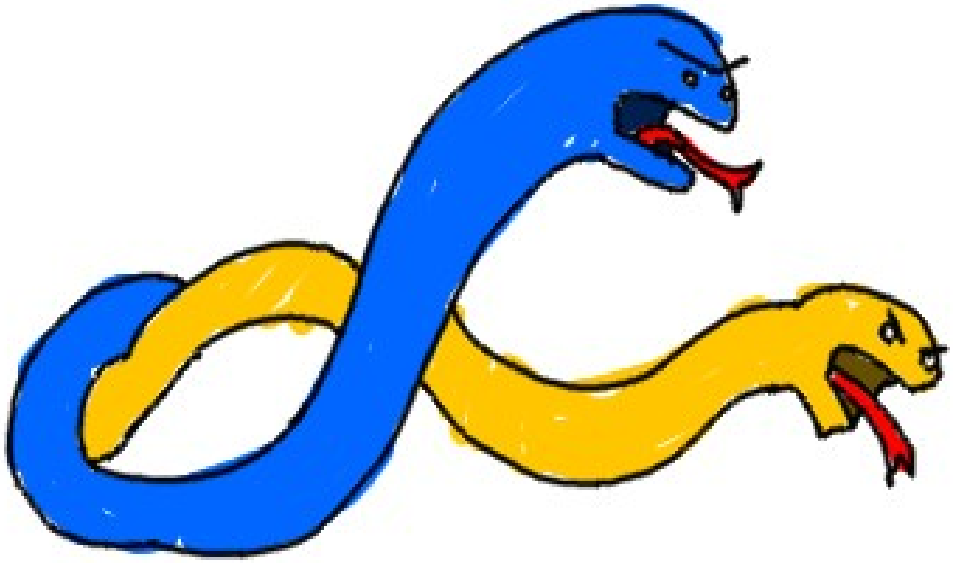
Python vs. C++



Written by Rachel J. Morris, last updated 2013-05-24

Conditional Statements

- Conditional Statements

Conditional	Python	C++
Equal to	`if ( var1 == var2 ):`	`if ( var1 == var2 )`
Not equal to	`if ( var1 != var2 ):`	`if ( var1 != var2 )`
Less than	`if ( var1 < var2 ):`	`if ( var1 < var2 )`
Less than or equal to	`if ( var1 <= var2 ):`	`if ( var1 <= var2 )`
True (way 1)	`if ( var1 == True ):`	`if ( var1 == true )`
True (way 2)	`if ( var1 ):`	`if ( var1 )`
False (way 1)	`if ( var1 == False ):`	`if ( var1 == false )`
False (way 2)	`if ( not var1 ):`	`if ( !var1 )`

Conditional Statements

- Linking Conditional Statements together
  - Are two (or more) statements true? Is one true and one false? Are both false?

Python

```
if ( age > 18 and age < 21 ):
    print( "You can vote, but you cannot drink!" )

if ( grade == "F" or grade == "D" ):
    print( "You failed :(" )
```

C++

```
if ( age > 18 && age < 21 )
{
    cout << "You can vote, but you cannot drink!" << endl;
}

if ( grade == "F" || grade == "D" )
{
    cout << "You failed! :(" << endl;
}
```

Conditional Statements

- Conditional Statements are used both for **If Statements**, and **Loops**.
- They always are either **True** or **False**

If Statements

- Python uses **if**, **elif**, and **else**.
- C++ uses **if**, **else if**, and **else**.

Python

```
if ( balance < 0 ):
    print( "You owe money!" )
elif ( balance == 0 ):
    print( "You have no money in your account" )
else:
    print( "How much would you like to withdraw?" )
```

If Statements

- Python uses **if**, **elif**, and **else**.
- C++ uses **if**, **else if**, and **else**.

C++

```
if ( balance < 0 )
{
    cout << "You owe money!" << endl;
}
else if ( balance == 0 )
{
    cout << "You have no money in your account" << endl;
}
else
{
    cout << "How much would you like to withdraw?" << endl;
}
```

Switch Statements

- C++ also has **Switch Statements**, which work specifically with integers.
- Switch statements only check for whether an **int** variable **equals** a value; won't do < or > checks.

Switch Statements

```
cout << "Choose a menu option:" << endl;
cout << "1. Play game" << endl;
cout << "2. View Help" << endl;
cout << "3. Quit" << endl;

int choice;
cin >> choice;

switch( choice )
{
    case( 1 ):
        PlayGame();  // Function call
    break;

    case( 2 ):
        DisplayHelp();   // Function call
    break;

    case( 3 ):
        return 0;    // Quit program
    break;

    default:
        cout << "Invalid option!" << endl;
    break;
}
```

Switch Statements

```
switch( nextLevel )
{
    case 5:
        rank = “Adventurer”;
        score += 50
    break;

    case 10:
        rank = “Knight”

    case 11:
        score += 100
    break;

    default:
        score += 10
    break;
}
```

- The **switch** statement encloses the entire behavior based on the **variable's value**.
- Can contain **case** statements and a **default** value. Default is executed if none of the **case** statements match the variable's value.

Switch Statements

```
switch( nextLevel )
{
    case 5:
        rank = “Adventurer”;
        score += 50
    break;

    case 10:
        rank = “Knight”

    case 11:
        score += 100
    break;

    default:
        score += 10
    break;
}
```

- Break is put at the end of the **case** condition. **break;**
- If break; isn't added, then the program will ***also*** execute the code in the **case** statement immediately below.
- This may be something you want in some cases.

Nesting If Statements

- You can “Nest” blocks of code like **if statements** or **loops** within other **if statements** and **loops**.

```
if ( playerAlive == True ):
    if ( state == "battle" ):
        print( "A: Attack" )
        print( "B: Block" )
        print( "D: Dodge" )
        print( "R: Run away" )

    elif ( state == "navigating" ):
        print( "N: North" )
        print( "S: South" )
        print( "E: East" )
        print( "W: West" )

    input( "What will you do next?" )

else:
    print( "Game Over" )
```

Nesting If Statements

```
if ( playerAlive == True ):
    if ( state == "battle" ):
        print( "A: Attack" )
        print( "B: Block" )
        print( "D: Dodge" )
        print( "R: Run away" )

    elif ( state == "navigating" ):
        print( "N: North" )
        print( "S: South" )
        print( "E: East" )
        print( "W: West" )

    print( "What will you do next?" )
    choice = input( "? " )

else:
    print( "Game Over" )
```

```
if ( playerAlive == true ) {
    if ( state == "battle" ) {
        cout << "A: Attack" << endl;
        cout << "B: Block" << endl;
        cout << "D: Dodge" << endl;
        cout << "R: Run away" << endl;
    }
    else if ( state == "navigating" ) {
        cout << "N: North" << endl;
        cout << "S: South" << endl;
        cout << "E: East" << endl;
        cout << "W: West" << endl;
    }

    cout << "What will you do next?\n? ";
    cin >> choice;
}
else {
    cout << "Game Over" << endl;
}
```

If Statements

Additional Reading

- http://www.cplusplus.com/doc/tutorial/control/
- http://www.learncpp.com/cpp-tutorial/52-if-statements/