50pts).
   - README.md with some stuff filled in (25pts).
   - —hello-world.py— passes the test cases (25pts).