

Data Types

- Different data types we will see in programming:
 - Integer
 - Float
 - Char/string
 - Boolean
 - Array/List
 - Dictionary/Hash

Integer

- Integers are “whole” numbers
 - 1, 4, 324
- Usually just write as “int” in programming context
- All integer operations return integer value
- Special attention to division
 - $5/2$ is 2 !! (not 2.5)

in python 2.x

Float

- Floats are numbers with decimal precision:
 - 2.3, 0.003, 324.34
- Operations between floats return float
- Operations between float and int return float
 - $3.14 + 1$ is 4.14
 - $3 - 1.0$ is 2.0
 - $5.0 / 2$ is 2.5

Char and String

- char is a single character
 - 'a' , '1' , '=' , ' '
- String is a collection of characters
 - "Name" , "Last Name" , "A more complicated string"
- **Note:** in python there is only one data type for both string and char. All are called string
 - The type of all of these are string
 - 'a' , "a" , "-" , 'Last Name' , "just an example string"

Arithmetic Operations

- An operand is a number or variable used by the operator

```
balance = credit - debit
```

- Result of operator depends on the types of operands
 - if both operands are int, the result is int too.
 - if one or both operators are double, the result is double

Integer vs. Double Devision

- (Assume we use Python 2.x) Devision with at least one operator of type double produces the expected results.

```
print 17.0/5
```

```
3.4
```

- Result is the same if either dividend or divisor is of type int.
- Be careful with the integer devision!

```
print 17/5
```

```
3
```

Arithmetic Expressions

- Use spacing to make expressions readable
 - Which is easier to use?

$x+y^*z$ or $x + y^* z$

Arithmetic Expressions

- Use spacing to make expressions readable
 - Which is easier to use?

$x+y^*z$ or $x + y^* z$

- Precedence rules for operators are the same as used in algebra classes.
- Use parentheses to alter the order of operations
 - $x + y^* z$ (y is multiply z first)
 - $(x + y)^*z$ (x and y are added first)

Arithmetic Expressions

**Mathematical
Formula**

**Python
Expression**

$$b^2 - 4ac$$

`b*b - 4*a*c`

$$x(y + z)$$

`x*(y + z)`

$$\frac{1}{x^2 + x + 3}$$

`1/(x*x + x + 3)`

$$\frac{a + b}{c - d}$$

`(a + b)/(c - d)`

Composite Assignment Operator

- There is a shorthand notation that combines the assignment operator (=) and an arithmetic operator so that a given variable can have its value changed.
- The general form is `Variable Op= Expression`

Example	Equivalent to
<code>count += 2</code>	<code>count = count + 2</code>
<code>total -= discount</code>	<code>total = total - discount</code>
<code>bonus *= 2</code>	<code>bonus = bonus * 2</code>
<code>time /= rush_factor</code>	<code>time = time / rush_factor</code>
<code>change %=100</code>	<code>change = change % 100</code>
<code>amount *= cnt1 + cnt2</code>	??

Composite Assignment Operator

- There is a shorthand notation that combines the assignment operator (=) and an arithmetic operator so that a given variable can have its value changed.
- The general form is `Variable Op= Expression`

Example	Equivalent to
<code>count += 2</code>	<code>count = count + 2</code>
<code>total -= discount</code>	<code>total = total - discount</code>
<code>bonus *= 2</code>	<code>bonus = bonus * 2</code>
<code>time /= rush_factor</code>	<code>time = time / rush_factor</code>
<code>change %=100</code>	<code>change = change % 100</code>
<code>amount *= cnt1 + cnt2</code>	<code>amount = amount*(cnt1 + cnt2)</code>

Composite Assignment Operator

- What is the value of x after the following statements?

```
x = 0
```

```
x += 3.0 * 4
```

```
x -= 2
```

- A. 22.0
- B. 12.0
- C. 10.0
- D. 14.0

Boolean Type

- A boolean type is that whose variables can have only two values: `False` and `True`.

```
# prints the value of boolean variables:  
flag = False  
print "flag =" , flag  
  
flag = True  
print "flag =" , flag
```

Boolean Type

- A boolean type is that whose variables can have only two values: `False` and `True`.

```
# prints the value of boolean variables:  
flag = False  
print "flag =" , flag  
  
flag = True  
print "flag =" , flag
```

```
flag = False
```

```
flag = True
```

Boolean Type

- The following actions are possible, but generally not recommended!
- Values of type boolean can participate in arithmetic computation with numeric values (both integer and floating point)
 - True is treated as 1
 - False is treated as 0
- Try these:
 - `x = True`
 - `print x * 5.0`
 - `print x - 4`

Programming Example

- We want to write a program that computes the balance of your account.
- How can we compute the account balance?
- $\text{balance} = \text{amount of debit} - \text{amount of credit}$

Programming Example

- Computer needs a set of instruction (an algorithm)
- Facts:
 - computer does not know any thing! (it includes: what is debit? what is credit? where are they?)

Programming Example

- Algorithm:
 1. Give debit to the computer
 2. Give credit to the computer
 3. tell computer to subtract credit from debit and store it somewhere
 4. tell computer to print that the value which was stored in previous step!

Programming Example

- Let's rewrite it in a more formal way:
- Algorithm:
 1. Get a value of debit from user and store it in a variable called `debit`
 2. Get a value of credit from user and store it in a variable called `credit`
 3. Set `balance = debit - credit`
 4. `print balance`

Asking the user for input

- Sometimes we would like to take the value for a variable from the user via their keyboard.
- You can pass a string to `raw_input` to be displayed to the user before pausing to get the input:

```
>>> name = raw_input("what is your name?\n")
```

- `raw_input()` always return the input as string.
- You can try to convert the input (which is string) to integer value using the `int()` function:

```
>>> x = raw_input("Enter an integer number:\n")  
>>> x = int(x)
```

A simple Python program to compute balance

- Open a file and write this piece of code and run it

```
debit = raw_input("Please enter your debit: ")
debit = int(debit)
credit = raw_input("Please enter your credit: ")
credit = int(credit)

balance = credit - debit

print "Your account balance is: ", balance
```

```
Please enter your debit: 50
Please enter your credit: 200
Your balance is: 150
```

Let's practice

- Write a program that allows the user to enter a time in seconds and then outputs how far an object would drop if it is in freefall for that length of time. Assume that the object starts at rest, there is no friction or resistance from air, and there is a constant acceleration of 32 feet per second squared due to gravity. Use the equation:

$$dist = \frac{acceleration * time^2}{2}$$

Program Design Process

- Programming is a creative process
 - There is no complete set of rules for creating a program
- Problem solving phase
 - Result is an algorithm that solves the problem.
- Implementation phase
 - Result is the algorithm translated into a Python code.

Program Design Process

Design an algorithm

1. Set acceleration = 32
2. Get a value for time from the user
3. Compute dist using this formula (acceleration*time^2) / 2
4. Display dist

step	dist	acceleration	time
1	/	/	/
2	/	32	/
3	/	32	5
4	400	32	5

$$\frac{32 \times 5^2}{2} = 400$$

More Example

- Write a program that allows the user to enter a value for radius (of a circle) in centimetres and then outputs the area and perimeter of the circle
- We know the equations!
 - area $r^2\pi$
 - perimeter $2r\pi$
- Assume that user enters whole numbers as radius

Program Design Process

Design an algorithm

1. $\pi = 3.1415$
2. Get a value for radius from the user
3. $\text{area} = \pi * \text{radius} * \text{radius}$
4. $\text{perimeter} = \pi * \text{radius} * 2$
5. Display area
6. Display perimeter

More Example

- Write a program that asks the user to enter the length and width (in feet) of a rectangle carpet and determines its total cost given the material cost is 23\$ per square yard.
- Assume that user enters whole numbers as length and width
- We know that each square yard is 9 square feet!

Suggested Reading

- Chapter 2