

Variables and I/O

Types

- Strings
 - Enclosed in quotation marks
 - "Hello, World!"
- Integers
 - 4, 3, 5, 65
- Floats
 - 4.5, 0.7
- What about "56"?

Variables and Assignment

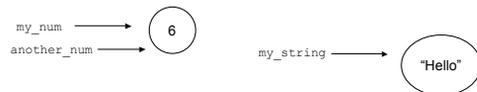
- A name that refers to a value
- Python uses *dynamic typing*

```
my_num = 6
my_string = "Hello"
another_num = my_num
```

Variables and Assignment

- = often read as "gets the value"
- `my_num` and `another_num` *refer* to the same *object*

```
my_num = 6
my_string = "Hello"
another_num = my_num
```



Variables and Assignment

- Numbers and strings are *immutable*

```
my_num = 6
my_string = "Hello"
another_num = my_num
my_num = 7
my_num = "CS"
```



Variable Names

- A combination of letters, digits, and _
- Must begin with a letter
- Case sensitive
- OKAY
 - `csiscool`, `my_variable` `variable2`
- NOT OKAY
 - `cs is cool`, `2ndvariable`, `print`
- Why not `print`?

Exercises

1. Assign the value 9 to the variable *my_num*
2. Assign the value "17" to the variable *my_string*
3. Print *my_num+my_string*
4. What happens?
5. Assign the value 17 to the variable *my_string*
6. Print *my_num+my_string*
7. What happens?
8. Assign the value "print" to the variable *print_var*
9. What happens?

Operators

- You've seen +
- -, *, /, ** (exponentiation)
- % - remainder
 - $12\%6$
 - $12\%5$
- What is the result of $5/2$?

Operators

- What is the result of $5/2$? **2**
- Why?
 - if both operands are integers, integer division is performed and the result must be an integer
 - result is truncated

Precedence

- PEMDAS
 - parentheses
 - exponents
 - multiplication
 - division
 - addition
 - subtraction
- Evaluation done left to right

Alternatives

- +=, -=, *=, /=
- `num += 3` -> `num = num + 3`

Exercises

1. Determine the results of the following:
 1. $5+9/4*3-2$
 2. $(5+9)/(4*(3-2))$
 3. $5**2+1/4-4$
 4. $5**(2+1)/(4-5)$
 5. $5**(2+1)/(4-4)$
 6. $((4-2)/(3-8))$
 7. $((5+3)/3(2+1))$

Strings

- Concatenation
 - print "Hello, " + "World!"
 - print "Hello " + "Class!"
 - print "Hello" + "Class!"
- Repetition
 - print "Hello" * 3
 - print "Hello," * 3

Strings

- Can be in single or double quotes
 - "hello" or 'hello'
- Escape sequences encode special characters
 - \n = newline, \t = tab, \\ = \, \" = ", \' = '
 - can also use " in string enclosed by " and ' in string enclosed by "'
 - "it's fun", 'a "sample" string'
 - 'it\'s fun', "a \'sample\' string"
- <http://docs.python.org/ref/strings.html>
 - lists python escape sequences

Exercises

1. Try the following commands:
 1. print "\tName: Bob"
 2. print "\t Name:\n Bob"
 3. print "Name:\a Bob"
 4. print "\a"*10

Composition

- What is the result of the following:

```
age = 19
print "Your age is " + age
```
- Instead, use ',' to compose statements

```
age = 19
print "Your age is ", age
```

Keyboard Input

- `input(<prompt>)` reads an integer/float from the keyboard
- `raw_input(<prompt>)` reads a string from the keyboard
- Syntax
 - `variable_name = input(<prompt>)`
 - `variable_name = raw_input(<prompt>)`
- Examples
 - `mynum = input("Enter number: ")`
 - `mystring = raw_input("Enter string: ")`

Keyboard Input

- Examples

```
mynum = input("Enter number: ")
# same as
print "Enter number: "
mynum = input()
```
- Recall, an int can be a string, but a string cannot be an int

Exercises

1. Write the algorithm for a program that prompts the user for two integers and displays the sum, difference, product, and quotient of the numbers
2. Write a program that implements the algorithm you wrote for exercise 1

Exercises

3. Write the algorithm for a program that stores your name, age, street number, street name, city, state, and zip code in separate variables and displays the data in the following format:

```
My name is : Mickey Mouse  
My age is: 75  
My address is: 1234 Main Street, San Francisco, CA  
94121
```

4. Write a program that implements the algorithm you wrote for exercise 3