COMP 204
Variables
Mathieu Blanchette,
based on material from Carlos Oliver and Christopher Cameron
Announcement

Midterm exam on Oct 17:
Time shifted from 6:00-8:00pm to 6:30-8:00pm.
Basic operations on numbers and strings

Operations on numbers:

- Addition: $7+12 \# 19 \text{ (type: int)}$
- Subtraction: $3.14 - 2.78 \# 0.3600000000000003 \text{ (type: float)}$
- Multiplication: $2 * 3.1416 \# 6.2832 \text{ (type: float)}$
- Division: $33 / 8 \# 3.3 \text{ (type: float)}$
  - $33 / 11 \# 3.0 \text{ (type: float)}$
- Remainder of integer division: $27 \% 10 \# \text{ (type: int)}$
- Exponentiation: $4**3 \# 4^3 = 64 \text{ (type: float)}$
- $2 + 6 * 2 - 8**2 / 4 \# -2.0 \text{ (type: float, due to division)}$
  - Note priority of operations: Exponentiation, multiplication/division, addition/substraction
- $(2 + 6) * (2 - 8**2 / 4) \# -112.0 \text{ (type: float)}$
  - Use parentheses to group terms as desired

Operations on strings:

- Concatenation: 'Hello'+ 'World' \# 'HelloWorld'
Basic operations on booleans

- **Conjunction (and):**
  - True and True  # True
  - True and False  # False
  - False and True  # False
  - False and False  # False

- **Disjunction (or):**
  - True or True  # True
  - True or False  # True
  - False or True  # True
  - False or False  # False

- **Negation (not):**
  - not True  # False
  - not False  # True
Comparisons

A comparison is an operation that compares two objects and yields a boolean value

- **Test equality**
  - \(3.14 == 3.14\) # True
  - Note the use of double-equal sign
  - \('ACTG'== 'GTCA'\) # False
  - \('ACTG'== 'acgt'\) # False

- **Test non-equality**
  - \(3.14 != 3.1416\) # True
  - \('ACGT'!= 'ACGT'\) # False

- **Smaller-than, smaller-or-equal**
  - \(3.14 < 3.1416\) # True
  - \(3.14 < 3.14\) # False
  - \(3.14 <= 3.14\) # True
  - \('ACGT'<= 'ACCT\) # True

- **Same thing for greater-than, greater-or-equal**
Mixing it up

- $(2 \times 3.14) < 6$ and 'TGA'=='TGA' # False
- $( (2 \times 3.14) < 6 \text{ or } (17-3 == 14) )$ and 'TGA'=='TGA' # True
- not ('TGA'=='TGA' or 'TGA'!='TGA') # False
- 'AA'> 4 #TypeError: '>' not supported between instances of 'str' and 'float'

So Python is just a fancy calculator?
No: Programming is about linking multiple operations together
For this, it is useful to be able to save to memory the results of an operation.

To this end, we use **variables**.
Variables

Variables allow a program to remember values throughout the execution of the program. This is how a program uses the computer’s memory. A variable has a *name* and a *value*. A program can

- Create new variables
- Set the value of variables
- Look up the value of variables to include them in expressions
- Change the value of variables (hence the name)
Variables - example

Goal: Calculate the molecular mass of complex molecules.

weightCarbon = 12
# This creates a variable weightCarbon, assigns it value 12

weightOxygen = 16
# This creates a variable weightOxygen, assigns it value 16

print('The weight of carbon is: ', weightCarbon)
# This looks up the value of variable weightCarbon, performs the print statement

print('The weight of oxygen is: ', weightOxygen)

weightCO2 = weightCarbon + 2 * weightOxygen
# This first evaluates the right-hand side, based on the current values of weightCarbon and weightOxygen. This yields 44.
# It then creates the variable weightCO2 and assigns it the value 44.
# Nothing gets printed so far

print('The weight of CO2 is:', weightCO2)
weightCarbon = 12
# This creates a variable weightCarbon,
# assigns it value 12

weightOxygen = 16
# This creates a variable weightOxygen,
# assigns it value 16

print('The weight of carbon is: ', weightCarbon)
# This looks up the value of variable weightCarbon,
# performs the print statement

print('The weight of oxygen is: ', weightOxygen)

weightCO2 = weightCarbon + 2 * weightOxygen
# This first evaluates the right-hand side,
# based on the current values of weightCarbon
# and weightOxygen. This yields 44.
# It then creates the variable weightCO2
# and assigns it the value 44.
# Nothing gets printed so far

print('The weight of CO2 is: ', weightCO2)
weightCarbon = 12
# This creates a variable weightCarbon, # assigns it value 12

weightOxygen = 16
# This creates a variable weightOxygen, # assigns it value 16

print('The weight of carbon is: ', weightCarbon)
# This looks up the value of variable weightCarbon, # performs the print statement

print('The weight of oxygen is: ', weightOxygen)

weightCO2 = weightCarbon + 2 * weightOxygen
# This first evaluates the right-hand side, # based on the current values of weightCarbon # and weightOxygen. This yields 44. # It then creates the variable weightCO2 # and assign it the value 44. # Nothing gets printed so far

print('The weight of CO2 is: ', weightCO2)
weightCarbon = 12
# This creates a variable weightCarbon, assigns it value 12

weightOxygen = 16
# This creates a variable weightOxygen, assigns it value 16

print('The weight of carbon is: ', weightCarbon)
# This looks up the value of variable weightCarbon, performs the print statement

print('The weight of oxygen is: ', weightOxygen)

weightCO2 = weightCarbon + 2 * weightOxygen
# This first evaluates the right-hand side, based on the current values of weightCarbon and weightOxygen. This yields 44.
# It then creates the variable weightCO2 and assign it the value 44.
# Nothing gets printed so far

print('The weight of CO2 is: ', weightCO2)
weightCarbon = 12
weightOxygen = 16
print('The weight of carbon is: ', weightCarbon)
print('The weight of oxygen is: ', weightOxygen)
weightCO2 = weightCarbon + 2 * weightOxygen
print('The weight of CO2 is: ', weightCO2)

# Improved measurement of atomic masses
weightCarbon = 12.001
print('The weight of CO2 is: ', weightCO2)
# weightCO2 remains 44
weightCarbon = 12
weightOxygen = 16
print('The weight of carbon is: ', weightCarbon)
print('The weight of oxygen is: ', weightOxygen)
weightCO2 = weightCarbon + 2 * weightOxygen
print('The weight of CO2 is: ', weightCO2)

# Improved measurement of atomic masses
weightCarbon = 12.001
print('The weight of CO2 is: ', weightCO2)
# weightCO2 remains 44

weightCO2 = weightCarbon + 2 * weightOxygen
# now weightCO2 becomes 44.001
print('The weight of CO2 is: ', weightCO2)
Variables - example 2

Goal: Write a program that computes the body mass index (BMI) of a person: \( BMI = \frac{weight}{(height^2)} \)

weight = 69
height = 1.8
BMI = weight/(height**2)
print('A person with weight ', weight, ' and height ', height, ' has BMI = ',BMI)

# suppose the weight changes
weight = 74

# The value of BMI still has not changed
print('A person with weight ', weight, ' and height ', height, ' has BMI = ',BMI)

# We need to recalculate BMI to get the correct BMI value
BMI = weight/(height**2)
print('A person with weight ', weight, ' and height ', height, ' has BMI = ',BMI)
Variables - example 2 improved

Goal: Write a program that asks the user for their weight and height and then computes BMI.
How? Use the input(String) function, which prompts the user to enter data, and returns the string that was typed.

```python
weight = input('Please enter your weight (in kg)'
height = input('Please enter your weight (in m)'
BMI = weight/(height**2)
print('Your BMI is ',BMI)
```

Problem: We get an error:
TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'

Use the Python shell to find out what the type of the weight and height variables are.
```
type(weight)    # OMG, it’s a String, not an integer
type(height)    # and this one too!
```
That’s because the input function always produces a string, irrespective of what is actually typed by the user.
Converting between types

Python allows data to be converted from one type to another using type conversion functions:
- `int(someObject)` creates an integer whose value is based on `someObject`
- `float(someObject)` creates a float whose value is based on `someObject`
- `str(someObject)` creates a string whose value is based on `someObject`

Examples:
- `name='Mathieu'` # name is a String
- `weight='68 '` # weight is a String
- `height='1.8 '` # height is a String
- `weightInt = int(weight)` # weightInt is an int, with value 68
- `heightFloat = float(height)` # heightFloat is a float, with value 1.8
- `heightInt = int(height)` # heightInt is an int with value 1.
  # Note: int() truncates decimal values
- `nameInt = int(name)` # this causes an error, because the content of the name String cannot be converted to a number
BMI program corrected

We use the type conversion functions to convert the output of the input function to float.

```python
weight = input('Please enter your weight (in kg)')
weightFloat = float(weight)
height = input('Please enter your weight (in m)')
heightFloat = float(height)
BMI = weightFloat/(heightFloat**2)
print('Your BMI is ',BMI)
```

Or more succinctly, we directly convert the output of the input function to a float, without saving the String in a variable:

```python
weight = float( input('Please enter your weight (in kg)') )
height = float( input('Please enter your weight (in m)') )
BMI = weight/(height**2)
print('Your BMI is ',BMI)
```